How polarimetry may contribute to deep road scene analysis in adverse weather conditions

Rachel Blin

Normandie Université

THÈSE

Pour obtenir le grade de Docteur de Normandie Université

Spécialité Informatique

l'École Doctorale Mathématiques, Information, Ingéniérie des Systèmes

How polarimetry may contribute to deep road scene analysis in adverse weather conditions

Comment l'imagerie polarimétrique peut contribuer à l'analyse profonde de scènes routières en conditions météorologiques dégradées

Présentée et soutenue par

Rachel Blin

Dirigée par Samia Ainouz

Thèse soutenue publiquement le [28/09/2021] devant le jury composé de

Mme. Véronique Cherfaoui	Professeure, l'Université de Technologie de Compiègne, Heudiasyc	Rapportrice
M. Vincent Frémont	Professeur, Centrale Nantes, LS2N	Rapporteur
M. Frédéric Précioso	Professeur à Polytech'Nice-Sophia, I3S	Examinateur
M. Olivier Morel	Habilité à diriger des recherches, Université de Bourgogne, ImViA	Examinateur
Mme. Samia Ainouz	Professeure, INSA de Rouen, LITIS	Directrice
M. Fabrice Mériaudeau	Professeur, Université de Bourgogne, ImViA	Co-encadrant
M. Stéphane Canu	Professeur, INSA de Rouen, LITIS	Co-encadrant

INSA ROUEN NORMANDIE

MI∫IΣ

litis

Acronyms

2D Two Dimensions. 31, 63–70

3D Three Dimensions. 4, 10, 23, 31, 32, 54, 55, 63–70

4D Four Dimensions. 74

Adam Adaptive moment estimation. 101, 103, 110

ADAS Advanced Driver-Assistance Systems. I, III, 16, 30, 73

AP Average Precision. 47, 93

ASFF Adaptively Spatial Feature Fusion. 59

AVOD Aggregate View Object Detection. 65

BDD100K Berkeley Deep Drive dataset. 69, 74, 89, 93, 95, 125, 131

BEV Bird Eye View. 65, 66

BiFPN Bi-directional Feature Pyramid Network. 61

CE Cross Entropy. 45, 46

CIE Lab Lightness, Green-magenta chromatic axis, Blue-Yellow chromatic axis. 16, 23, 26–28, 32, 125, 126

CNN Convolutional Neural Network. 3, 37, 39, 41, 43, 48, 51, 57, 59, 61

CPNDet Corner Proposal Network for Object Detection. 62

CycleGAN Cycle-Consistent Generative Adversarial Network. I, III, 5, 6, 11, 12, 34, 48, 49, 51, 86–89, 91, 93, 95, 121, 135, 137, 138, 141, 143

DETR DEtection TRansformer. 61, 62

DNN Deep Neural Network. 33, 34, 57, 73, 86, 98, 108, 112

DPM Deformable Part Model. 116

DSSD Deconvolutional Single Shot Detector. 60

FCOS Fully Convolutional One Stage object detector. 60

FID Fréchet Inception Distance. 91

FIR Far Infrared. 64

FL Focal Loss. 45, 46, 60, 99

FPN Feature Pyramid Networks. 43, 58–60, 62

fps Frames per second. 59, 68, 75, 78, 99, 120

FSAF Feature Selective Anchor-Free. 60

FV Front View. 63, 65, 66

GAN Generative Adversarial Networks. 48, 49

GCNet Global Context Network. 58

GPS Global Positioning System. 69

GPU Graphic Processing Unit. 67

HSV Hue, Saturation, Value. 16, 23, 25, 26, 32, 83, 85, 103, 107, 108, 112, 125, 126, 128, 135

HTC Hybrid Task Cascade. 58

ILSVRC ImageNet Large Scale Visual Recognition Challenge. 33, 67

IMU Inertial Measurement Unit. 69

IOU Intersection Over Union. 44, 47, 58, 59, 61, 118

KITTI Karlsruhe Institute of Technology and Toyota Technological Institute dataset. 4, 10, 68, 69, 74, 89, 93, 95, 108, 110, 112

LiDAR Light Detection And Ranging. I, III, 1, 4, 6, 7, 10, 12, 16, 30–32, 53, 55, 56, 63–66, 68–70, 137, 141

MAL Multiple Anchor Learning. 61

mAP Mean Average Precision. 47, 68, 93, 101, 103, 110, 125, 126, 128, 133, 134

MIR Middle Infrared. 64

MLP Multilayer Perceptron. 34, 42, 51

MS COCO Microsoft Common Object in COntext. 3, 9, 58, 60, 67, 89, 91, 99, 100, 103, 108, 110, 112, 113

MSE Mean-Squared Error. 36, 45

MV3D Multi-View 3D network. 65

MVX-Net Multimodal Voxel Network. 63

NASNets Neural Architecture Search Networks. 58

NiN Network in Network. 64

NIR Near Infrared. 64

NMS Non-Maximum Suppression. 44, 45, 57–62, 116–118, 126, 128, 130, 133, 135, 138, 142

PANet Path Aggregation Network. 58

PASCAL VOC Pattern Analysis, Statistical Modelling and Computational Learning Visual Object Classes. 66, 100, 101

R-CNN Region Based Convolutional Neural Network. 3, 9, 57–60, 63, 64, 68

R-FCN Region-based Fully Convolutional Network. 58

Radar Radio Detection And Ranging. 1, 4, 7, 10, 53, 55, 56, 64, 65, 69, 70

ReLU Rectified Linear Unit. 40

ResNet Residual Network. 57, 60, 64, 65, 99, 108, 121, 126, 130

RGB Red, Green, Blue. I, III, 3–5, 10, 11, 16, 23, 25, 26, 28, 32, 55, 62–66, 82, 83, 85, 87–89, 91, 93, 95, 98, 103, 105, 108, 110, 112, 113, 121, 125, 126, 128, 130–133, 137, 141

ROI Region Of Interest. 3, 42, 57, 58, 62, 65

RPN Region Proposal Network. 42, 58

RRPN Radar Region Proposal Network. 65

SAPD Soft-Anchor-Point Object Detection. 60

SAR Synthetic-Aperture Radar. 54, 83

SENets Squeeze-and-Excitation Networks. 57

SPPNet Spatial Pyramid Pooling Network. 57

SSD Single Shot MultiBox Detector. 3, 9, 59, 60, 65, 99

SVM Support Vector Machine. 57

TPU Tensor Processing Unit. 67

TSD Task-aware Spatial Disentanglement. 59, 68

VGG Visual Geometry Group network. 57–60, 108

YCrCb Luminance, Chrominance (red-yellow), Chrominance (blue-yellow). 16, 23, 29, 32, 125, 126, 128

YOLO You Only Look Once. 3, 9, 59, 64, 99

Glossary

2D Geometrical space where two parameters are required to determine the position of an element. XXI, 31, 69, 72

3D Geometrical space where three parameters are required to determine the position of an element. XVI, XXI, 4, 25, 27, 29, 72

4D Geometrical space where four parameters are required to determine the position of an element. 74

Adam Optimizer based on adaptive moment estimation. 101

ADAS Electronic systems assisting drivers. I

AP A metric used to measure the accuracy of a model. XXI, XXII, 47, 68, 72, 102, 105

ASFF An algorithm palliating the inconsistency across the scales of a Feature Pyramid Network. 59, 68

AVOD A network for multimodal fusion. 65

BDD100K A dataset containing road scenes. XVIII, XXII, 69, 90, 93, 94

BEV The bird eye view of a LiDAR point clouds. 65, 72

BiFPN A derivative of Feature Pyramid Networks. 61

CE A regression loss function. 45

CIE Lab A Color space that describes an object by its perceptual lightness, and the four unique colors of human vision. XVI, 15, 16, 26–28, 126–129

CNN A type of neural network mostly containing convolutional layers. 3

CPNDet A deep neural network for object detection. 62, 68

CycleGAN Unsupervised algorithm for image-to-image translation. I, XVII, XVIII, XX, 50, 51, 88, 89, 123, 124

DETR A deep neural network for object detection. 61, 68

DNN A Neural Network with more than two layers. 33

DPM Machine Learning algorithm for object detection. 116

DSSD A deep neural network for object detection. 60, 68

FCOS A deep neural network for object detection. 60, 68

FID A metric used to evaluate the quality of images generated by a GAN. 91

FIR A modality capturing waves of the Far Infrared spectrum (25μm-350μm). 64

FL A regression loss function. 45

FPN A Network with a pyramidal architecture enabling to combine low-resolution with high resolution features. XVII, 42, 43

fps A metric measuring the frame rate of a sensor. 59

FSAF An algorithm attributing attention weights to anchor boxes. 60, 68

FV The front view of a LiDAR point clouds. 63, 72

GAN Unsupervised networks enabling to generate images. 48

GCNet A deep neural network for object detection. 58, 68

GPS A device used to determine the satellite position of a vehicle. 69

GPU A device used to efficiently process images. XXI, 67, 68

HSV A Color space that describes an object by its tint, its amount of gray and its brightness value. XVI, XVIII, XIX, XXI, 15, 16, 25, 26, 73, 83–86, 104–107, 111, 112, 126, 127, 129

HTC A deep neural network for object detection. 58, 68

ILSVRC A challenge for visual recognition at a large scale. 33, 66–68

IMU A device used to determine the speed and acceleration of a vehicle. 69

IOU A metric evaluating how close is the predicted bounding box from the ground truth. XVII, XXI, 44, 72

KITTI A dataset containing road scenes. XVIII–XXII, 4, 68, 69, 72, 90, 93, 110, 176–178

LiDAR A sensor modeling a scene in 3D. I, XVI, 15, 31, 32, 69, 72

MAL An algorithm reducing anchor boxes priors. 61, 68

mAP A metric used to measure the mean accuracy of a model. XVIII, XX, 47, 72, 94, 134

MIR A modality capturing waves of the Middle Infrared spectrum (5μm-25μm). 64

MLP A neural network constituted of several layers of fully connected formal neurons. 34

MS COCO A dataset containing diverse scenes for common object detection. XIX, XXI, XXII, 3, 66–68, 93, 100, 104, 109

MSC-MultiBox A deep neural network for object detection. 59

MSE A measure that characterizes the precision of an estimator. 36

MV3D A network for multimodal fusion. 65

MVX-Net A network for multimodal fusion. 63

NASNets A deep neural network for object classification. 58

NiN A network for object classification. 64

NIR A modality capturing waves of the Near Infrared spectrum (0.7μm-5μm). 64

NMS An algorithm aiming to keep the most relevant proposition for the object detection task. XVII, XIX, XX, XXII, 44, 46, 118, 127, 129, 131, 132, 134, 135

PANet A deep neural network for object detection. 58, 68

PASCAL VOC A dataset containing diverse scenes for the object detection task. XIX, XXII, 66, 68, 100, 102

R-CNN A deep neural network for object detection. 3, 68

R-FCN A deep neural network for object detection. 58, 68

Radar A sensor using electromagnetic waves to determine the position and the speed of an object. 1, 69

ReLU An activation function which output is the maximum between 0 and the input. 40

ResNet A type of neural network that handles deep learning tasks. 57, 126, 127, 129

RGB A Color space that describes an object by the chromaticities of the red, green and blue. I, XV, XVIII–XX, XXII, XXIII, 15, 23, 24, 69, 72, 77, 78, 82–85, 89, 92–94, 104–106, 110–112, 117, 121–124, 126, 127, 129, 131, 132, 134, 135

ROI A subregion of an image which contains relevant information. XVII, 3, 42

RPN A network that returns regions of interest of an input image. XVII, 42

RRPN A network for multimodal fusion. 65

SAPD An algorithm attributing attention weights to anchor boxes. 60, 68

SAR A radar that is usually used to capture satellite images of landscapes. 54

SENets A deep neural network for object classification. 58

SPPNet A deep neural network for object detection. 57, 68

SSD A deep neural network for object detection. 3, 68

SVM A classification algorithm. 57

TPU A device used to efficiently process tensors. 67, 68

TSD An algorithm to avoid misalignment between the classification and the regression sub-networks of object detectors. 59

VGG A type of neural network that handles deep learning tasks. 57

YCrCb A Color space that describes an object by its luminance, its red-yellow chrominance and its blue-yellow chrominance. XVI, 15, 16, 29, 30, 126, 127, 129

YOLO A deep neural network for object detection. 3, 68

Notations

AP Average Precision metric. 47, 91, 101, 103, 110, 125, 126, 131, 132

A Calibration matrix of the linear polarimetric camera. XVI, XXI, 18, 20, 27, 87, 92, 169

B Orthonormal basis. XVI, 16, 27

CE Cross Entropy loss function. 45

C Value of the loss function. 36

D_X Discriminative model of the domain X. 49, 51

D_Y Discriminative model of the domain Y. 49, 51

D Discriminative model. XX, 48, 123

ER Error Rate evolution function. 91, 93, 103, 105, 111, 112, 125, 127, 129, 131, 132

E_x Amplitude of the magnetic field according to the horizontal axis. 16, 17

E_y Amplitude of the electric field according to the vertical axis. 16, 17

FL Focal Loss function. 46

FN Number of False Negatives. 47

FP Number of False Positives. 47

G Generative model. 48

H_r Height of the rectified feature map. 40

$H_{\mathbf{polar}}$ Height of the polarimetric images. 120

H Height of the input of a convolution layer. 37–40

IOU Intersection Over Union function. 44, 45, 119

I_0 Intensity corresponding to the polarizer oriented at 0°. XV, XVIII–XX, 18–20, 22, 77, 78, 82, 83, 85, 86, 88, 89, 92, 99, 100, 103, 108, 117, 118, 122, 125, 130, 169, 170, 176–178

I_{135} Intensity corresponding to the polarizer oriented at 135°. XV, XVIII, XX, 18, 20, 22, 78, 82, 83, 88, 92, 169, 170, 176–178

I_{45} Intensity corresponding to the polarizer oriented at 45°. XV, XVIII–XX, 18, 20, 22, 77, 78, 82–84, 86, 88, 92, 99, 100, 103, 108, 117, 118, 125, 130, 169, 170, 176–178

I_{90} Intensity corresponding to the polarizer oriented at 90°. XV, XVIII–XX, 18, 20, 22, 77, 78, 82, 83, 86, 88, 92, 99, 100, 103, 108, 117, 118, 125, 130, 169, 170, 176–178

I_{HH} Intensity corresponding to an horizontal transmitted and horizontal received light wave. 83

I_{HV} Intensity corresponding to an horizontal transmitted and vertical received light wave. 83

I_{VH} Intensity corresponding to an vertical transmitted and horizontal received light wave. 83

I_{VV} Intensity corresponding to an vertical transmitted and vertical received light wave. 83

I_{α_i} Intensity corresponding to the polarizer oriented at an angle α_i. XV, 17–19, 82

I Intensities vector. XXI, 18, 20, 83, 87, 92, 169, 170

MSE Mean-Squared Error function. 36

M_{XY} Mapping function of domain X towards domain Y. 48, 49, 51, 87

M_{YX} Mapping function of domain Y towards domain X. 48, 49, 51, 87

S_0 First Stokes parameter. XV, XVIII, XIX, XXI, 17, 18, 20–22, 82–87, 92, 103, 108, 118, 125, 130

S_1 Second Stokes parameter. XV, XVIII, XIX, XXI, 17, 18, 20–22, 82–84, 86, 87, 92, 103, 108, 118, 125, 130

S_2 Third Stokes parameter. XV, XXI, 17, 18, 20–22, 82, 87, 92, 103, 108, 125, 130

S Stokes vector. XV, XXI, 17–20, 83, 87, 92

TP Number of True Positives. 47

W_r Width of the rectified feature map. 40

$W_{\mathbf{polar}}$ Width of the polarimetric images. 120

W Width of the input of a convolution layer. 37–40

Id Identity matrix. 170

Ker Kernel of a space. 170

α_i Orientation's angle of the polarizer. XV, 17–19, 82

$\bar{r}$ Recall values that exceed a given one. 47

β Down-weighting factor of the Focal Loss. 46

°**C** Degree Celsius. 30

°**F** Degree Fahrenheit. 30

ℓ_1 Manhattan distance. 49

ℓ_2 Euclidean distance. 87

η Hyper-parameter that controls the influence of the reconstruction term. 49, 51, 88

$\hat{I}_0$ Generated Intensity corresponding to the polarizer oriented at 0°. 87

$\hat{I}_{90}$ Generated Intensity corresponding to the polarizer oriented at 90°. 87

$\hat{I}$ Generated Intensities vector. 87

$\hat{S}_0$ Generated First Stokes parameter. 87

$\hat{S}_1$ Generated Second Stokes parameter. 87

$\hat{S}_2$ Generated Third Stokes parameter. 87

$\hat{S}$ Generated Stokes vector. 87

κ Learning rate. 37, 121

λ Wavelength of an electromagnetic wave. XVI, 30, 31

$\langle . \rangle$ Temporal mean of a signal. 17

$\lfloor . \rfloor$ Floor operation. 38

$\mathbb{E}$ Expectation of a distribution. 48, 49, 87

$\mathbb{R}$ Real numbers. 87, 88

$\mathcal{C}_1$ Calibration constraint. XXI, 87, 91, 92

$\mathcal{C}_2$ First admissibility constraint. XXI, 87, 91, 92

$\mathcal{C}_3$ Second admissibility constraint. XXI, 87, 91, 92

$\mathcal{L}_{\mathcal{C}_1}$ Calibration constraint related loss. XVIII, XXI, 87, 88, 92

$\mathcal{L}_{\mathcal{C}_2}$ Admissibility constraint related loss. XVIII, XXI, 87, 88, 92

$\mathcal{L}_{\mathbf{CycleGAN}}$ CycleGAN loss. 49, 88

$\mathcal{L}_{\mathbf{GAN}}$ GAN loss. 48, 49

$\mathcal{L}_{\mathbf{classification}}$ Classification loss function of the network. 45

$\mathcal{L}_{\mathbf{final}}$ Polarimetric CycleGAN loss. 88

$\mathcal{L}_{\mathbf{reco}}$ Reconstruction loss. 49

$\mathcal{L}_{\mathbf{regression}}$ Regression loss function of the network. 45

$\mathcal{L}$ Loss function of the network. 45

μ Hyper-parameter that controls the admissibility constraint. 88

ν Hyper-parameter that controls the calibration constraint. 88

ω Pulsation of an electromagnetic field. 16

ϕ Angle Of Polarization. XV, XVIII, 20–22, 83–86, 103, 108

π Archimedes' constant, approximately equal to 3.14159. 20, 85

ψ_x Phase of the magnetic field. 16, 17

ψ_y Phase of the electric field. 16, 17

ψ Phase of an electromagnetic field. 16

ρ Degree Of Polarization. XV, XVIII, 20–22, 83–86, 103, 108

σ Gaussian weight of the soft-NMS algorithm. 45, 118

$\tilde{A}$ Pseudo-inverse of the calibration matrix of the linear polarimetric camera. 20, 169

ϖ Weight factor of the classification loss function. 45

$\vec{E}(t)$ Electromagnetic field. 16

$\vec{u_x}$ horizontal axis defining the orthonormal basis. 16

$\vec{u_y}$ Vertical axis defining the orthonormal basis. 16

$\vec{z}$ Direction of propagation of the electrical field. 16

ζ Weight factor of the regression loss function. 45

c Speed of light, approximately equal to 299800 $km.s^{-1}$. 31

d Distance of an object to the LiDAR emitter. 31

e Prediction error of a neuron. 36, 37

f_p Pooling function. 40

k Wave number. 16

mAP mean Average Precision metric. 47, 93, 102, 105, 111, 112, 125–127, 129, 132

p_X Distribution of domain X. 48, 49

p_Y Distribution of domain Y. 49, 87

p_Z Distribution of domain Z. 48

p_c Convolution padding. XVII, 38–40

p_s Precision at a given recall. 47

p_t Vector containing the evaluated distance of the estimated probabilities to the ground truth. 45, 46

$p_{\mathbf{interp}}$ Maximum precision for which recall is greater than a given one. XVII, 47, 48

p Vector of the estimated probabilities of belonging to each class. 45, 46

r Recall level. XVII, 47, 48

s Convolution stride. XVII, 38–40

t_0 Emission time of the light pulse. 31

tr Reception time of the most important echo of the reflected light pulse. 31

w_0 Bias or threshold of an artificial neuron. 34

w_p Pooling window width. 40

w Weights associated to the entry of an artificial neuron. XVI, 34, 36–39

x Input of an artificial neuron. XVI, 34, 36–39

$y_{\mathbf{pooling}}$ Output of a pooling layer. 40

$y_{\mathbf{rectified}}$ Output of a non-linear rectification. 39, 40

y Output of an artificial neuron. 34, 36–38

z Ground truth label. 36, 45, 46

a Offset to the cyan → magenta axis of the CIE Lab color space. XVI, 27, 28

B Blue axis of the RGB color space. XV, XVIII, 24, 26, 29, 30, 77, 78

b Offset to the blue → yellow axis of the CIE Lab color space. XVI, 27, 28

Cb Blue chrominance axis of the YCrCb color space. XVI, 29, 30

Cr Red chrominance axis of the YCrCb color space. XVI, 29, 30

G Green axis of the RGB color space. XV, XVIII, 24, 26, 29, 30, 77, 78

H Hue axis of the HSV color space. XVI, 25, 26

L Luminance axis of the CIE Lab color space. XVI, 27

R Red axis of the RGB color space. XV, XVIII, 24, 26, 29, 30, 77, 78

S Saturation axis of the HSV color space. XVI, 25, 26

V Value axis of the HSV color space. XVI, 25, 26

Y Luma axis of the YCrCb color space. XVI, 29, 30

Abstract

Autonomous vehicles and Advanced Driver-Assistance Systems (ADAS) have shown outstanding improvements these past few years thanks to a more accurate and reliable road scene analysis. These enhancements are mostly due to the emergence of deep learning, enabling a very accurate road object detection. However, even if nowadays autonomous vehicles can be found in several countries, they show limits when then visibility is altered. Non-conventional modalities are the best solution to overcome this limitation, thanks to their ability to see beyond human vision, yet without being robust to any test. In this thesis, we aim to address this challenge by using polarimetric imaging, describing objects by their physical properties invariant to visibility changes.

In this thesis, we first give the background knowledge on multimodality. The polarization formalism is detailed, followed by the color models, aiming to represent human trivariant vision. Infrared imaging and Light Detection And Ranging (LiDAR) point clouds are explained as they play an important role in autonomous navigation. The theory behind deep learning, especially regarding convolutional neural networks is then addressed. Among them, the object detectors are described as they play an important role in this thesis. An overview of Cycle-Consistent Generative Adversarial Networks (CycleGAN) is also given.

The literature review comes next to bridge the gap between these two fields to understand how they work together to enable autonomous navigation. The applications of polarimetric imaging are drawn up, as well as the limitations of current non-conventional modalities used in autonomous systems. It gives an intuition on the use of polarimetric features to enhance road scene analysis in complex situations. The different object detectors are also presented, followed by the multimodal fusion architectures.

Afterwards, the datasets constituted to carry out the needed experiments are then presented, including the acquisition process, their content and their labels, as well as the established polarimetric data formats. The designed CycleGAN generating polarimetric images under constraints from Red, Green, Blue (RGB) ones is also sketched. Empirical evidence show that the polarimetric equivalent of the benchmarks of the literature are an asset to enhance road object detection.

Finally, different experiments are conducted to demonstrate that polarimetric features learnt in good weather conditions are still able to efficiently detect road objects under fog. From these experiments, a multimodal color-based and polarimetric fusion scheme is designed. Not only it enables to enhance object detection under fog, but it generalizes the obtained results to several visibility conditions, including various densities of fog and dense rain.

Résumé

Les véhicules autonomes et les ADAS ont connu des améliorations remarquables ces dernières années grâce à une analyse plus précise et plus fiable des scènes routières. Ces améliorations sont principalement dues à l'émergence de l'apprentissage profond, qui permet une détection très précise des obstacles routiers. Cependant, même si l'on trouve aujourd'hui des véhicules autonomes dans plusieurs pays, ils atteignent leurs limites lorsque la visibilité est altérée. Les modalités non conventionnelles sont la meilleure solution pour surmonter ces limitations, grâce à leur capacité à voir au-delà de la vision humaine, sans pour autant être robustes à tout test. Dans cette thèse, nous répondons à ce défi en utilisant l'imagerie polarimétrique, décrivant les objets par leurs propriétés physiques, invariantes aux changements de visibilité.

Dans cette thèse, le concept de multimodalité est d'abord introduit. Le formalisme de polarisation est détaillé, suivi par les modèles colorimétriques, visant à représenter la vision trivariante humaine. L'imagerie infrarouge et les nuages de points LiDAR sont également abordés, car ils jouent un rôle important dans la navigation autonome. La théorie de l'apprentissage profond, notamment les réseaux de neurones convolutifs, est ensuite abordé. Parmi eux, les détecteurs d'objets sont décrits car ils jouent un rôle important dans cette thèse. Un aperçu du CycleGAN est également donné.

L'état de l'art vient ensuite faire le pont entre ces deux domaines afin de comprendre comment leur combinaison permet la navigation autonome. Les applications de l'imagerie polarimétrique sont parcourures, ainsi que les limites des modalités non conventionnelles actuellement utilisées dans les systèmes autonomes. Cela permet d'avoir une intuition sur l'utilisation des caractéristiques polarimétriques pour améliorer l'analyse des scènes routières en situations complexes. Les différents détecteurs d'objets sont également présentés, suivis des architectures de fusion multimodale.

Ensuite, les jeux de données constitués pour réaliser les expériences nécessaires sont présentés, y compris le processus d'acquisition, leur contenu et leurs étiquettes, ainsi que les formats de données polarimétriques établis. La méthode conçue pour générer des images polarimétriques sous contraintes à partir des images RGB est également esquissée. Les preuves empiriques montrent que l'équivalent polarimétrique des bases de données repères de la littérature permet d'améliorer la détection d'obstacles routiers.

Enfin, différentes expériences sont menées pour démontrer que les caractéristiques polarimétriques apprises lorsque la visibilité est bonne peuvent décrire les obstacles routiers dans le brouillard. Ces expériences ont permis de concevoir un schéma de fusion multimodal basé sur l'imagerie couleur et polarimétrique. Ce schéma permet d'améliorer la détection d'objets sous le brouillard et étend les résultats obtenus à d'autres situations, y compris diverses densités de brouillard et la pluie dense.

"Une danseuse souffre en silence."

Remerciements

L'heure est maintenant aux remerciements, que j'écris avec un peu de recul pour, contrairement à ma soutenance de thèse, n'oublier personne.

Je tiens tout d'abord à remercier les membres du jury qui ont accepté de rapporter et d'examiner mon travail de thèse. Vos commentaires que ce soit dans vos rapports ou à l'oral pendant la soutenance m'ont fait très chaud au cœur et m'ont permis de prendre du recul sur mes travaux et de les voir à leur juste valeur, chose que j'ai toujours eu du mal à faire.

Je voudrais également remercier Samia, qui m'a dirigée tout au long de cette thèse et qui m'a appris une qualité que, selon moi, tout scientifique devrait posséder, à savoir que tout scientifique doit savoir s'entourer de profils complémentaires au sien pour aller encore plus loin dans la recherche. Je tiens également à la remercier pour l'opportunité. En effet, en sortant d'école d'ingénieur je n'aurais jamais misé sur moi, ce que Samia m'a appris à faire depuis les PAO que l'on a fait ensemble. Je la remercie pour les bons moments passés ensemble, durant les acquisitions par exemple (elle se souviendra de Renée ou de la cité de Saint-Étienne-du-Rouvray) et pour n'avoir jamais compté le temps de repos qu'elle m'accordait quand je me sentais fatiguée.

Je remercie également Stéphane et Fabrice pour leur co-encadrement tout au long de cette thèse. Stéphane, qui m'a donné l'envie de continuer dans cette voie avant d'être diplômée et avec qui j'ai beaucoup appris. Fabrice également, pour sa rigueur et sa capacité à m'avoir poussée à toujours me poser les bonnes questions sur mes résultats obtenus.

Quant à mes collègues, je vais commencer par tous les remercier, au cas où j'oublierai l'un d'entre eux. Je pense tout d'abord à tous ceux qui ont facilité le travail et les démarches administratives que j'ai eu à effectuer durant ces trois années. Sandra et Brigitte, qui maintiennent le navire du LITIS à flots par leur efficacité et leur générosité quant au temps qu'elles accordent aux autres. Je remercie Elsa pour ses prêts de matériel, toujours dans la bonne humeur. Je n'oublie pas Gilles, qui, malgré son emploi du temps surchargé, est là pour prêter une oreille attentive et bienveillante. Jean-François, toujours dévoué pour aider son prochain qui a des problèmes informatiques en tout genre (même quand il s'agit de recocher l'option pour avoir internet en filaire). Laurent, pour prendre les problèmes des doctorants toujours très à cœur et avec qui j'ai eu le plaisir de travailler en tant que responsable des doctorants. Benoist et Béatrice, pour leur disponibilité afin que nous puissions utiliser au mieux les services du Criann. Gérôme, pour avoir été un super coach sportif autant que la pandémie l'a permis.

Je tiens également à remercier tous les professeurs aux côtés de qui j'ai effectué mes missions enseignement. Benoît et Gilles (encore une fois) pour m'avoir appris la

rigueur qu'il fallait pour enseigner et l'empathie vis à vis des élèves nécessaire pour être professeur. Romain et Clément, pour avoir toujours été réactifs et arrangeants quant aux horaires de cours mais aussi pour avoir toujours tenu compte de mon avis et de mon ressenti vis à vis des étudiants. Je remercie encore une fois Samia pour sa disponibilité quand j'avais besoin d'une explication sur un de ses cours. Alexandrina, pour m'avoir permis de découvrir un autre aspect de la théorie de l'information. Je voudrais aussi remercier tout particulièrement Nicolas, pour son implication en tant que chef de département ASI (je maintiens cette appellation), que j'appréciais déjà en tant qu'étudiante du département et qui s'est renforcée en tant qu'apprentie professeur qui a travaillée à ses côtés. Je remercie également les autres intervenants avec qui j'ai pu travailler, Ismaïla, Anis et Hamza, avec qui il a toujours été possible de s'entraider, de s'arranger et d'échanger dans la bonne humeur et à qui je souhaite une très bonne continuation.

Je remercie plus personnellement mes collègues et amis, les autres doctorants et post-doctorants du laboratoire. Je pense en premier à Jean-Baptiste, à côté de qui j'ai beaucoup appris personnellement et qui est une des personnes les plus intéressantes qu'il m'ait été donné de rencontrer dans ma vie[1]. J'accorde également une mention spéciale à Franco, qui est toujours là pour véhiculer sa bonne humeur, même s'il devrait savoir que la machine pour payer en carte n'est pas un téléphone. Je remercie également Cyprien et Ismaïla, avec qui nous avons formé la ligne du fond de la salle 5, se penchant sur des sujets scientifiques très sérieux, tels que le tracé de la frontière entre les pays de la tomate et les pays de la patate. Je pense aussi à Mathieu, pour ses histoires toutes plus incroyables les unes que les autres que j'ai toujours adoré écouter. Sandratra, pour être toujours présent quand il s'agit de mettre en place de plans bouffe. Marwa, pour sa conversation, sa bonne humeur et sa façon de suivre les calculs à la trace sur le Criann. I also would like to thank Rosana, who is always cheerful no matter what. Je remercie également Maël, Enrique, Djamila, Flavie, Linlin, Florencia, Caterine, Maria et tous ceux que j'ai pu oublier et avec qui j'ai partagé, ne serait-ce qu'un moment qui m'a donné le sourire.

Je pense aussi à mes amis en dehors du LITIS qui m'ont permis de décrocher de ma thèse quand j'en avais besoin. Malak, je sais que tu es toujours prête pour danser sur notre playlist[2] et pour tenir le bureau des plaintes. May, depuis la Libye tu es toujours là, à te retrouver par hasard dans les pays dans lesquels je suis, pour le meilleur et pour le pire[3]. Claire1 et Claire2[4], pour les instants de solidarité, de jeux de société (et de jeux vidéo pour Claire2) et la bouffe partagée et ce depuis ASI. Ingrid, que j'ai appris à vraiment connaître plus récemment mais qui m'a permis de me changer les idées. Je vais maintenant remercier en vrac toutes les personnes qui ont partagé un bout de mon chemin, qui sont encore là pour certains et que la vie a fait prendre un autre chemin pour d'autres, mais qui ont tous laissé un morceau d'eux qui a contribué à la personne que je suis, laissant un impact positif dans ma vie. Guillaume, Ryme, Rania, Bahia, Myriam, Kévin, Eliot, Mélodie, Constance, Alexandre, Robin, Silvia,

[1] L'incubaclodobus c'est l'avenir, ne vous en déplaise.

[2] Le contenu étant trop honteux, pour notre réputation je ne le dévoilerai pas ici.

[3] Même si la plupart du temps c'est pour commenter Un Dîner Presque Parfait.

[4] On a bien compté dans un langage qui se respecte, Claire0 c'est moi.

Figure 1: Photo pas assez contractuelle.

Marie-Joëlle, Bouali, Andrew et tous les autres que j'oublie.

Maintenant vient le tour de mon cercle plus proche, ma famille. Papa, maman, je souhaite vous remercier de m'avoir toujours soutenue dans tous mes projets. Une thèse ça peut être long mais vous avez toujours été présents sans compter, avec tout ce que sans compter implique. Je voudrais remercier en particulier ma sœur Andréa, qui a toujours été là du début à la fin pour m'apporter du soutien moral. Nos appels du vendredi soir (qui se sont transformés en plusieurs fois par semaine), pour parler de tout et de rien et pour tous ces conseils qui m'ont beaucoup aidée au moment de chercher mon travail. Je te dédicace une grande partie de cette thèse qui, surtout avec la pandémie, n'aurait pas pu avancer aussi vite sans toi. Je remercie également ma sœur Wendy pour sa bonne humeur à chacun de nos appels et que j'espère pouvoir amener un jour à la fête du ventre à Rouen. Je remercie également Anthony, mon frère, pour toutes les discussion et les rires liés à nos souvenirs d'enfance.

Enfin, je tiens à remercier particulièrement Benjamin. Lui qui a commencé par être l'ami d'amis, puis le collègue avec qui je discutais tous les matins sans connaître le nom trois semaine durant et encore l'installateur de CUDA sur mon ordinateur. Je te remercie tout d'abord pour tout ça. Je te remercie également pour la vidéo de Pokémon, sans laquelle on tu n'aurais peut-être jamais été quelqu'un d'autre que l'ami d'ami ou le collègue qui installe CUDA. Durant toute cette thèse tu as été mon repère et parfois j'ai eu l'impression que tu m'a donné plus que ce que je pouvais te donner en retour, par ton sang froid, imperturbable à toute épreuve et par ta maîtrise technique de l'informatique que je n'atteindrai sans doute jamais. Plus qu'un compagnon, tu es le meilleur ami, le deuxième joueur aux jeux vidéos et aux jeux de sociétés, le partenaire de confinement et le créateur de challenges pour la cuisine évitant tout un tas d'ingrédients. Tu es un rocher aux fondations bien ancrées qui ne bouge pas quelle que soit la tempête et qui a réussi à me retenir à chaque fois que j'allais chavirer et je souhaite à tout le monde d'avoir un tel rocher dans sa vie.

En bonus, je tiens à remercier celui qui ne regarde pas l'aspect physique, ni l'aspect mental d'ailleurs. Celui qui sait démontrer son affection par des coups de tête ou de truffe, et qui plaidera allégeance tant que la gamelle sera pleine. Voir illustration de Tigrou en Figure 1 pour apprécier cette beauté suprême.

Contents

List of Figures

List of Tables

Introduction

Context and motivations

The 1960s mark the emergence of works on autonomous navigation. These projects are materialized in 1977, with the construction of the first semi-automated car [5]. A major milestone is achieve in 1995 with NavLab [6], completing the first mostly autonomous drive of more than four thousands kilometers. This breakthrough opened the door to several autonomous driving challenges, including the DARPA grand-challenge [7] in 2002, aiming to evaluate the performances of autonomous vehicles. Autonomous navigation has boomed since 2010s with the emergence of deep learning, which has shown outstanding performances at this task. Nowadays, the SAE classification [8], a six-levels classification system, is used to give indications on the level of automation of the different vehicles. The six levels, ranging from fully manual to fully automated systems, are illustrated in Figure 2[5]. Several autonomous vehicles can be found all around the world with different autonomy levels. Among which, we can cite the Waymo[6] autonomous car, a systems that operates at level 4 autonomy, the Tesla Autopilot system[7], that enables a level 2 autonomy or the Rouen Normandy autonomous lab[8], that provides a level 3 autonomy in a restricted area.

However, the above-mentioned systems find limitations in complex situations such as adverse weather conditions [9, 10]. Enabling autonomous navigation when the visibility is altered is one of the grand challenges that need to be addressed to reach the level 5 autonomy. Current autonomous systems use non-conventional modalities as they offer the best results to palliate this limitation. Infrared imaging is used to enhance pedestrians, cyclists and motorcyclists detection [11] and enables a detection of road users at a larger range under fog, rain and snow [12]. However, it shows limits under heavy fog or rain, as they can modify the thermal footprint of objects [13]. Radio Detection And Ranging (Radar) is another non-conventional modality used to enhance road object detection in adverse weather conditions [14], yet without guarantee optimal accuracy since it is affected by thermal noise [15]. LiDAR is mainly used to enhance road scene analysis under low visibility [16] but come up with a lot of noise due to snowflakes or drops misinterpretations [17].

To provide an accurate road scene analysis when the visibility is altered, other non-conventional modalities need to be explored. They would come up with further

[5]Source: `https://newsroom.intel.com/`

[6]`https://waymo.com/`

[7]`https://www.tesla.com/autopilot`

[8]`https://www.rouennormandyautonomouslab.com/`

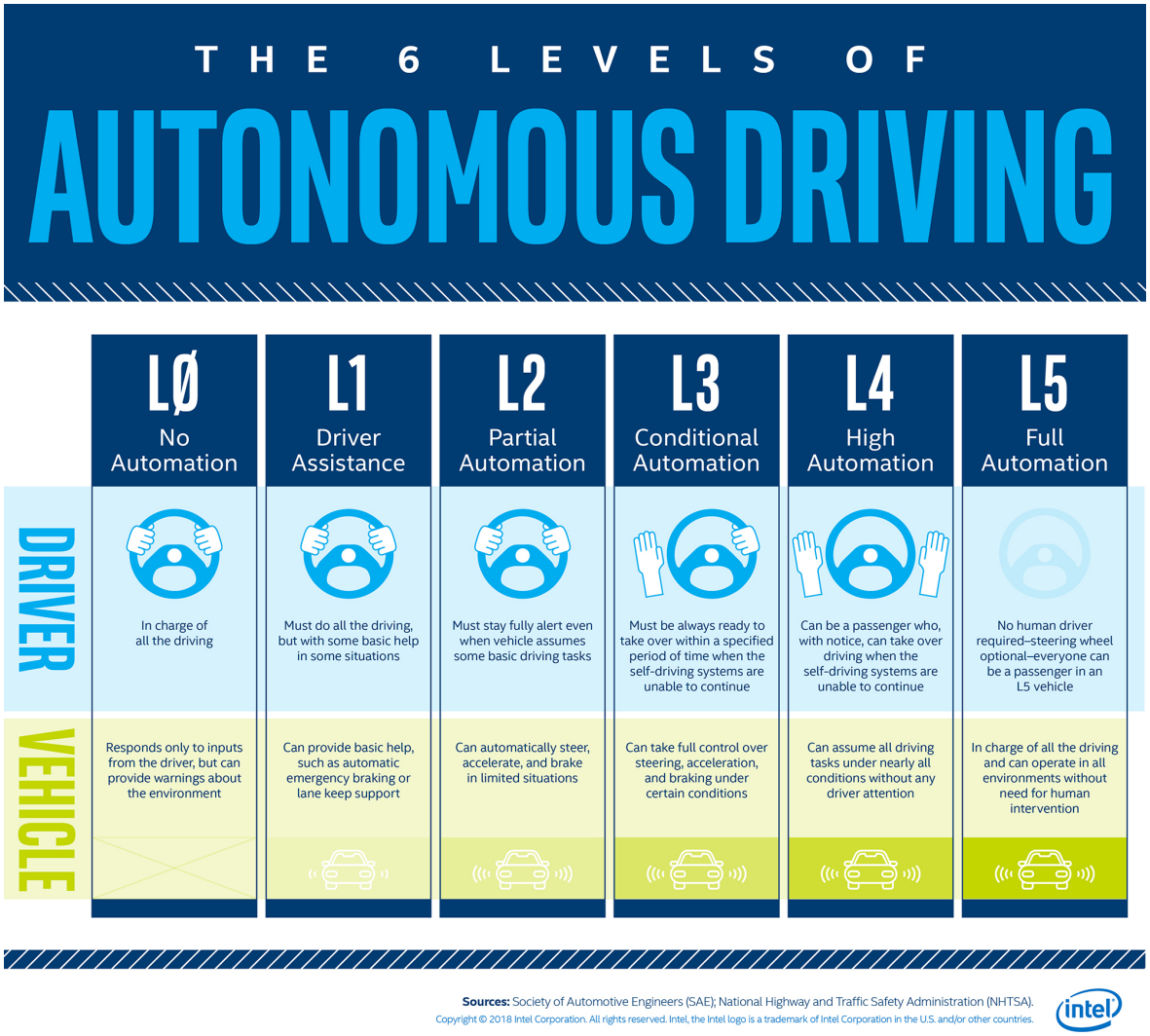

Figure 2: The six automation levels of autonomous systems.

information, more robust to external changes. In this thesis, we decided to focus our research on polarimetric imaging to enhance the detection of the different road users. This non-conventional modality characterizes an object from its physical features, invariant to strong illuminations or low visibility [18].

Polarimetric imaging

Polarization is a rich modality describing the light wave reflected by the object it impinges on. As a matter of fact, when an unpolarized light wave is being reflected by an object, it becomes partially polarized. In other terms, it travels in a well determined direction [18]. The direction in which the wave is travelling depends on the surface's material and is invariant to the visibility conditions. The reflected wave can be described by a set of measurable parameters called the Stokes vector.

Polarimetric imaging consists in capturing polarimetric intensities of the scene, in order to recover the three Stokes parameters, giving direct information on the physical

properties of each pixel. Thanks to this property, polarimetric images find applications in a wide range of fields. Polarimetric medical imaging use this principle to distinguish defective cells from healthy ones from their physical composition [19], while they are indiscernible to the naked eye. Polarimetric images also prove to be a real added value to dehaze a scene [20], since physical properties of haze particles are different from the ones of the rest of the scene. More recently, this non-conventional modality has shown outstanding results to enhance autonomous navigation, by providing an accurate depth map of the scene [21] or by reducing the false alarm rate when detecting cars [22]. These results, as well as the application domains of polarimetric imaging, are encouraging towards its use to overcome adverse weather conditions in road scenes.

Object detection

Object detection has gained recent interest, mostly due to the expansion of deep learning frameworks. This task forms part of a wide research field, resulting in more and more accurate models. It takes an important place in road scene analysis since it provides direct information on the position and the nature of the different objects of the scene. These information are needed in autonomous systems' decision pipelines and paramount to guarantee road users' safety.

Nowadays, there are several paradigms to perform object detection, all based on Convolutional Neural Networks (CNN) widely used in the computer vision field. Two stages detection is the first formulation of this problem and still leads to off-the-shelf detectors. It consists in first finding Region Of Interest (ROI) in images processed by CNN. These ROI are then regressed to better fit the object localization and classified to find the nature of the object. Several architectures, such as Faster Region Based Convolutional Neural Networks (R-CNN) [23] and its derivatives [24] are milestones that still regularly outperform architectures of the state of the art.

Single shot detectors are another paradigm that is very popular since it enables object detection in real time. It consists in performing the classification and the regression steps simultaneously from the image processed by CNN. The ROI proposed by two stages detectors are replaced by a grid with each cell generating a fixed number of proposition in You Only Look Once (YOLO) [1] and anchor boxes priors in Single Shot MultiBox Detector (SSD) [2] and RetinaNet [3]. A recent approach [25] gets rid of priors by formulating object detection as a direct set prediction problem.

Another paradigm sees object detection as the prediction of a set of key points. It produces heatmaps giving the location of points of interest, mainly the top left and the top right corners of the bounding box locating the object [26], sometimes enriched with the center of the bounding box [27]. It can also predict four extreme points (top most, bottom most, left most and right most) [28] or even more key points drawing the global shape of the object [29].

All these architectures are evaluated on benchmarks addressing the object detection task of several common objects at a large scale. The Microsoft Common Object in COntext (MS COCO) dataset [4] is nowadays the mostly used since it contains more than 120,000 complex indoor and outdoor scenes with more than 880,000 instances from 91 classes. Even if these pipelines are tested on color-based RGB images, they

can be modified to process other modalities. Hence, they are often used as basis to multimodal fusion architectures.

Multimodal fusion

As mentioned previously, multimodality is the core of road scene analysis in complex situations. To perform an efficient multimodal fusion, it is usual to answer the following questions:

What to fuse? The choice of the modalities to fuse is highly dependant of the aimed application. Infrared images are often combined with color-based images to enhance pedestrian detection in every situation [30]. LiDAR point clouds and color-based images are joint together to perform Three Dimensions (3D) road object detection [31]. On the other hand, Radar signals and color-based images are merged to increase the speed and accuracy of detections [32]. Finally, LiDAR point clouds and the map of the environment are fused to predict both the intention of the other road users as well as their 3D localization [33].

When and how to fuse? Three fusion pipelines can be used to efficiently fuse multimodal information, with their advantages and drawbacks [34]. The Early fusion scheme combines raw or pre-processed data at an early stage. While this pipeline fully exploits raw data from their joint representation and results in lighter architectures, it is inflexible to sensor replacement and not robust to sensor breakdown. The Late fusion scheme on the contrary is highly flexible since it processes each modality separately. It is therefore more robust to sensor breakdown but results in computationally expensive pipelines. Finally, the Middle fusion scheme is the best of both worlds since it first processes each modality separately and combines them halfway of the network. It leads to lighter and more flexible architectures. However, a lot of neural architecture search is required to find the optimal Middle fusion design.

As for conventional object detectors, several benchmarks aim to evaluate the performances of the different fusion schemes. The KAIST dataset [35] is often used to evaluate the pipelines aiming to enhance pedestrian detection, as it contains color-based and thermal images. Karlsruhe Institute of Technology and Toyota Technological Institute dataset (KITTI) [36] is one of the mostly used datasets, as it contains LiDAR point clouds and RGB images, and is labelled to detect various road users. The Waymo Open dataset [37] is a larger alternative to KITTI.

Contributions

In this thesis, we came up with three main contributions, which are the constitution of the first large multimodal polarimetric and color-based dataset for road object detection in several weather conditions, improvements on road scene analysis in adverse weather conditions using polarimetric imaging and a fusion pipeline for color-based and polarimetric images. Regarding the first application, our contributions can be summarized as follows:

I We propose several datasets, containing different kind of road scenes in different weather conditions, labelled for road object detection. Some acquisitions are made outdoors while some others are made in a tunnel simulating adverse weather conditions.

II We come up with six polarimetric data formats, providing well chosen combinations of polarimetric features, encoded for machine learning.

III We design a CycleGAN under physical constraints, generating polarimetric images from RGB ones, which provides a polarimetric equivalent of benchmarks for road object detection.

As for the improvements on road scene analysis in adverse weather conditions, our contributions are:

I We demonstrate that off-the-shelf object detectors of the literature can be used to process efficiently polarimetric road scenes.

II We show that polarimetric features learnt in good weather conditions are still valid to detect road objects under fog unlike color-based features.

Finally, concerning the fusion pipeline for color-based and polarimetric images, our contributions are:

I We compare several fusion pipelines, including an Early fusion pipeline and five designed filters used for Late fusion purposes, to find the most effective to enhance road scene analysis in adverse weather.

II We generalize the results obtained under fog to dense rain and to several fog densities and find the limitations of polarimetric features under very dense fog.

The different contributions have been published in the following papers:

- Journal papers

  1 Rachel Blin, Samia Ainouz, Stéphane Canu, Fabrice Mériaudeau. "*The PolarLITIS dataset: road scenes under fog*", Transactions on Intelligent Transportation Systems (T-ITS), 2021.

  2 Rachel Blin, Samia Ainouz, Stéphane Canu, Fabrice Mériaudeau. "*Road scene analysis under fog: towards an optimal conventional-non-conventional fusion scheme*", submitted to Transactions on Intelligent Vehicles (T-IV), 2021.

  3 Cyprien Ruffino, Rachel Blin, Samia Ainouz, Gilles Gasso, Romain Hérault, Fabrice Mériaudeau, Stéphane Canu. "*Generating physically admissible polarimetric images as data augmentation for road-scene analysis*", Currently in preparation.

- Conference papers

1 Rachel Blin, Samia Ainouz, Stéphane Canu, Fabrice Mériaudeau. "*Adapted learning for Polarization-based car detection*", Quality Control by Artificial Vision (QCAV), 2019.

2 Rachel Blin, Samia Ainouz, Stéphane Canu, Fabrice Mériaudeau. "*Road scenes analysis in adverse weather conditions by polarization-encoded images and adapted deep learning*", IEEE International Conference on Intelligent Transportation Systems (ITSC), 2019.

3 Rachel Blin, Samia Ainouz, Stéphane Canu, Fabrice Mériaudeau. "*A new multimodal rgb and polarimetric image dataset for road scenes analysis*", Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshops), 2020.

4 Rachel Blin, Samia Ainouz, Stéphane Canu, Fabrice Mériaudeau. "*Multimodal Polarimetric And Color Fusion For Road Scene Analysis In Adverse Weather Conditions*", IEEE International Conference on Image Processing (ICIP), 2021.

5 Rachel Blin, Samia Ainouz, Stéphane Canu, Fabrice Mériaudeau. "*Road scene analysis: A study of polarimetric and color-based features under various adverse weather conditions*", submitted to IEEE International Conference on Computer Vision Workshops (ICCV Workshops), 2021.

Outline

This thesis is divided into six chapters as follows:

- Chapter 1 introduces the concept of multimodality by presenting polarimetric imaging, color-based imaging, infrared imaging and LiDAR point clouds.
- Chapter 2 gives the background knowledge on Deep Learning frameworks, including object detectors and CycleGAN.
- Chapter 3 reviews the applications of polarimetric images in the literature and the role of multimodality in autonomous driving. It also describes the different object detectors and the multimodal fusion architectures.
- Chapter 4 details the different datasets constituted to carry out the experiments of this thesis.
- Chapter 5 demonstrates the role of polarimetric imaging to enhance road scene analysis under fog.
- Chapter 6 presents the color-based and polarimetric fusion pipelines that improve road object detection in a wide range of adverse weather conditions.
- Finally, we draw conclusions about our work and present the future perspectives.

Introduction

Contexte et motivations

Les années 1960 marquent l'émergence de travaux sur la navigation autonome. Ces projets se concrétisent en 1977, avec la construction de la première voiture semi-automatique [5]. Une étape majeure est franchie en 1995 avec NavLab [6], qui réalise le premier trajet de plus de quatre mille kilomètres en autonomie complète. Cette percée technologique a ouvert la porte à plusieurs défis de conduite autonome, dont la compétition lancée par DARPA [7] en 2002, visant à évaluer les performances des véhicules autonomes. La navigation autonome est en plein essor depuis les années 2010 avec l'émergence de l'apprentissage profond qui a montré des performances exceptionnelles pour accomplir cette tâche. Aujourd'hui, la classification SAE [8], un système de classification à six niveaux, est utilisée pour donner des indications sur le niveau d'autonomie des différents véhicules. Les six niveaux, qui vont des systèmes entièrement manuels aux systèmes entièrement automatisés, sont illustrés en Figure 3[9]. Plusieurs véhicules autonomes existent dans le monde entier avec différents niveaux d'autonomie. Parmi eux, on peut citer la voiture autonome Waymo[10], un système qui fonctionne au niveau 4 d'autonomie, le système Autopilot de Tesla[11], qui permet une autonomie de niveau 2 ou encore le laboratoire autonome Rouen Normandie[12], qui fournit une autonomie de niveau 3 dans une zone restreinte.

Cependant, les systèmes mentionnés ci-dessus montrent leurs limites en situations complexes telles que des conditions météorologiques défavorables [9, 10]. Permettre la navigation autonome lorsque la visibilité est altérée est l'un des grands défis à relever pour atteindre le niveau 5 d'autonomie. Les systèmes autonomes actuels utilisent des modalités non conventionnelles car elles permettent d'obtenir les meilleurs résultats pour pallier cette limitation. L'imagerie infrarouge est utilisée pour améliorer la détection des piétons, des cyclistes et des motocyclistes [11] et permet une détection des usagers de la route à une plus grande distance sous le brouillard, la pluie et la neige [12]. Cependant, l'imagerie infrarouge montre ses limites en cas de brouillard épais ou de pluie dense qui peuvent modifier l'empreinte thermique des objets [13]. Le Radar est une autre modalité non conventionnelle utilisée pour améliorer la détection d'obstacles routiers en conditions météorologiques défavorables [14], sans toutefois garantir une précision optimale puisqu'il est affecté par le bruit thermique [15]. Le LiDAR est prin-

[9] Source : `https://newsroom.intel.com/`

[10] `https://waymo.com/`

[11] `https://www.tesla.com/autopilot`

[12] `https://www.rouennormandyautonomouslab.com/`

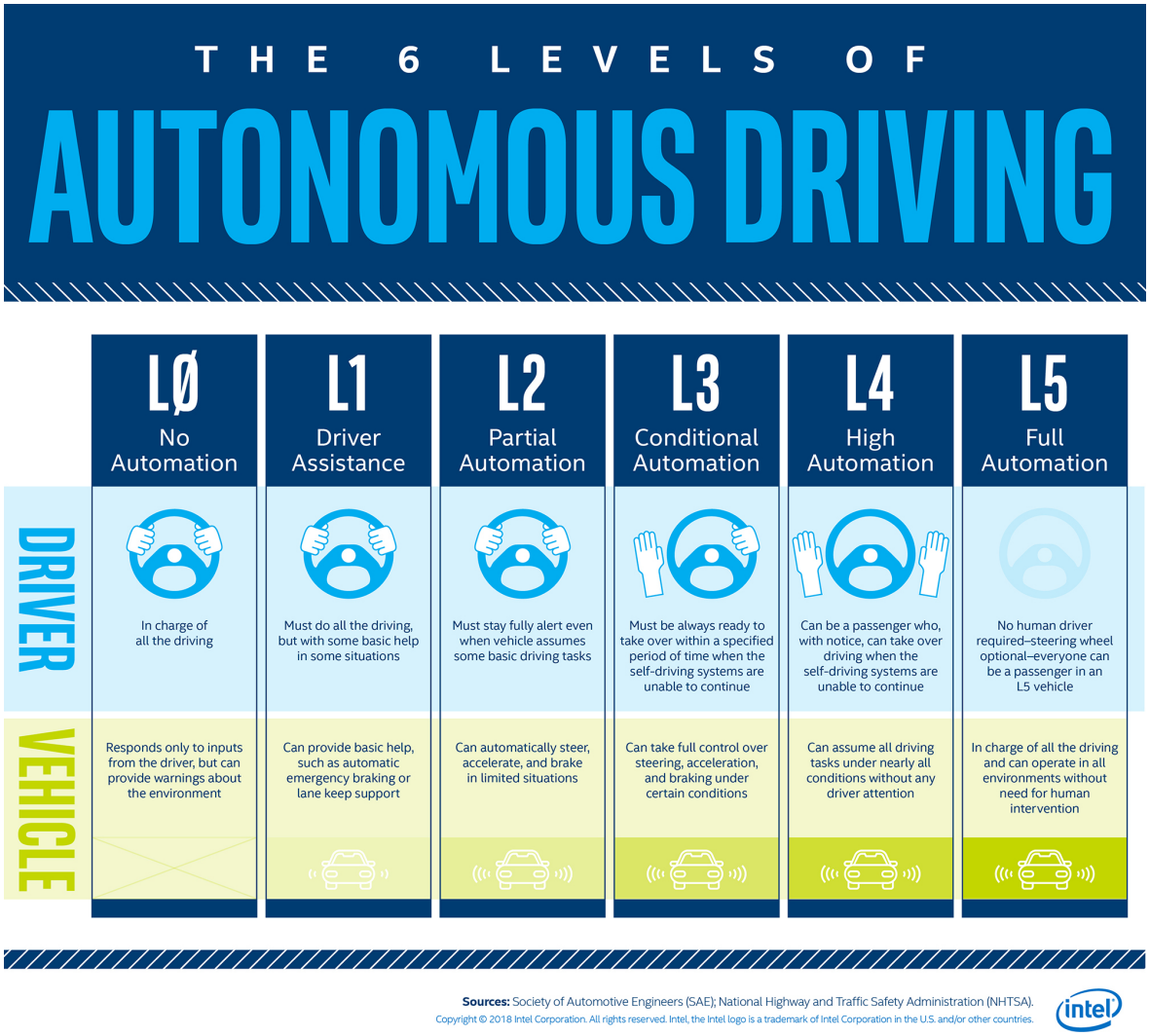

Figure 3: Les six niveaux d'autonomie des systèmes autonomes.

cipalement utilisé pour améliorer l'analyse des scènes routières par faible visibilité, mais le signal peut être très bruité à cause d'erreurs d'interprétation des flocons de neige ou des gouttes.

Pour permettre l'analyse précise d'une scène routière lorsque la visibilité est altérée, d'autres modalités non conventionnelles doivent être explorées. Elles apporteraient des informations supplémentaires, plus robustes aux changements externes. Dans cette thèse, nous avons décidé de concentrer nos recherches sur l'imagerie polarimétrique pour améliorer la détection des différents usagers de la route. Cette modalité non conventionnelle caractérise un objet à partir de ses caractéristiques physiques, invariantes aux forts éclairages ou à la faible visibilité [18].

Détection d'objet

La détection d'objets a récemment suscité l'intérêt de nombreux chercheurs, principalement en raison de l'expansion de d'apprentissage profond. Cette tâche s'inscrit dans

un large champ de recherche, résultant en des modèles de plus en plus précis. Elle occupe une place importante dans l'analyse des scènes routières car elle fournit des informations directes sur la position et la nature des différents objets de la scène. Ces informations sont nécessaires dans les pipelines de décision des systèmes autonomes et sont primordiales pour garantir la sécurité des usagers de la route.

Il existe aujourd'hui plusieurs paradigmes pour effectuer la détection d'objets, tous basés sur les réseaux neuronaux convolutifs qui sont très utilisés dans le domaine de la vision par ordinateur. La détection en deux étapes est la première formulation de ce problème et résulte encore aujoud'hui en des détecteurs très performants. Elle consiste à trouver d'abord les régions d'intérêt dans les images traitées par réseaux neuronaux convolutifs. Les coordonnées de ces régions d'intérêt sont ensuite modifiées pour mieux correspondre à la localisation de l'objet et classés pour trouver la nature de l'objet. Plusieurs architectures, comme Faster R-CNN [23] et ses dérivés [24] sont des jalons qui surpassent encore régulièrement les architectures de l'état de l'art.

Les détecteurs à coup unique sont un autre paradigme très populaire puisqu'il permet la détection d'objets en temps réel. Il consiste à effectuer simultanément les étapes de classification et de régression à partir de l'image traitée par un réseau neuronal convolutif. Les régions d'intérêt proposées par les détecteurs à deux étapes sont remplacés par une grille dont chaque cellule génère un nombre fixe de propositions dans YOLO [1] et par des boîtes d'ancrage préalablement définies dans SSD [2] et RetinaNet [3]. Une approche récente [25] se débarrasse des prédictions préalablement initialisées en formulant la détection d'objets comme un problème de prédiction d'ensemble direct.

Un autre paradigme modélise la détection d'objets comme la prédiction d'un ensemble de points clés. Il produit des cartes thermiques donnant la localisation de points d'intérêt, principalement les coins supérieurs gauche et droit de la boîte englobante localisant l'objet [26], parfois enrichie du centre de la boîte englobante [27]. Il peut également prédire quatre points extrêmes (le plus haut, le plus bas, le plus à gauche et le plus à droite) [28] ou encore plus de points clés dessinant la forme globale de l'objet [29].

Toutes ces architectures sont évaluées sur des bases de données repères, traitant la détection d'objets courants à grande échelle. Le jeu de données MS COCO [4] est aujourd'hui le plus utilisé car il contient plus de 120 000 scènes complexes d'intérieur et d'extérieur avec plus de 880 000 instances issues de 91 classes. Même si ces pipelines sont testés sur des images en couleur, ils peuvent être modifiés pour traiter d'autres modalités. Ils sont donc souvent utilisés comme base pour les architectures de fusion multimodale.

Fusion multimodale

Comme mentionné précédemment, la multimodalité est au cœur de l'analyse des scènes routières en situations complexes. Pour réaliser une fusion multimodale efficace, il est d'usage de répondre aux questions suivantes :

Quoi fusionner ? Le choix des modalités à fusionner dépend de l'application visée. Les images infrarouges sont souvent combinées avec des images couleur pour améliorer

la détection des piétons dans toutes les situations [30]. Les nuages de points LiDAR et les images couleur sont combinés pour effectuer la détection 3D d'obstacles routiers [31]. D'autre part, les données Radar et les images couleur sont fusionnées pour augmenter la vitesse et la précision des détections [32]. Enfin, les nuages de points LiDAR et la carte de l'environnement sont fusionnés pour prédire à la fois l'intention des autres usagers de la route et leur localisation 3D [33].

Quand et comment fusionner ? Trois pipelines de fusion peuvent être utilisés pour fusionner efficacement des informations multimodales, chacune ayant des avantages et des inconvénients [34]. Le schéma de fusion précoce combine les données brutes ou prétraitées à un stade précoce. S'il exploite pleinement les données brutes à partir de leur représentation conjointe et permet d'obtenir des architectures plus légères, il n'est pas facilement modifiable si un capteur a besoin d'être remplacé et n'est pas robuste aux pannes de ces derniers. Le schéma de fusion tardive, au contraire, est très flexible puisqu'il traite chaque modalité séparément. Il est donc plus robuste aux pannes de capteurs, mais entraîne des pipelines coûteux en calcul. Enfin, le schéma de fusion intermédiaire est le meilleur des deux mondes puisqu'il traite d'abord chaque modalité séparément et les combine à mi-chemin du réseau. Il permet d'obtenir des architectures plus légères et plus flexibles. Cependant, il est nécessaire d'effecture de la recherche d'architecture neuronale pour trouver l'architecture de fusion intermédiaire optimale.

Comme pour les détecteurs d'objets conventionnels, plusieurs bases de données repères servent à évaluer les performances des différents schémas de fusion. Le jeu de données KAIST [35] est souvent utilisé pour évaluer les pipelines visant à améliorer la détection des piétons, car il contient des images thermiques et des images couleur. Le jeu de données KITTI [36] est l'un des plus utilisés, car il contient des nuages de points LiDAR et des images RGB, et est étiqueté pour détecter divers usagers de la route. Le jeu de données Waymo Open [37] est une alternative à KITTI, contenant plus d'images.

Contributions

Dans cette thèse, nous avons apporté trois contributions principales, qui sont la constitution du premier grand jeu de données multimodales polarimétriques et basées sur la couleur pour la détection d'obstacles routiers dans plusieurs conditions météorologiques, des améliorations sur l'analyse de scènes routières dans des conditions météorologiques dégradées en utilisant l'imagerie polarimétrique et un pipeline de fusion pour les images couleur et polarimétriques. En ce qui concerne la première application, nos contributions peuvent être résumées comme suit :

I Nous proposons plusieurs jeux de données, contenant différents types de scènes routières dans différentes conditions météorologiques et étiquetées pour la détection d'obstacles routiers. Certaines acquisitions sont faites en extérieur tandis que d'autres sont réalisées dans un tunnel simulant des conditions météorologiques défavorables.

II Nous proposons six formats de données polarimétriques, chacun proposant des combinaisons bien choisies de caractéristiques polarimétriques, encodées pour l'apprentissage automatique.

III Nous concevons un CycleGAN sous contraintes physiques, générant des images polarimétriques à partir d'images RGB, et fournissant ainsi un équivalent polarimétrique des bases de données repères utilisées pour la détection d'obstacles routiers.

Quant aux améliorations apportées à l'analyse de la scène routière en conditions météorologiques dégradées, nos contributions sont les suivantes :

I Nous démontrons que les détecteurs d'objets disponibles dans la littérature peuvent être utilisés pour traiter efficacement les images polarimétriques de scènes routières.

II Nous montrons que les caractéristiques polarimétriques apprises dans de bonnes conditions météorologiques sont utilisables pour détecter des obstacles routiers dans le brouillard, contrairement aux caractéristiques basées sur la couleur.

Enfin, concernant le pipeline de fusion pour les images couleur et polarimétriques, nos contributions sont les suivantes :

I Nous comparons plusieurs pipelines de fusion, dont un pipeline de fusion précoce et cinq filtres conçus à des fins de fusion tardive, afin de trouver le plus efficace pour améliorer l'analyse des scènes routières en conditions météorologiques dégradées.

II Nous étendons les résultats obtenus sous le brouillard à la pluie dense et à plusieurs densités de brouillard et trouvons les limites des caractéristiques polarimétriques sous un brouillard très dense.

Les différentes contributions ont été publiées dans les articles suivants :

- Articles de revues

  1 Rachel Blin, Samia Ainouz, Stéphane Canu, Fabrice Mériaudeau. "*The PolarLITIS dataset: road scenes under fog*", Transactions on Intelligent Transportation Systems (T-ITS), 2021.

  2 Rachel Blin, Samia Ainouz, Stéphane Canu, Fabrice Mériaudeau. "*Road scene analysis under fog: towards an optimal conventional-non-conventional fusion scheme*", soumis à Transactions on Intelligent Vehicles (T-IV), 2021.

  3 Cyprien Ruffino, Rachel Blin, Samia Ainouz, Gilles Gasso, Romain Hérault, Fabrice Mériaudeau, Stéphane Canu. "*Generating physically admissible polarimetric images as data augmentation for road-scene analysis*", Actuellement en préparation.

- Articles de conférences

1 Rachel Blin, Samia Ainouz, Stéphane Canu, Fabrice Mériaudeau. "*Adapted learning for Polarization-based car detection*", Quality Control by Artificial Vision (QCAV), 2019.

2 Rachel Blin, Samia Ainouz, Stéphane Canu, Fabrice Mériaudeau. "*Road scenes analysis in adverse weather conditions by polarization-encoded images and adapted deep learning*", IEEE International Conference on Intelligent Transportation Systems (ITSC), 2019.

3 Rachel Blin, Samia Ainouz, Stéphane Canu, Fabrice Mériaudeau. "*A new multimodal rgb and polarimetric image dataset for road scenes analysis*", Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshops), 2020.

4 Rachel Blin, Samia Ainouz, Stéphane Canu, Fabrice Mériaudeau. "*Multimodal Polarimetric And Color Fusion For Road Scene Analysis In Adverse Weather Conditions*", IEEE International Conference on Image Processing (ICIP), 2021.

5 Rachel Blin, Samia Ainouz, Stéphane Canu, Fabrice Mériaudeau. "*Road scene analysis: A study of polarimetric and color-based features under various adverse weather conditions*", soumis à IEEE International Conference on Computer Vision Workshops (ICCV Workshops), 2021.

Organisation

Cette thèse est divisée en six chapitres comme suit :

- Le chapitre 1 introduit le concept de multimodalité en présentant l'imagerie polarimétrique, l'imagerie couleur, l'imagerie infrarouge et les nuages de points LiDAR.

- Le chapitre 2 présente les connaissances de base sur l'apprentissage profond, notamment les détecteurs d'objets et les CycleGAN.

- Le chapitre 3 passe en revue l'état de l'art des applications des images polarimétriques et le rôle de la multimodalité dans la conduite autonome. Il décrit également les différents détecteurs d'objets et les architectures de fusion multimodales.

- Le chapitre 4 détaille les différents jeux de données constitués pour réaliser les expériences de cette thèse.

- Le chapitre 5 démontre le rôle de l'imagerie polarimétrique pour améliorer l'analyse des scènes routières dans le brouillard.

- Le chapitre 6 présente les pipelines de fusion utilisant l'imagerie couleur et polarimétrique améliorant la détection d'obstacles routiers dans un large éventail de conditions météorologiques dégradées.

- Enfin, nous tirons des conclusions sur notre travail et présentons les perspectives futures.

Chapter 1

Background on multimodality

Contents

1.1 Introduction

The world is composed of diverse phenomena that are perceived differently through the lens of the different living beings. This set of perceptions constitutes the characteristics of each phenomenon, which can be captured by different acquisition frameworks. The information resulting of each acquisition framework, or in other terms, each sensor, is called ”modality” [38].

Color-based imaging, sound, or both information combined resulting in a video are examples of modalities inspired by the Human perception. Since it is rare that a single modality provides the complete knowledge of a natural phenomenon, these modalities are usually combined to provide rich characteristics to describe it. However, the human perception is not sufficient to face complex situations. Some sensors are designed to acquire information circumventing to Human perception, which enable to describe phenomenon difficult or impossible to characterize otherwise. The sensors

of this category are known as non-conventional sensors, capturing non-conventional modalities.

Non-conventional modalities are used in a wide range of fields. They can be processed on their own, or in complement to conventional ones. In the medical branch, Computed Tomography combined with Magnetic Resonance Imaging and Positron Emission Tomography to name a few, enable a precise description of the internal functioning of the body [39]. Non-conventional modalities, such as infrared imaging [12, 40, 41] or Lidar [42, 43, 44], combined with color-based images are used in autonomous vehicles or ADAS to reliably describe road scenes. These modalities are essential given that they are more robust to day/night variation or adverse weather conditions than conventional imaging; yet without totally enabling autonomous driving in every situation [9].

In this thesis, the impact of polarimetric imaging in improving road scene analysis under adverse weather conditions is addressed. As a matter of fact, this non-conventional modality describes an object by its physical information, even under poor illumination or strong reflections [18]. This section clarifies the formalism of polarimetric images, as well as four color spaces including RGB, Hue, Saturation, Value (HSV), Lightness, Green-magenta chromatic axis, Blue-Yellow chromatic axis (CIE Lab) and Luminance, Chrominance (red-yellow), Chrominance (blue-yellow) (YCrCb). Infrared imaging and LiDAR technologies are also described, as they play a significant role in the autonomous driving field.

1.2 Linear Polarization formalism

In its propagation plan, the electromagnetic field of a plane progressive transverse wave [45], with a pulsation ω and a phase ψ, in the orthonormal basis $B = \{\vec{u_x},\vec{u_y}\}$ has the following equation:

$$\vec{E}(t) = E_x(t)\cos(-k\vec{z} + \omega t + \psi_x(t))\vec{u_x} + E_y(t)\sin(-k\vec{z} + \omega t + \psi_y(t))\vec{u_y} \ , \qquad (1.1)$$

where k is the wave number, t is the time, $\vec{z}$ the direction of propagation and E_x, E_y are respectively the amplitudes of $\vec{E}(t)$ according to $\vec{u_x}$ and $\vec{u_y}$. E_y and E_x are respectively related to the electric and the magnetic fields of the electromagnetic wave. ψ_x and ψ_y are respectively the components according to $\vec{u_x}$ and $\vec{u_y}$ of the phase of the electromagnetic wave. Figure 1.1 illustrates the composition of an electromagnetic wave.

Polarization is the property of light waves describing the direction in which the wave is travelling. It is better understood by introducing the three states of polarization of the light [18]:

- the wave is totally polarized when the direction of its electrical field is well determined, in this case, it is elliptic, linear or circular,
- the wave is unpolarized when the light waves oscillate in totally random directions,

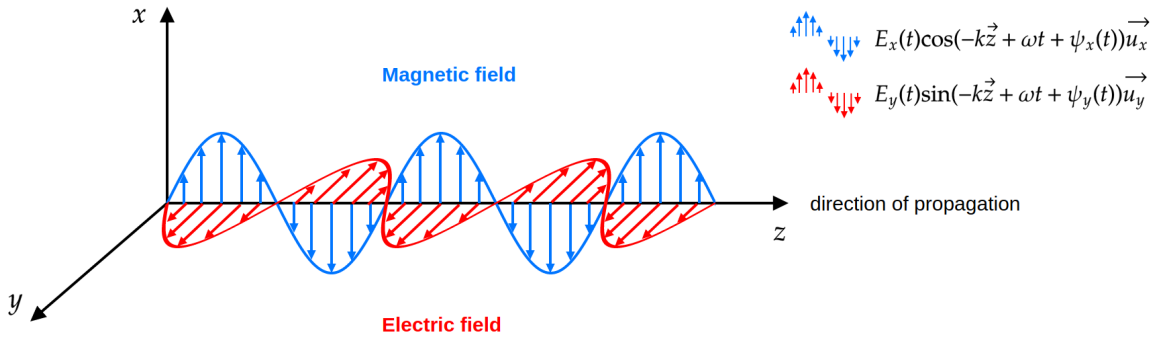

Figure 1.1: Electric and magnetic fields of a light wave.

- it is partially polarized when there is a combination of a polarized part and an unpolarized part.

Polarimetric imaging consists in giving the polarization state of the reflected light wave for each pixel of a scene. It is historically used to dissociate metallic object from dielectric surface [46]. The mechanism of the polarization is that when an unpolarized light wave is being reflected by an object, it becomes partially linearly polarized. The reflected light wave can be described by a measurable vector, called the linear Stokes vector, $S = \begin{bmatrix} S_0 & S_1 & S_2 \end{bmatrix}^\top$. It is defined as the co-variance parameters of the electromagnetic wave components of equation (1.1):

$$\begin{bmatrix} S_0 \\ S_1 \\ S_2 \end{bmatrix} = \begin{bmatrix} \langle {E_x}^2 \rangle + \langle {E_y}^2 \rangle \\ \langle {E_x}^2 \rangle - \langle {E_y}^2 \rangle \\ 2\langle E_x E_y \cos(\psi_y - \psi_x) \rangle \end{bmatrix} ,$$

where $\langle . \rangle$ refers to the temporal mean of the signal.

By its construction, the Stokes parameters satisfy the physical admissibility constraints defined by the following equations:

$$S_0 > 0 \quad \text{and} \quad {S_0}^2 \geqslant {S_1}^2 + {S_2}^2 \ . \tag{1.2}$$

The first constraint means that any object reflects a light. The second constraint means that the total energy is always greater than the sum of the partial ones. The reflected wave is thus totally polarized if the equality holds meaning that ${S_0}^2 = {S_1}^2 + {S_2}^2$. It is partially polarized if we have strict inequality and unpolarized if $S_0 > 0$, $S_1 = 0$, $S_2 = 0$.

In order to obtain polarimetric images, a polarizer oriented at a specific angle α_i is placed between the scene and the sensor. Most of polarimetric cameras use four linear polarizers, oriented at four different angles ($\alpha_{i,i=1:4} = \{0°, 45°, 90°, 135°\}$), enabling to get simultaneously four different intensities $I_{\alpha_i,i=1:4}$ of the same scene. Figure 1.3 illustrates the output of a polarimetric camera for each orientation. The light wave is filtered in order to recover its polarized part in the desired orientation. This filtering process is illustrated for each of the four linear polarizers in Figure 1.2. To get the three Stokes parameters, at least three different orientations of the polarizer are needed.

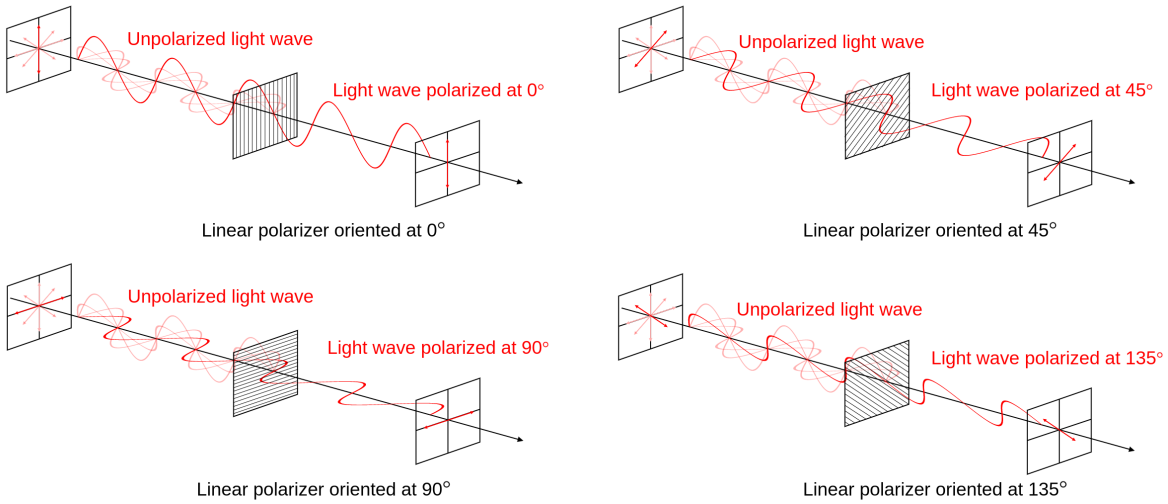

Figure 1.2: Illustration of the filtering action of the four linear polarizers on an unpolarized light wave. They transmit a polarized light wave in the desired orientation. Here, the four linear polarizers are respectively oriented at 0°, 45°, 90° and 135°.

The relationship between each intensity I_{α_i} and the Stokes parameters is illustrated in Figure 1.4 and given by:

$$I_{\alpha_i} = \frac{1}{2}\begin{bmatrix}1 & \cos(2\alpha_i) & \sin(2\alpha_i)\end{bmatrix}\begin{bmatrix}S_0\\S_1\\S_2\end{bmatrix} \quad , \qquad \forall i = 1,\dots,4 \ . \tag{1.3}$$

In a more compact representation, equation (1.3) can be written in the following way:

$$I = AS \ , \tag{1.4}$$

where $I = \begin{bmatrix}I_0 & I_{45} & I_{90} & I_{135}\end{bmatrix}^\top$ refers to the four intensities according to each angle of the polarizer. $S = \begin{bmatrix}S_0 & S_1 & S_2\end{bmatrix}^\top$ is the Stokes vector and $A \in \mathbb{R}^{4\times 3}$ is commonly called the calibration matrix of the linear polarizer, defined as:

$$A = \frac{1}{2}\begin{bmatrix}1 & \cos(2\alpha_1) & \sin(2\alpha_1)\\1 & \cos(2\alpha_2) & \sin(2\alpha_2)\\1 & \cos(2\alpha_3) & \sin(2\alpha_3)\\1 & \cos(2\alpha_4) & \sin(2\alpha_4)\end{bmatrix} = \frac{1}{2}\begin{bmatrix}1 & 1 & 0\\1 & 0 & 1\\1 & -1 & 0\\1 & 0 & -1\end{bmatrix} \ . \tag{1.5}$$

Knowing the intensities $I_{\alpha_i, i=1:4}$ reaching the camera and the calibration matrix A, the only unknowns in equation (1.4) are the Stokes parameters. As the matrix A is not square, the most used solution in the literature to get the Stokes parameters for each pixel is the least mean square solution. The Stokes vector is then calculated by the following formula:

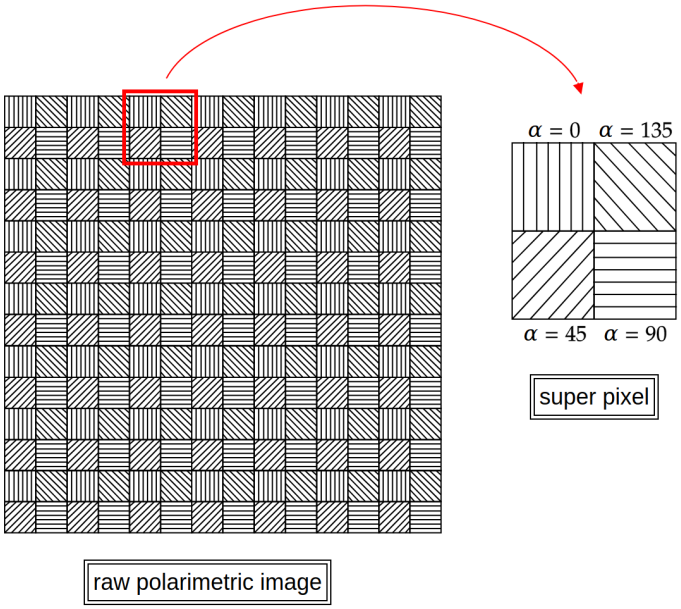

Figure 1.3: Output of a polarimetric camera with four linear polarizers respectively oriented at 0°, 45°, 90° and 135°. The raw polarimetric image is constituted of super pixels. Each super pixel contains the values of the four intensities related to the four linear polarizers for each pixel of the scene. The intensities are reconstructed by selecting their corresponding value in each super pixel.

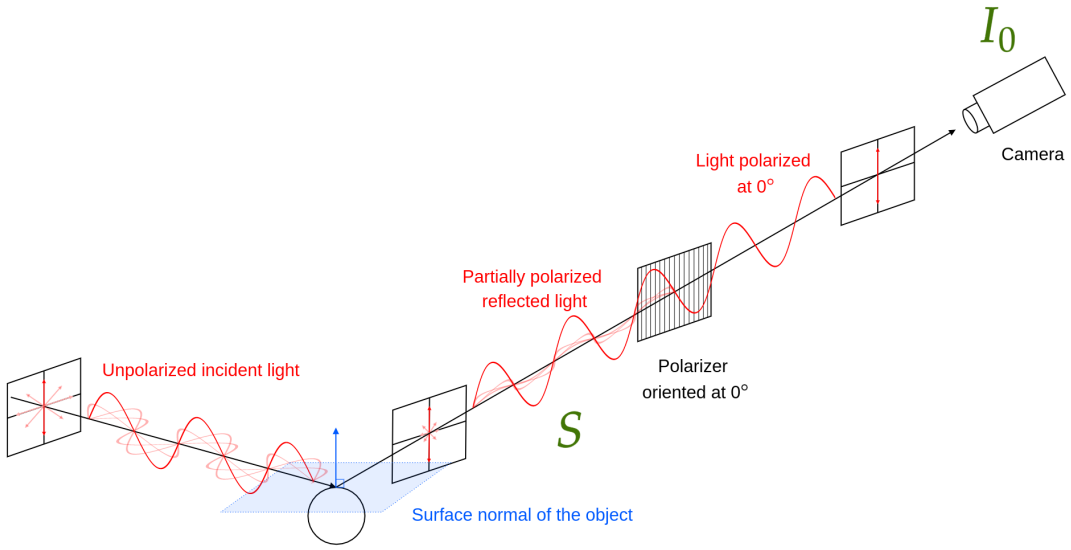

Figure 1.4: Illustration of the polarization process of the light wave. The incident light is being reflected by the object it impinges on and becomes partially polarized. This reflected light can be described by the Stokes vector S. It is then filtered by a polarizer oriented at a desired angle α_i. The filtered wave is captured by the camera to get the intensity I_{α_i} associated to the object. Here the process is illustrated for $\alpha_i = 0$ which is $I_{\alpha_i} = I_0$.

$$S = \tilde{A}I \ , \tag{1.6}$$

where $\tilde{A} = (A^\top A)^{-1}A^\top$ the pseudo-inverse matrix of A. The proposed mean square solution is submitted to some additional constraints on the acquired intensities. Indeed, if we combine equations (1.4) and (1.6), we get the following equation:

$$I = A\tilde{A}I \ . \tag{1.7}$$

This equality holds if and only if:

$$I_0 + I_{90} = I_{45} + I_{135} \ . \tag{1.8}$$

The proof of the condition in equation (1.8) can be found in appendix C.

Note that from equations (1.4), (1.6) and (1.8), the Stokes vector can be given by:

$$S = \begin{bmatrix} I_0 + I_{90} \\ I_0 - I_{90} \\ I_{45} - I_{135} \end{bmatrix} \ . \tag{1.9}$$

Other important physical parameters can be obtained from the Stokes parameters, the Angle Of Polarization (ϕ) and the Degree Of Polarization (ρ) [47]. ϕ and ρ can be determined from the obtained Stokes vector with the following formulas:

$$\phi = \frac{1}{2}\arctan2\left(\frac{S_2}{S_1}\right) \ , \tag{1.10}$$

$$\rho = \frac{\sqrt{{S_1}^2 + {S_2}^2}}{S_0} \ . \tag{1.11}$$

$\rho \in [0, 1]$ is one of the most important physical properties. It refers to the quantity of polarized light in a wave. It is equal to 1 for a totally polarized light, between 0 and 1 for the partially polarized light and to 0 for an unpolarized light. $\phi \in \left]-\frac{\pi}{2}; \frac{\pi}{2}\right]$ is the orientation of the polarized part of the wave with regards to the incident plan. ρ and ϕ can be represented in a unitary circle following equations (1.2), (1.10) and (1.11). This representation can be found in Figure 1.5. Figure 1.6 illustrates an example of a road scene and its representation within this non-conventional space.

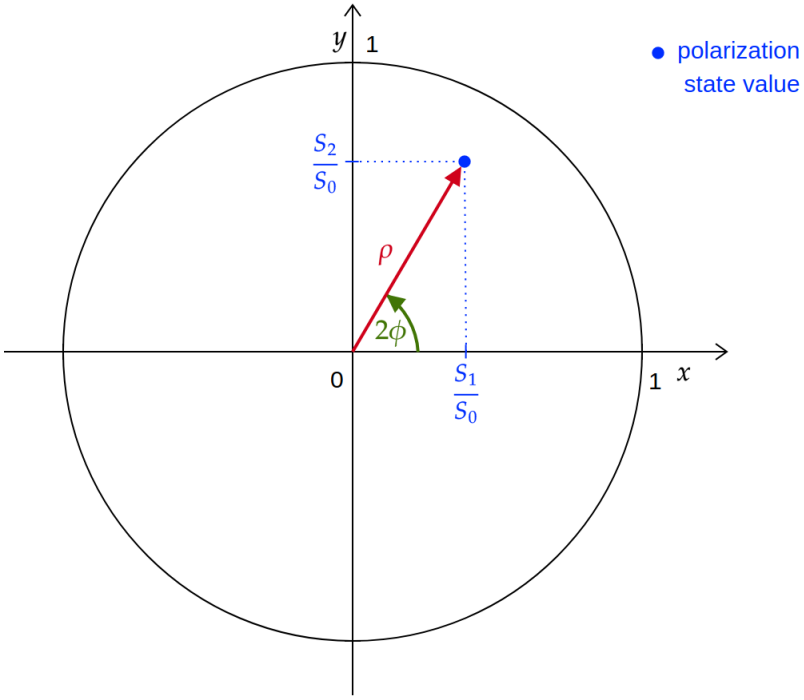

Figure 1.5: Representation of ρ and ϕ in the unitary circle. The abscissa is the value of the second Stokes parameter S_1 normalized by the first Stokes parameter S_0 and the ordinate is the value of the third Stokes parameter S_2 normalized by S_0. ρ is the norm of the vector of the polarization state at coordinates $(\frac{S_1}{S_0}, \frac{S_2}{S_0})$ and ϕ is half of the angle formed by the abscissa and the vector of the polarization state.

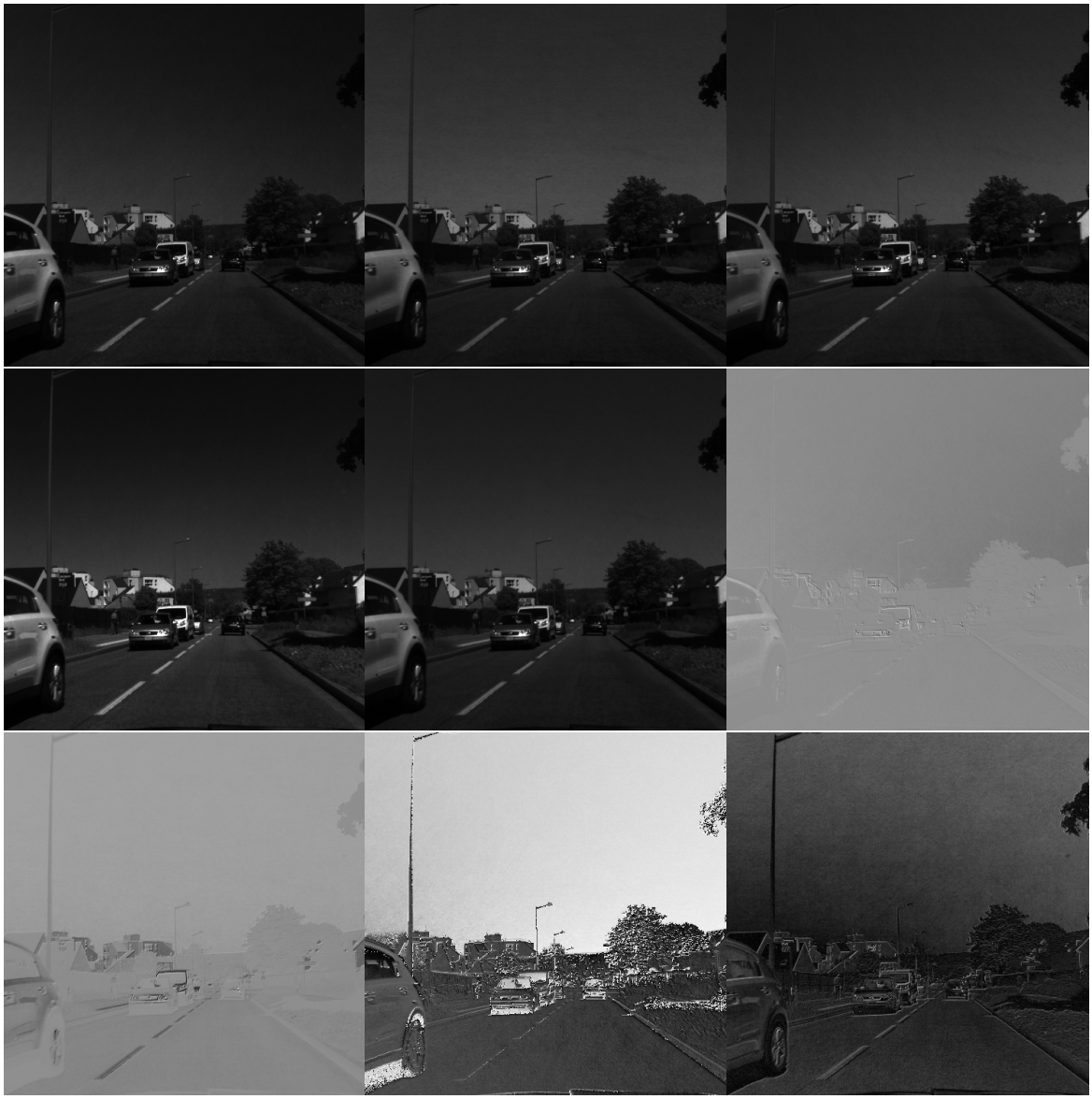

Figure 1.6: Example of a polarimetric image. From left to right and from top to bottom: I_0, I_{45}, I_{90}, I_{135}, S_0, S_1, S_2, ϕ and ρ.

1.3 Color models

Color models are mathematical representations of color. They are usually represented by three values to match the human trivariant color vision. Each color is characterized by a triplet of values that can be represented by a point in a 3D space. The set of all the colors described by a color model is a color space.

There are plenty of different color spaces that aim to highlight different properties of an image. While some of them describe the image according to three chromaticities, others enable to visualize its hue or its saturation. In the computer vision field, the representation of an image in a given color space is made by concatenating three grayscale images. It is possible to convert an image from one color space to another since they contain the same amount of information while represented differently.

In this section, the four color spaces that are used in this thesis are detailed. First, the RGB color space is described, followed by the HSV color space, the CIE Lab color space and finally details are given about the YCrCb color space.

1.3.1 RGB color space

The RGB color space is described by three chromaticities which are the red, the green and the blue. Figure 1.7 represents the RGB color space. This color space is created according to the Young-Helmholtz theory [48], stating that all the visible colors can be described by the red, the green and the blue light waves. These three light waves refer to the three cones of the human eye, which restitute a color by combining the red, the green and the blue information they receive.

The first RGB image is constituted in 1861 by James Clerk Maxwell [49] by projecting three color-filtered images (using a red, a green and a blue filters) on a white wall. This first experiment is the base of the additive synthesis of images. In the case of the RGB color space, the additive synthesis is made by using the three additive primary colors: red, green and blue. An illustration of an RGB image and of the content of each channel can be found in Figure 4.15.

In computer vision, the principle of combining red, green and blue chromaticities is still applied to constitute RGB images. These three chromaticities constitute the channels of the image. Each pixel of an image has a color value which is a combination of the values respectively of the red, green and blue channels of this pixel. These three values are integers usually within the [0, 255] range for 8 bits images and sometimes within the [0, 65535] range for 16 bits images. The shade of a color is deduced by the greatest value of the three channels. If the greatest value is in the red channel, the color has a red shade. The same principle applies to the blue and green channels. In the meantime, values close to 0 give a darker shade to the color whereas the ones close to 255 (or 65535 for 16 bits images) give a lighter shade to the color. White is reached when the three channels are equal to 255 for 8 bits images (or 65535 for 16 bits images) whereas black is reached when the three channels are equal to 0. Figure 1.7 illustrates an example of how a color is defined, according to its red, green and blue channels values in 8 bits images and Figure 4.15 shows an RGB image and its channels decomposition.

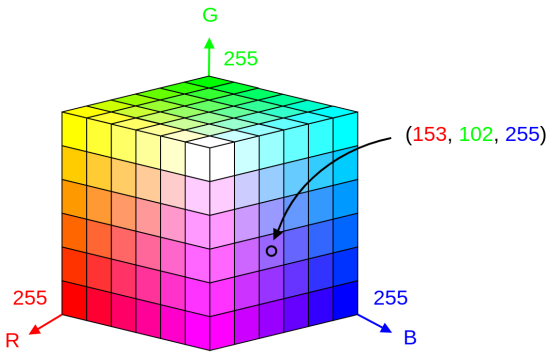

Figure 1.7: RGB color space representation in 3D space. R, B and G respectively represent the red, the blue and the green axis. The color in the example has a red value of 153, a green value of 102 and a blue value of 255.

Figure 1.8: Example of an RGB image and its channels decomposition. From left to right, the red channel, the green channel, the blue channel and their concatenation to constitute an RGB image.

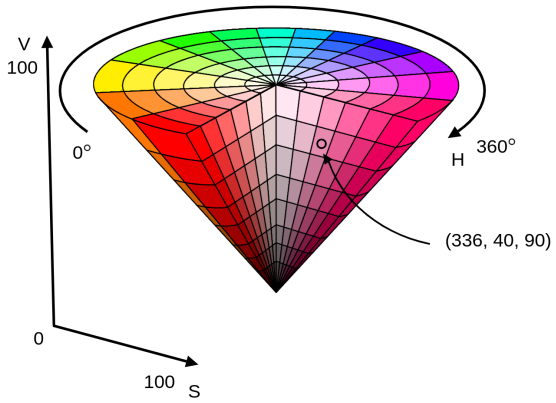

Figure 1.9: HSV color space representation in 3D space. H represents the hue axis, S represents the saturation axis and V represents the value axis. The color in the example has a hue of 336°, a saturation of 40 and a value of 90.

The RGB color space is considered a reference to all the other color spaces. The equations to convert an image from each color space to the RGB color space (stored in 8 bits) are given in the following sections.

1.3.2 HSV color space

The HSV color space is created in 1970 by computer graphics researchers [50]. The intuition behind this color space is to describe colors so they can align with the human perception. Indeed, describing a color by its proportion of red green and blue is not intuitive. This color model is thus described by the hue of a color, its saturation and its value. The goal of this color space is to model the colors under the light. The maximum values of colors can be seen as the white light projection on a colored object, enabling to have a bright and intense color. The other values of the colors can be seen as a dimmer light projection on the same colors which results in darker and less bright colors. The HSV color space is illustrated Figure 1.9.

The HSV color space can be described as a hexcone model of which the base is the optimal color limits with white at the center of the base of the cone and black at the vertex of the cone. The three parameters describing the HSV color space are the followings:

- The hue (H), describing the color shade within the [0°, 360°] range. It describes the transition between the primary colors in an angular way. The transition starts with the primary red at 0°, then passes through the primary green at 120° and the primary blue at 240° to come back to the primary red at 360°.
- The saturation (S), describing how pure the color is with a value within the [0, 100] range. A saturation of 100 leads to the pure color whereas a saturation of 0 refers to white.
- The value (V), describing how dark the color is with a value within the [0, 100] range. A value of 100 results in a bright color whereas a value of 0 refers to black.

Figure 1.10: Example of an HSV image and its channels decomposition. From left to right, the hue, the saturation, the value and their concatenation to constitute an HSV image.

It is important to note that, in the HSV color space, two colors are complementary if, mixed in the same proportions, their saturation is equal to 0.

In the computer vision field, the three components of images in the HSV color space are rescaled to constitute 8 bits images. H becomes an integer within the [0, 179] range while S and V become integers within the [0, 255] range. Figure 1.10 shows an HSV image and its channels decomposition.

To convert an HSV image to the RGB color space [50], the following parameters are first computed:

$$\begin{cases} V & = \frac{V}{100} \times 255 \\ S & = \frac{S}{100} \\ \tilde{H} & = \left\lfloor \frac{H}{60} \right\rfloor \mod 6 \\ F & = \frac{H}{60} - \tilde{H} \\ L & = V \times (1 - S) \\ M & = V \times (1 - F \times S) \\ N & = V \times (1 - (1 - F) \times S) \end{cases}$$

they are then placed into different configurations to constitute the RGB image, according to the value of H_i, which are:

$$(R, G, B) = \begin{cases} (V, N, L) & \text{if } \tilde{H} = 0 \\ (M, V, L) & \text{if } \tilde{H} = 1 \\ (L, V, N) & \text{if } \tilde{H} = 2 \\ (L, M, V) & \text{if } \tilde{H} = 3 \\ (N, L, V) & \text{if } \tilde{H} = 4 \\ (V, L, M) & \text{if } \tilde{H} = 5 \end{cases}$$

1.3.3 CIE Lab color space

The CIE Lab color space is defined from the CIE XYZ color space [51], an enhanced version of the RGB color space enabling a better spacial representation of colors. It is created in 1976 to better characterize the surface's colors [52]. Its goal is to describe the colors' repartition similarly to the color offsets perception of the human eye. This

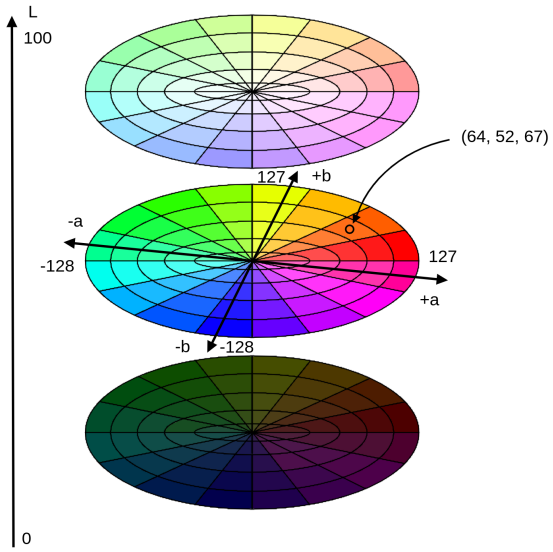

Figure 1.11: CIE Lab color space representation in 3D space. L represents the luminance axis, A is the offset to the cyan $\rightarrow$ magenta axis and B is the offset to the blue $\rightarrow$ yellow axis. The color in the example has a luminance of 64, a 52 offset to the cyan $\rightarrow$ magenta axis and a 67 offset to the blue $\rightarrow$ yellow axis.

color space is described by three parameters. The first parameter is the luminance of the surface (L) and the two others (a and b) are the offsets of the color to the gray surface with the same luminance. The offset to the gray surface is defined by the offset to the cyan $\rightarrow$ magenta axis (a) and the one to the blue $\rightarrow$ yellow axis (b). These three parameters constitute the three channels of an image in the CIE Lab color space. This color space is illustrated in Figure 1.11.

The existing colors in the CIE Lab color space actually constitute a complex cornet-shaped geometrical volume with the upper vertex corresponding to white and the lower vertex to black. For a better understanding, this complex geometry is here approximated with three circular slices corresponding to three different luminances (see Figure 1.11). The center of each of the three circles constitutes a gray level (a=b=0). If two points are at an equal distance from the gray surface, their color difference is equal, which means they have the same contrast.

For 8 bits images, each point of the CIE Lab color space is within the following color ranges:

- The luminance L is an integer set within the [0, 100] range,
- The offset towards the cyan (-a) $\rightarrow$ magenta (+a) axis is an integer within the $[-128, 127]$ range,
- The offset towards the blue (-b) $\rightarrow$ yellow (+b) axis is an integer within the $[-128, 127]$ range.

In this configuration, a negative value of "b" gives a blue shade to the color whereas a positive value of "b" gives a yellow shade to the color. In the same way, a negative value

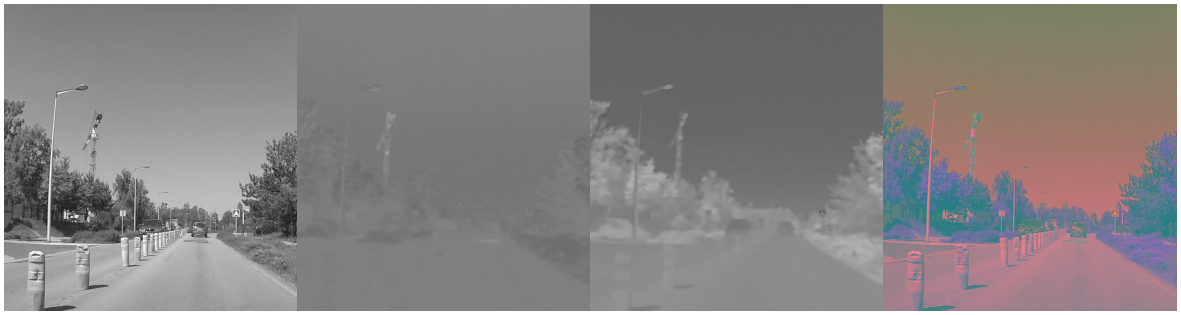

Figure 1.12: Example of a CIE Lab image and its channels decomposition. From left to right, the luminance, the offset towards the cyan (-a) → magenta (+a) axis, the offset towards the blue (-b) → yellow (+b) axis and their concatenation to constitute a CIE Lab image.

of "a" gives a cyan shade to the color whereas a positive value of "a" gives a magenta shade to the color. As for the luminance, a value close to 0 gives a darker shade to the color whereas a value close to 100 gives a lighter shade to the color. Figure 1.12 shows a CIE Lab image and its channels decomposition.

To convert a value from the CIE Lab color space to the RGB color space [52, 51], it must be converted to the CIE XYZ color space following:

$$\begin{cases} X &= X_n f\left(\frac{L+16}{116} + \frac{a}{500}\right) \\ Y &= Y_n f\left(\frac{L+16}{116}\right) \\ Z &= Z_n f\left(\frac{L+16}{116} - \frac{b}{200}\right) \end{cases}$$

with:

$$\begin{cases} X_n &= 95.0489 \\ Y_n &= 100 \\ Z_n &= 108.8840 \end{cases}$$

and:

$$f(x) = \begin{cases} x^3 & \text{if } x > \frac{6}{29} \\ \frac{1}{3}(\frac{29}{6})^2\left(x - \frac{4}{29}\right) & \text{else} \end{cases}$$

Once the CIE Lab coordinates are converted to CIE XYZ coordinates, they can be converted to RGB coordinates following:

$$\begin{cases} R_l &= 3.24096994X - 1.53738318Y - 0.49861076Z \\ G_l &= -0.96924364X + 1.8759675Y + 0.04155506Z \\ B_l &= 0.05563008X + -0.20397696Y + 1.05697151Z \end{cases}$$

with R_l, G_l and B_l are respectively the red, green and blue linear values on which we need to apply a gamma correction to obtain the final values, which is:

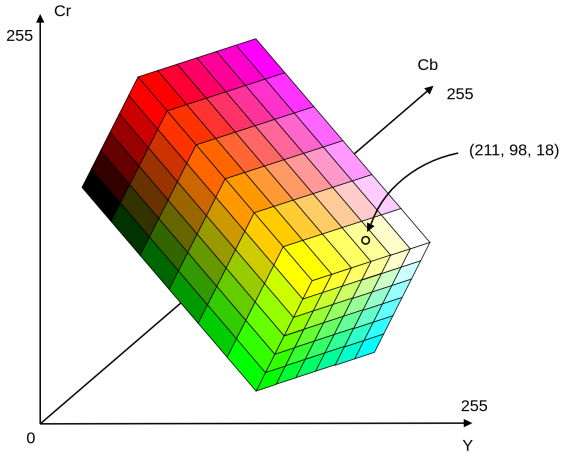

Figure 1.13: YCrCb color space representation in 3D space. Y represents the luma axis, Cr represents the red chrominance axis and Cb represents the blue chrominance axis. The color in the example has a luma of 211, a 98 red chrominance value and a 68 blue chrominance value.

$$\begin{cases} R &= \gamma(R_l) \\ G &= \gamma(G_l) \\ B &= \gamma(B_l) \end{cases}$$

with:

$$\gamma(x) = \begin{cases} 12.92x & \text{if } x \leq 0.0031308 \\ 1.055x^{\frac{1}{2.4}} - 0.055 & \text{else} \end{cases}$$

1.3.4 YCrCb color space

The YCrCb color space is created to address the compatibility between color television and black & white television. It follows the intuition that both grayscale and color images are the sum of all the colors that compose them. This color space is described by three components which are the luma, the red chrominance and the blue chrominance. The luma is the sum of the red, the green and the blue components of the image. The red chrominance and the blue chrominance are respectively the difference between the luma and the red component and the luma and the blue component. In this way, the black and white sensors only process the luma component of this data format. The color sensors compute the green component from the luma, as well as the red and the blue components to display color images. The YCrCb color space is represented in Figure 1.13.

In the computer science field, the three components of YCrCb images are integers within the [0, 255] range to constitute 8 bits images. Figure 1.14 shows an YCrCb image and its channels decomposition.

An image can be converted from the YCrCb color space to the RGB color space [53] using the following equations:

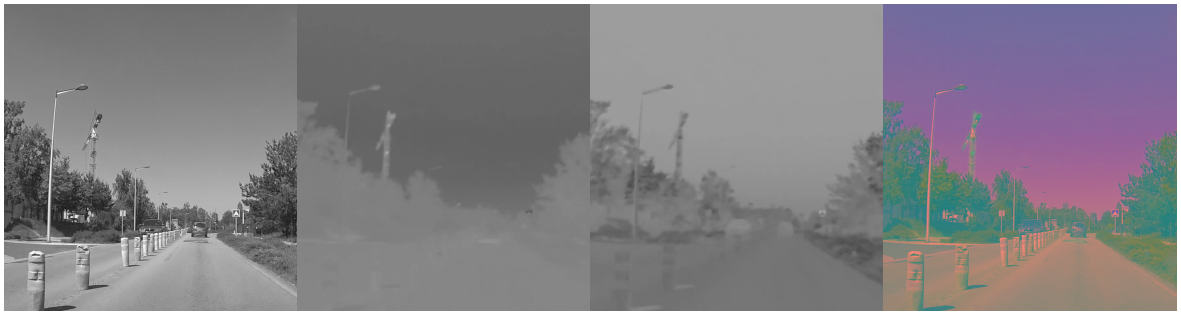

Figure 1.14: Example of an YCrCb image and its channels decomposition. From left to right, the luma, the red chrominance, the blue chrominance and their concatenation to constitute an YCrCb image.

$$\begin{cases} R &= Y + 1.402(Cr - 128) \\ G &= Y - 0.34414(Cb - 128) - 0.71414(Cr - 128) \\ B &= Y + 1.772(Cb - 128) \end{cases}$$

1.4 Other non-conventional imaging systems

In this section, the passive infrared and the LiDAR modalities are explained. These two non-conventional modalities play an important role in ADAS and autonomous vehicles. A brief background on their functioning is given to understand the state of the art of autonomous driving.

1.4.1 Passive infrared

Passive infrared sensors measure the quantity of infrared light radiating from objects [54]. All the objects that have a temperature greater than the absolute zero (-273.15°C=-459.67°F) emit heat in the form of electromagnetic radiation. These electromagnetic waves are invisible with the human eye since their wavelength are in the infrared domain. The wavelength of the emitted electromagnetic radiation depends on the temperature of an object. The hotter the object, the smaller the wavelength is. Figure 1.15 illustrates the different spectrum of an electromagnetic wave according to their wavelength λ.

In the computer vision field, passive infrared images are relatively used for person detection [55, 56, 57] since the corporal heat is usually greater than the ambient heat. This corporal heat remains the same whatever the luminosity of the scene. Because passive infrared imaging provides information invisible with the human eye, it is a complementary information to the color domain. Passive infrared images are grayscale images within the [0, 255] range for 8 bits images and within the [0, 65535] range for 16 bits images.

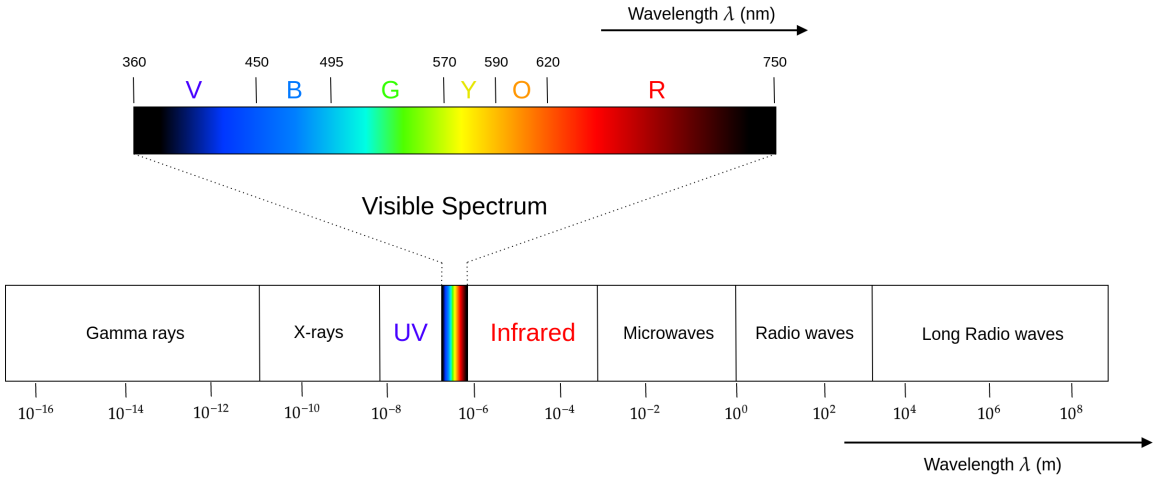

Figure 1.15: The different spectrum according to their wavelength λ. The visible spectrum has light waves between 360 nm and 750 nm whereas the infrared domain gathers all the light waves between 750 nm and 1 mm. Here, UV stands for the ultraviolet domain. In the visible spectrum, V, B, G, Y, O and R respectively stand for violet, blue, green, yellow, orange and red.

1.4.2 LiDAR point clouds

The LiDAR is an active sensor that measures the distance of an object from an emitter using a laser beam with a wavelength either in the ultraviolet, the visible or the infrared spectra [58]. The sensor actually measures the time laps between the emission of light pulse and the reception of the reflected light pulse. Figure 1.16 illustrates the mechanism behind the LiDAR sensor.

The following equation enables to measure the distance of an object from the LiDAR emitter:

$$d = \frac{c}{2}(tr - t_0)$$

with t_0 the emission time of the light pulse, tr the reception time of the most important echo of the reflected light pulse and $c = 299800\ km.s^{-1}$ the speed of light.

There are mainly two types of LiDAR:

- the 3D LiDAR that enables to map the target surface in 3D [59, 60],
- the Two Dimensions (2D) LiDAR that evaluates the distance of an obstacle at a given height [61].

LiDAR sensors are very accurate since they have a scope of about a 100 meters with only a few centimeters precision. Since the emitted and reflected light pulses travel at the speed of light, they can process a scene in real time.

LiDAR data are generally stored as 2D or 3D point clouds for respectively 2D and 3D sensors. The 3D point cloud can be converted to grayscale images in 8 bits (values normalized to fit in the [0, 255] range) or 16 bits images (values normalized to fit in the [0, 65535] range).

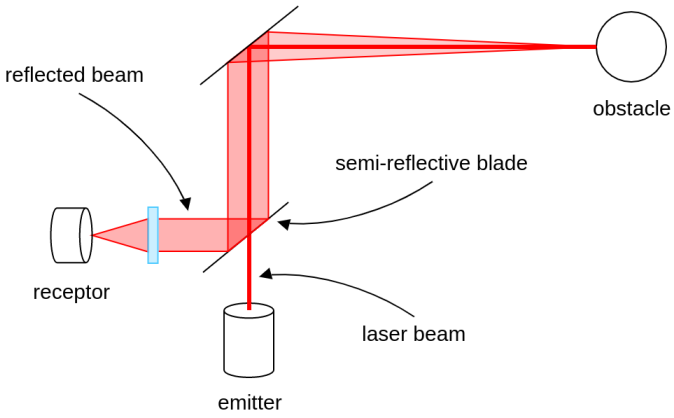

Figure 1.16: Mechanism of the LiDAR sensor. A light pulse is emitted and goes through a semi-reflective blade and is being reflected by a second reflective blade before reaching the obstacle. Once the obstacle is reached, the light pulse is being reflected once by the object and twice by respectively the reflective and the semi-reflective blades before reaching the receptor.

1.5 Summary

This chapter defines the concept of multimodality and gives details about the ones playing an important role in autonomous driving. The physical theory behind a linearly polarized light wave is first detailed. It enables to understand the functioning of the linear polarimetric camera used in this thesis, as we study the role of polarimetric imaging in improving road scene analysis in adverse weather. The different color spaces of conventional imaging are also detailed, including RGB, HSV, CIE Lab and YCrCb. Their specificities as well as the formulas to convert an image from a color space to another are given. Finally, two other non-conventional modalities that play an important role in autonomous driving are explained. The characterization of an object by its temperature is explained through the definition of infrared imaging. The mechanism behind the modelization of a scene in 3D using a LiDAR is also detailed. All these modalities provide complementary information useful to describe a road scene. Their impact in enhancing road scene analysis in complex situations is detailed in the followings chapters.

Chapter 2

Background on Deep Learning

Contents

2.1 Introduction

Deep Learning algorithms are a branch of Machine Learning algorithms which are a branch of Artificial Intelligence themselves. They aim to solve problems associated to large datasets, containing raw and non-processed data.

The year 2012 is a turning point in the history of Deep Learning. A Deep Neural Network (DNN) architecture, AlexNet [62], showed outstanding results at the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [63] that year. It greatly outperformed the Machine Learning algorithms used in this challenge so far. From that moment, DNN gained a lot of popularity and proved their added value in a wide range of fields.

Nowadays, DNN are used to perform numerous and various tasks, including image classification [64], image detection [65], image generation [66] and language processing [67] to name a few. These different tasks enable to enhance medical diagnosis [39], for automatic language translation [68] and are paramount to enable autonomous driving [69].

This chapter presents the necessary background knowledge on Deep Learning, more particularly on DNN for image processing. The concept of an artificial neuron is first detailed. The intuition behind the combination of several artificial neurons, constituting a multilayer perceptron, to solve more complex problems is then presented. The convolution layer, which is the core of Deep architectures performing image processing tasks, is explained. All these basics notions are essential to understand the functioning of Object Detectors. The main architectures performing object detection and their evaluation metrics are explained. Finally, CycleGAN are described. The theory behind all these concepts gives support to the different experiments carried out in this thesis.

2.2 Basics of Deep Learning

This section gives details on the basic concepts behind Deep architectures. It includes the artificial neuron, and the combination of artificial neurons to constitute a Multilayer Perceptron (MLP). The convolution function is then reminded before explaining how it is used to constitute a convolution layer, essential for image processing.

2.2.1 Artificial neurons and Multilayer Perceptron

An artificial neuron can be seen as the mathematical representation of a biological neuron. It is first introduced by Warren *et al.* in 1943 [70].

Each entry of an artificial neuron can be seen as a vector $x = \begin{bmatrix} x_1 & x_2 & ... & x_m \end{bmatrix}^\top$ of m elements. A weight $w = \begin{bmatrix} w_1 & w_2 & ... & w_m \end{bmatrix}^\top$ is associated to this entry. The difference between the element wise multiplication between the entry x and its weight w and a bias w_0 is processed by an activation function f. The output of an artificial neuron is the following:

$$y = f\Big(\sum_{j=1}^{m} w_j x_j - w_0\Big) \ , \tag{2.1}$$

where y is the output of the artificial neuron. Equation (2.1) is illustrated in Figure 2.2. The most common activation functions are illustrated in Figure 2.1.

Artificial neurons used on their own can implement simple operations such as the AND and OR boolean functions. In order to solve more complex problems, the multilayer perceptron is introduced in 1957 by Rosenblatt [71]. This architecture is a combination of several layers of artificial neurons. Each neuron of a layer (l) is connected to each neuron of layer $(l+1)$, implying that the outputs of the neurons of layer (l) are the inputs of the neurons of layer $(l+1)$. The architecture of a multilayer perceptron is illustrated in Figure 2.3.

To automatically correct the errors in the multilayer perceptron's predictions, the backpropagation is introduced in 1986 by Rumelhart *et al.* [72]. The backpropagation enables to corrects the weights of each layer regularly during the training process.

In order to update the weights of a neural network during the training process, it is paramount to quantify the error. To this end, a loss function is used depending on the nature of the input data and the problem to solve. The lower the loss function

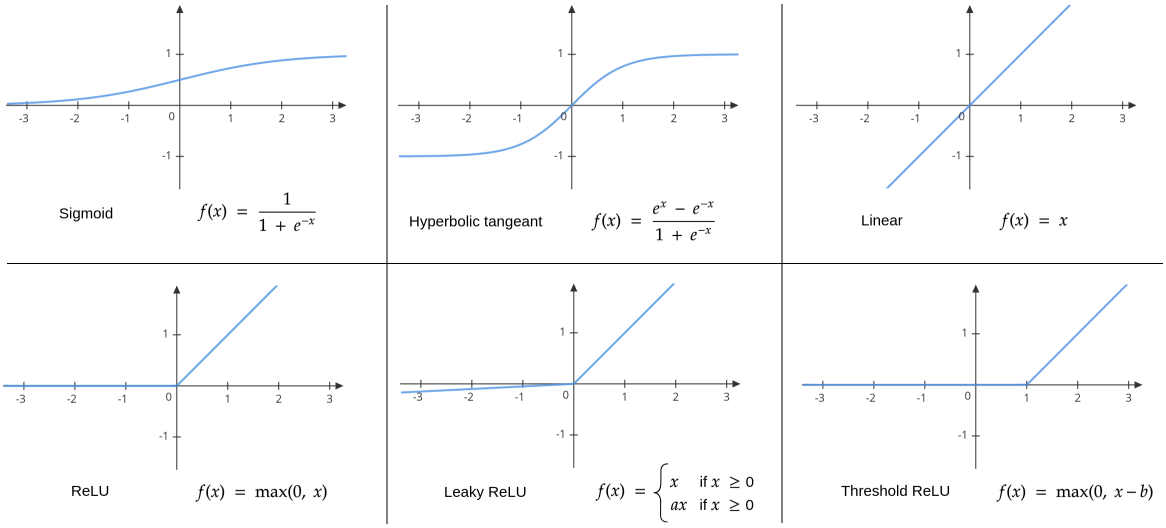

Figure 2.1: Example of activations functions f used by artificial neurons.

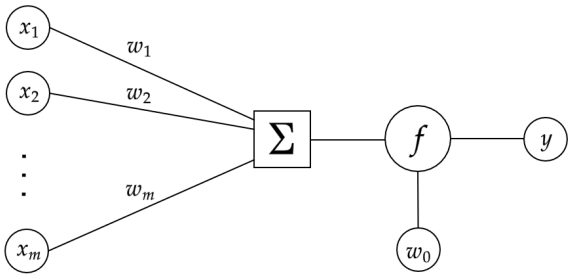

Figure 2.2: Illustration of an artificial neuron. The whole process is summarized up by equation (2.1).

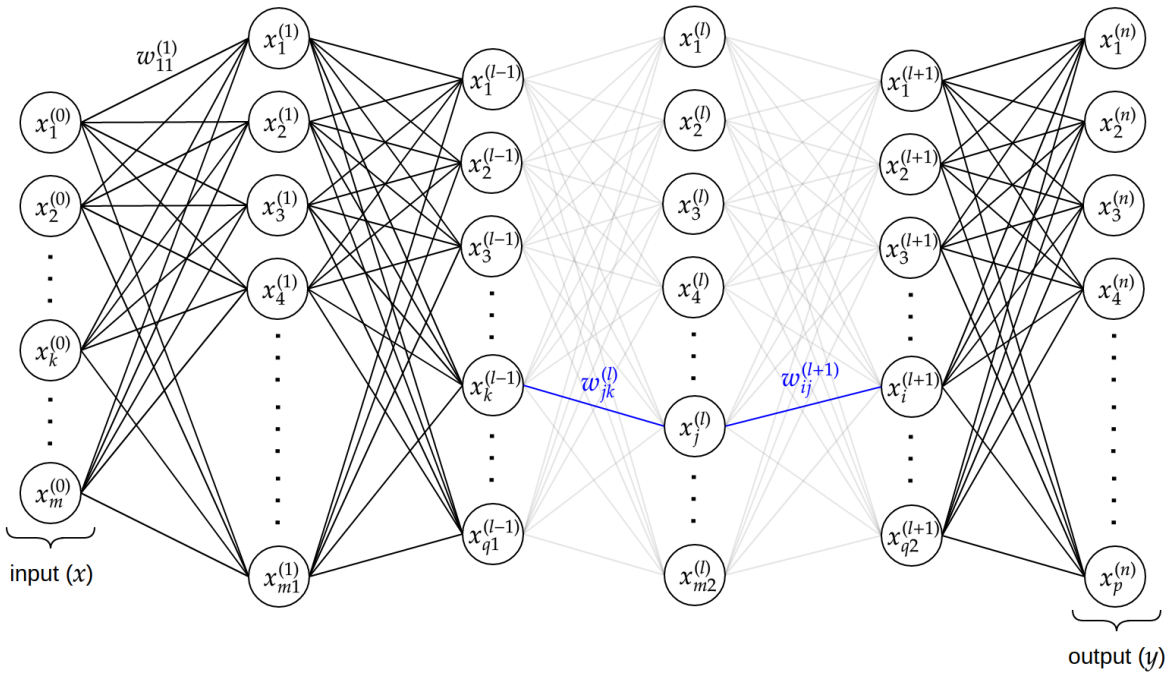

Figure 2.3: Illustration of the architecture of a multilayer perceptron. $x_k^{(l-1)}$ is the k^{th} element of layer $(l-1)$ output and the k^{th} element of layer (l) input. $w_{jk}^{(l)}$ is the weight of the connection between neuron k of layer $(l-1)$ and neuron j of layer (l).

value is, the more accurate the predictions are. The Mean-Squared Error (MSE) is an example of loss function [73], measuring the distance between the prediction and the ground truth, and has the following from:

$$MSE(y,z) = \frac{1}{p}\sum_{i=1}^{p}(y_i - z_i)^2 \tag{2.2}$$

where y is the network prediction, z is the ground truth and p is the number of neurons in the last layer of the perceptron.

The value of the loss function C is the initialization of the backpropagation algorithm. The error $e_j^{(l)}$ of the neuron j of layer (l) is computed the following way:

$$e_j^{(l)} = f'^{(l)}\Big(\sum_{k=1}^{q_1} w_{jk}^{(l)} x_k^{(l-1)}\Big)\sum_{i=1}^{q_2} w_{ij}^{(l+1)} e_i^{(l+1)} \quad \forall l = 1,\dots,n-1 \ , \tag{2.3}$$

$$e_i^{(n)} = C \quad \forall i = 1,\dots,p \ ,$$

where f' is the derivative of the activation function of the artificial neuron j of layer (l), q_1 and q_2 respectively the number of neurons in layers $(l-1)$ and $(l+1)$, $w_{jk}^{(l)}$, is the weight of the connection between neuron k of layer $(l-1)$ and neuron j of layer (l), $x_k^{(l-1)}$ is the output of neuron k of layer $(l-1)$ ($x_k^{(0)}$ is the k^{th} element of the network input) and C is the loss function value.

Using equation (2.3), the weights are updated the following way:

$$w_{ij}^{(l)} = w_{ij}^{(l)} - \kappa e_i^{(l)} x_j^{(l-1)} \quad \forall l = 1, \ldots, n \ ,$$

where $w_{ij}^{(l)}$, is the weight of the connection between neuron j of layer $(l-1)$ and neuron i of layer (l), κ the learning rate, $x_j^{(l-1)}$ is the output of neuron j of layer $(l-1)$ ($x_j^{(0)}$ is the j^{th} element of the network input) and $e_i^{(l)}$ is the error of neuron i of layer (l).

2.2.2 Convolutional Neural Networks

CNN aim to process data while keeping spatial and temporal dependencies. Their architecture is inspired by the human visual cortex. The objective of such architectures is to reduce the data features to the most informative ones while loosing the least information. This mechanism of describing data of the same category by general features, makes CNN scalable to massive datasets.

Several operations are needed to build a CNN. These operation are the convolution, the non-linear rectification and the pooling are described in this section.

Convolution

The convolution operation on two functions f and g is the mean of these two functions and is defined as follows:

$$(f * g)(x) = \int_{-\infty}^{+\infty} f(t)g(x-t)dt \tag{2.4}$$

where $(f * g)(x)$ is the weighted average of the function $f(t)$ at the moment x and $g(x-t)$ is the weighting shifted by amount x.

In CNN, the convolution operation is the sum of the elementwise multiplication between a sliding window and the data matrix. This window is called the convolution kernel or a filter. This operation enables to extract specific features on the data depending on the convolution kernel. The equation of a convolution layer is the following:

$$y(i,j) = \sum_{a=1}^{n} \sum_{b=1}^{n} w(a,b)x(i+a, j+b) \ , \tag{2.5}$$

$$\forall i = 1, ..., H-n+1 \ , \forall j = 1, ..., W-n+1 \ ,$$

where w is a $n \times n$ convolution kernel and the weights of the convolution layer, x is the input of a convolution layer of shape $H \times W$ and y the output of a convolution layer of shape $(H-n+1) \times (W-n+1)$.

The convolution kernel has the same depth as the data it processes. In the image processing field, the convolution kernel is a sliding window aiming to process a $n \times n$ neighborhood around each pixel of an image. When processing a grayscale image, i.e. a one channel image, the shape of the convolution kernel is $n \times n \times 1$. In the same way, when processing a three channels image, the shape of the convolution kernel is $n \times n \times 3$. If the kernel depth is greater than one, the convolution result is the sum of the convolution operations over each channel.

The result of the convolution operation over a matrix is a feature map. Each channel of the feature map extracts useful features to characterize the data. In image processing, each channel of the feature map contains a filtered version of the image (e.g. a blurred image or its edge detection). The depth of a feature map is the number of filters (or convolution kernels) used in the convolution layer to process the image. The weights of a convolution layer are the values of all the convolution kernels it contains.

The size of each feature map depends on different parameters. Its depth is the number of convolution filters used on the data. Its width and height, however, depend on two parameters: the convolution stride and the convolution padding.

The convolution stride is the equivalent of the moment x in equation (2.4). It controls how the filters convolves around the input volume. When processing matrices, the stride is the distance between the current position and the next position on which the convolution operation is applied. The stride s is a strictly positive integer, as it characterizes the distance between two elements of a matrix. Equation (2.5) with a stride s becomes:

$$y(i,j) = \sum_{a=1}^{n}\sum_{b=1}^{n} w(a,b)x((i-1)\times s + a + 1, (j-1)\times s + b + 1) \ , \tag{2.6}$$
$$\forall i = 1, ..., \left\lfloor \frac{H-n+s}{s} \right\rfloor \ , \forall j = 1, ..., \left\lfloor \frac{W-n+s}{s} \right\rfloor \ ,$$

where $\lfloor . \rfloor$ if the floor operation.

Figure 2.4 illustrates the effect of the stride variation on the output shape of the feature map. If the input matrix is of shape $H \times W$ and the convolution kernel of shape $n \times n$, the shape of the resulting feature map is $\lfloor \frac{H-n+s}{s} \rfloor \times \lfloor \frac{W-n+s}{s} \rfloor$, with s the convolution stride. The greater the stride is, the smaller the output width and height will be. In practice, reducing the output shape enables to gain computational efficiency since the output data has smaller spatial dimensions. It also extracts higher level features on the input data.

As mentioned above, another way to control the shape of a feature map is the convolution padding operation. Equation (2.6) of a convolution layer with a stride s and padding p_c becomes:

$$y(i,j) = \sum_{a=1}^{n}\sum_{b=1}^{n} w(a,b)x((i-1)\times s + a + 1, (j-1)\times s + b + 1) \ , \tag{2.7}$$
$$\forall i = 1, ..., \left\lfloor \frac{H-n+p_c+s}{s} \right\rfloor \ , \forall j = 1, ..., \left\lfloor \frac{W-n+p_c+s}{s} \right\rfloor \ ,$$

where p_c is an even positive integer, implying that $\frac{p_c}{2}$ columns are added respectively on the left and on the right borders of the input data and $\frac{p_c}{2}$ lines are added respectively on the top and on the bottom borders of the input data.

The padding operations can be classified into two categories:

- The Valid Padding: there is no padding applied to the matrix. In this case, if the input matrix is of shape $H \times W$, the convolution kernel if of shape $n \times n$ and the stride is s, the shape of the resulting feature map is $\lfloor \frac{H-n+s}{s} \rfloor \times \lfloor \frac{W-n+s}{s} \rfloor$,

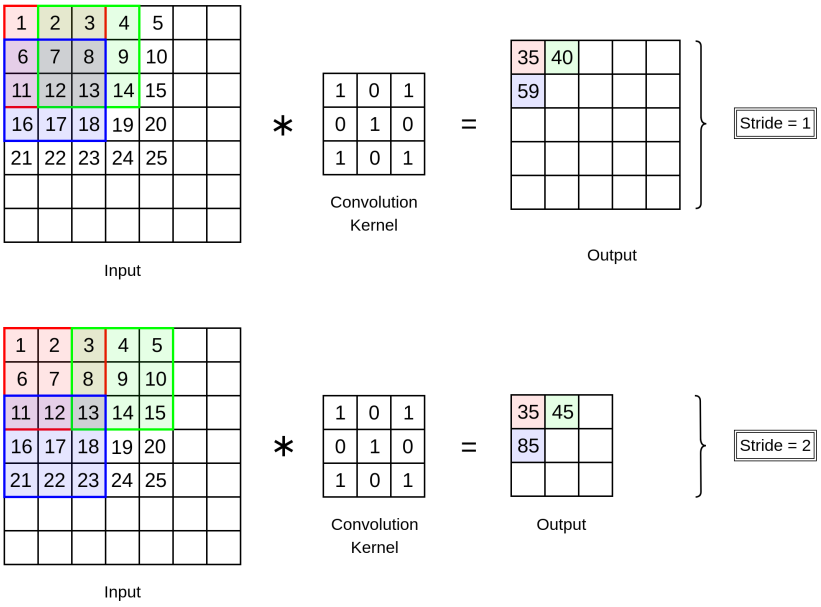

Figure 2.4: Example of a convolution operation when varying the convolution stride. On top, the convolution stride is one and on the bottom, the convolution stride is two. In red, the region of the matrix on which the convolution window is initialized. In green and in blue respectively the next regions of the matrix on which the convolution kernel is applied horizontally and vertically.

- The Same padding: a layer of values (usually zeros known as zero padding) are added around the edges of the input matrix. In this case, the height and width of the resulting feature map are $\lfloor \frac{H-n+p_c+s}{s} \rfloor \times \lfloor \frac{W-n+p_c+s}{s} \rfloor$. It is the same as the input matrix if the convolution stride and the convolution padding are set to $s = 1$ and $p_c = 2$ with a 3×3 kernel.

These two padding operations are illustrated Figure 2.5.

Non-linear rectification

In order to extract relevant features to describe objects, the network can not entirely rely on the convolution operation. Indeed, the convolution operation is linear which results in a linear function if used on its own. However, most of problems can not be solved with a linear function. This is the reason why non-linear information must be introduced into CNN to create non-linear decision boundaries.

Non-linear information is often introduced by applying a non-linear function to the resulting feature map. This process is called the non-linear rectification. The equation of a convolutional layer (see equation (2.7)) with non-linear rectification becomes:

$$y_{\text{rectified}}(i,j) = f\left(\sum_{a=1}^{n}\sum_{b=1}^{n} w(a,b)x((i-1)\times s+a+1,(j-1)\times s+b+1)\right) \quad , \quad (2.8)$$

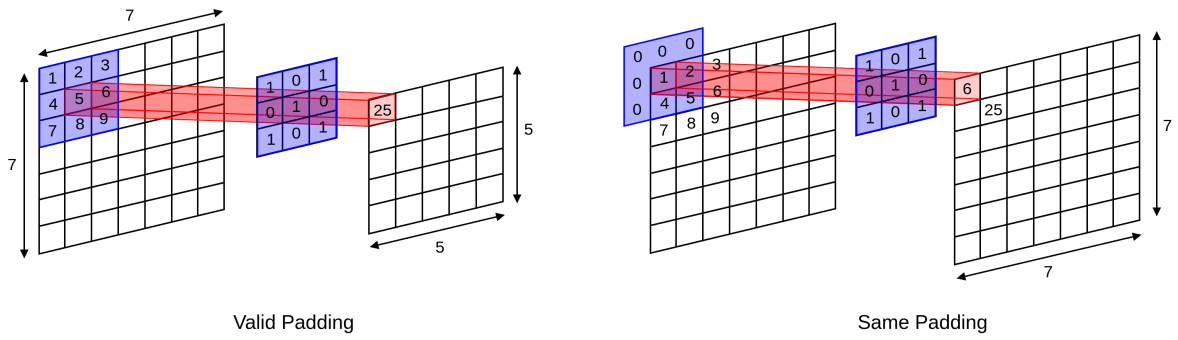

Figure 2.5: Illustration of the padding operations with a stride $s = 1$. On the left, the Valid Padding is illustrated on a 7×7 input matrix, resulting in a 5×5 output matrix after the convolution operation. On the right, the Same Padding is illustrated with a padding $p_c = 2$ on a 7×7 input matrix which remains the same shape after the convolution operation with a 3×3 kernel.

$$\forall i = 1, ..., \left\lfloor \frac{H - n + p_c + s}{s} \right\rfloor \ , \forall j = 1, ..., \left\lfloor \frac{W - n + p_c + s}{s} \right\rfloor \ ,$$

where f is the non-linear rectification function.

Different functions are used to perform non-linear rectification. Some of them are illustrated Figure 2.1. In practice, the Rectified Linear Unit (ReLU) function and its derivatives (Leaky ReLU, Threshold ReLU, etc.) are mainly used since they provide the best results.

Pooling

Once the rectified feature maps obtained, it is paramount to reduce their dimensions in order to reduce the computational power of the network. However, the reduction of spatial dimensions must not come at the cost of loosing relevant information extracted in the feature maps.

To reach this goal, the pooling operation is introduced after non-linear rectification. The principle of the pooling operation is to first define spatial neighborhood of the feature map, i.e. divide the image into $w_p \times w_p$ windows. Then, an operation is applied on each spatial window, reducing its content to one value. The equation of a convolution layer with non-linear rectification (see equation (2.8)) and pooling becomes:

$$y_{\text{pooling}}(i, j) = f_p(y_{\text{rectified}}((i-1) \times w_p + 1 : i \times w_p, (j-1) \times w_p + 1 : j \times w_p) \ ,$$

$$\forall i = 1, ..., \left\lfloor \frac{H_r}{w_p} \right\rfloor \ , \forall j = 1, ..., \left\lfloor \frac{W_r}{w_p} \right\rfloor \ ,$$

where f_p is the pooling function, w_p is the window width and H_r and W_r are respectively the height and the width of the rectified feature map $y_{\text{rectified}}$.

There are two main pooling operations illustrated in Figure 2.6:

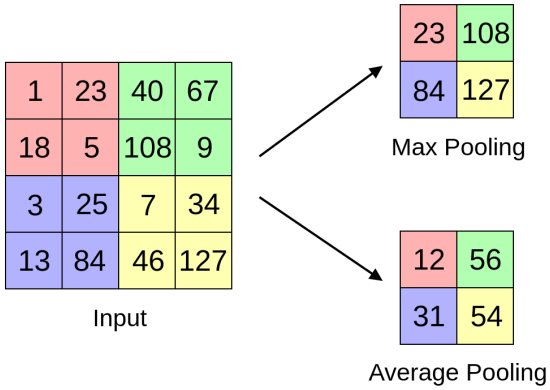

Figure 2.6: Illustration of the Max Pooling and Average Pooling operations on a given matrix. In this example, the different spatial neighborhoods are 2×2 windows (left) represented in red, green, blue and yellow. On the right, the result of the Max Pooling (top) and Average Pooling (bottom) operations are given for each spatial neighborhood.

- The Max Pooling selects the maximum value of each spatial neighborhood of the feature map. It provides the best results in practice since it is more likely to suppress the potential noise in rectified feature maps,
- The Average Pooling selects the average value of each spatial neighborhood of the feature map. It generally provides lower results than the Max Pooling operation since it takes the potential noise in rectified feature maps into account.

Pooling operations reduce the data spatial dimensions, which makes them easier to work with. By reducing the number of features to process, the computational efficiency of the network is increased and over-fitting is more likely to be limited. Pooling operations also make the network invariant to small transformations such as distortion or translation. Finally, pooling operations give an equivarient representation of the input matrix, i.e. invariant to the scale. In image processing, this implies that an object can be described the same way no matter if it is big or small.

To summarize up, this section reviews the different operations constituting a convolution layer. More specifically, it is the combination of convolution, non-linear rectification and pooling operations that makes CNN efficient tools to process images while increasing their computational efficiency. As a matter of fact, the described operations extract relevant features, characterizing efficiently all the objects of a same category, and invariant to the scale and to small transformations. It is important to note that, as the successive convolution operations describe the input data with higher-level features, the deeper the network is, the higher-level the extracted features will be.

2.3 Object detectors

Object detectors aim to detect objects in an image, i.e. give their nature and their position in it. The detection task consists in predicting a set of bounding boxes, each one containing an identified object. The predicted bounding boxes are actually the 4

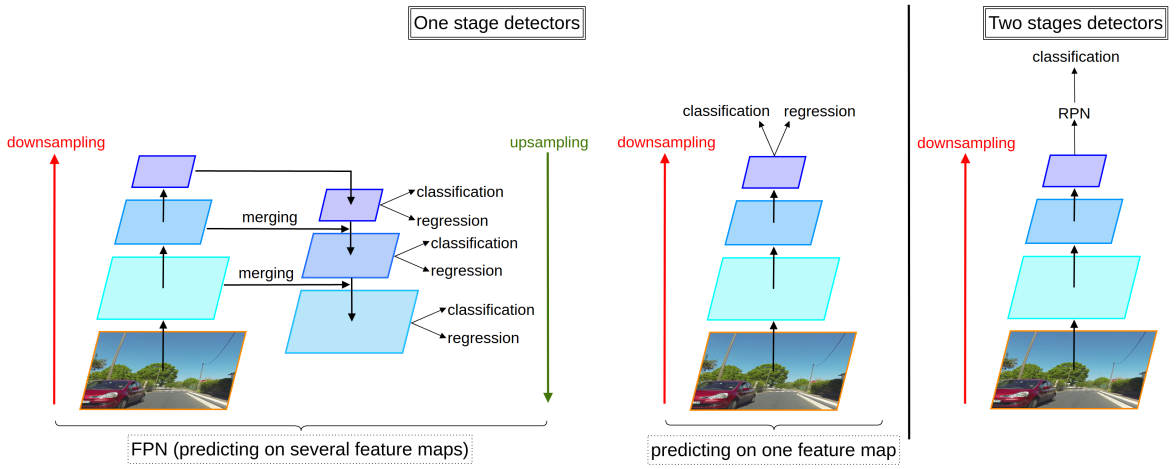

Figure 2.7: Illustration of one stage and two stages detectors. One stage detectors can make their predictions on one or several (using a Feature Pyramid Network (FPN)) feature maps. The two stages detectors are composed of two subnetworks, the first one (a Region Proposal Network (RPN)) predicting Regions Of Interest (ROI) and the second one that classifies them.

coordinates of the area containing the object, (x_{min}, y_{min}, x_{max}, y_{max}) or (x_{min}, y_{min}, width, height), and the nature of the object (its classification). In this section, the different tasks to predict the bounding boxes and evaluate their quality are described.

2.3.1 The different architectures

There are two main architectures to achieve object detection. The first one is the two stages detector which is composed of two successive networks to make its predictions. The second one is the one stage detector which is an end-to-end architecture enabling to make predictions. The different object detectors architectures are illustrated Figure 2.7 and are detailed in the following sections.

Two stages detectors

The first step of two stages detectors consist in predicting a set of ROI from a feature map using a Region Proposal Networks (RPN) [23]. These ROI are the coordinates of the detected objects and are unlimited in number. The second step consists in using a classification network over these ROI, i.e. giving the nature of the object. The classification network is a MLP, which last layer contains a neuron for each class. Each of these neurons returns the probability of the object to belong to the corresponding class and the sum of these probabilities is one. The highest probability determines the class of the object and is called the detection score.

The two stages detectors are usually slower since they are composed of two networks but more precise. However, because of the downsampling caused by the successive convolution and pooling layers, their performances on small objects can be altered.

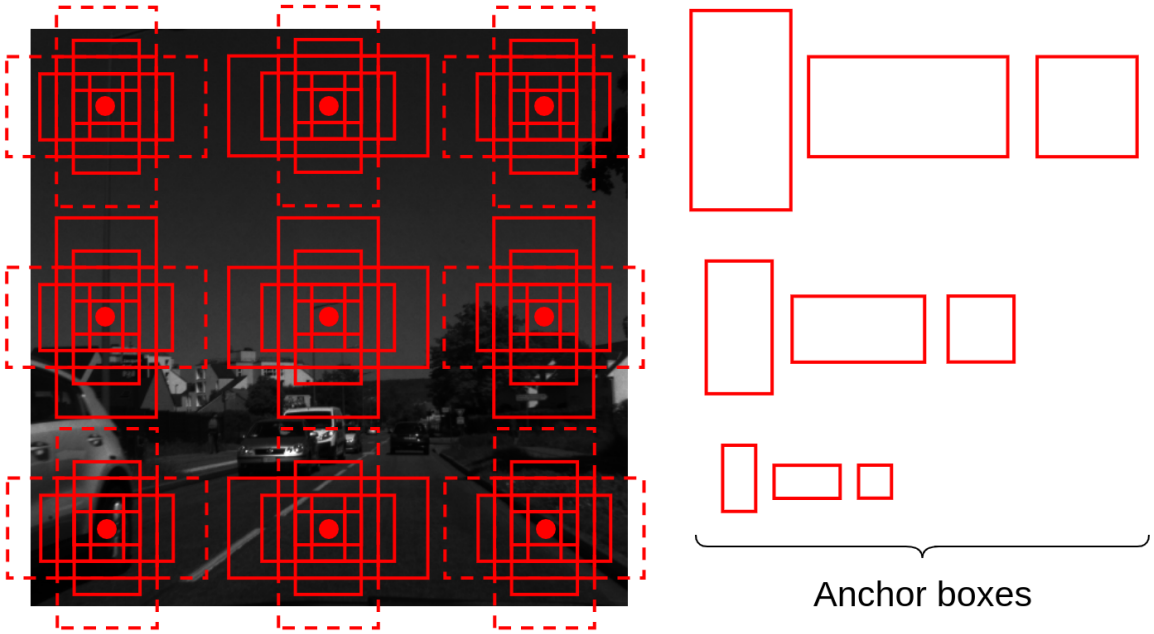

Figure 2.8: Example of anchor boxes initialization. In full lines, the final anchors boxes and in dashed lines, the anchors boxes suppressed because they are not within the image. The different anchor boxes predicted at each location (red dot) are illustrated on the right.

One stage detectors

One stage object detectors are initialized with anchors boxes over the image. The most relevant anchors boxes are selected and corrected to become the predicted bounding boxes. As a matter of fact, this initialization step is paramount since it is difficult for CNN to predict bounding boxes from scratch.

Anchor boxes of different sizes and shapes are initialized at regular positions of the image and only the ones within the images are selected. They cover all the image, enabling to detect all the objects it contains. Figure 2.8 illustrates the anchor boxes initialization in an image. In practice, there are thousands of anchor boxes initialized in an image. This number vary depending on the neural network.

Once the anchors boxes initialized, two tasks are performed at the same time. The first one consists in predicting from one or several feature maps the final coordinates of each anchor box. To predict on several feature maps, a Feature Pyramid Networks (FPN) [74] is usually used. It enables to provide a representation of the same object at different scales to reinforce the feature extraction. The second one is the characterization of each anchor box's content while providing a score similarly to the one stage detector. These two tasks are respectively called regression and classification. When predicting on one feature map, the subsampling caused by the several convolution and pooling layers may lead to a loss of information. This implies a loss of performances when it comes to small objects. FPN [74], however, prevent loss of information and improve small objects detection as the predictions are made at different scales. At the end of the process, one prediction is made for each anchor box (or several predictions, i.e. a prediction at each scale, for each anchor box in case of a FPN), resulting in

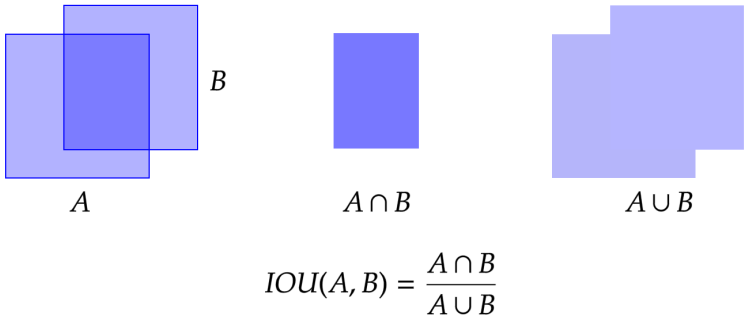

Figure 2.9: Illustration of the Intersection Over Union (IOU) operation. In dark blue, the intersection of bounding boxes A and B and in pale blue the union of bounding boxes A and B.

thousands of bounding boxes. A filtering step is thus necessary to select only the most relevant ones.

The filtering process depends on the Intersection Over Union (IOU) between the predicted bounding box and the ground truth bounding box. This operation is the following:

$$IOU(A, B) = \frac{A \cap B}{A \cup B} \ ,$$

where A and B are two bounding boxes, $\cap$ is the intersection operation and $\cup$ is the union operation.

The IOU $\in [0, 1]$ enables to measure how close are two bounding boxes. IOU $= 1$ if the two bounding boxes are the same and IOU $= 0$ is the two bounding boxes do not intersect. This operation is illustrated in Figure 2.9.

There are different algorithms to perform the filtering step. The two main algorithms are the Non-Maximum Suppression (NMS) [75] and the soft-NMS [76] and are detailed in algorithm 1. Their action is illustrated Figure 2.10. Their filtering processes are the followings:

- The NMS filters the proposals to keep only the most relevant ones. To achieve this goal, the propositions with the highest scores are selected among similar ones and the others are suppressed. This algorithm shows limits when there are two objects to be detected, close to one another.

- The soft-NMS enables to keep all the proposals. It selects the proposition with the highest scores among similar ones and modifies the scores of the others. This algorithm enables to detect close objects at a cost of keeping bounding boxes detecting the same object if they have a high detection score.

Note that algorithm 1 presents the common part of NMS and soft-NMS algorithms in black. The lines in red are used in the NMS algorithm while the ones in blue are used in the soft-NMS algorithm. The function f used in the soft-NMS algorithm is the following:

Algorithm 1 NMS and soft-NMS algorithms

Require: $\mathbf{B} = b_1, ..., b_n$ the initial detection boxes, $\mathbf{S} = s_1, ..., s_n$ their corresponding detection scores, N_t the NMS threshold.

```
D ← {}
while B ≠ ∅ do
    m ← argmax(S)
    M ← b_m                                   ▷ Select element with highest score
    D ← D ∪ M
    B ← B − M                                 ▷ Remove element from proposals
    for b_i in B do
        if IOU(M, b_i) ≥ N_t then
            B ← B − b_i                       ▷ NMS
            S ← S − s_i
        end if
        s_i ← s_i × f(IOU(M, b_i))            ▷ soft-NMS
    end for
end while
return D, S
```

$$f(IOU(\mathbf{M}, b_i)) = \exp\left(\frac{-IOU(\mathbf{M}, b_i)}{\sigma}\right) ,$$

where σ is the Gaussian weight of the soft-NMS algorithm.

2.3.2 Loss functions

The loss functions used in object detectors aim to both evaluate if the objects are correctly classified and how far are the predicted coordinates from the ground truth. This is the reason why a classification and a regression loss are defined to evaluate each of these tasks. The final loss of the network can be written:

$$\mathcal{L} = \varpi \mathcal{L}_{\text{classification}} + \zeta \mathcal{L}_{\text{regression}} ,$$

where $\mathcal{L}$ refers to the loss of the network, $\mathcal{L}_{\text{classification}}$ to the classification loss and $\mathcal{L}_{\text{regression}}$ the regression loss and ϖ and ζ their respective weight factors.

The MSE (see equation (2.2)) can be used as a regression loss since it evaluates the distance between two points. As for the classification loss, the Cross Entropy (CE) loss [77] or the Focal Loss (FL) [3] can be used since the network performs multiclass prediction. The CE loss equation is the following:

$$CE(p, z) = \sum_{i=1}^{c} \log(p_{t_i}) ,$$

$$p_{t_i} = \begin{cases} p_i & \text{if } z_i = 1 \\ 1 - p_i & \text{else} \end{cases} ,$$

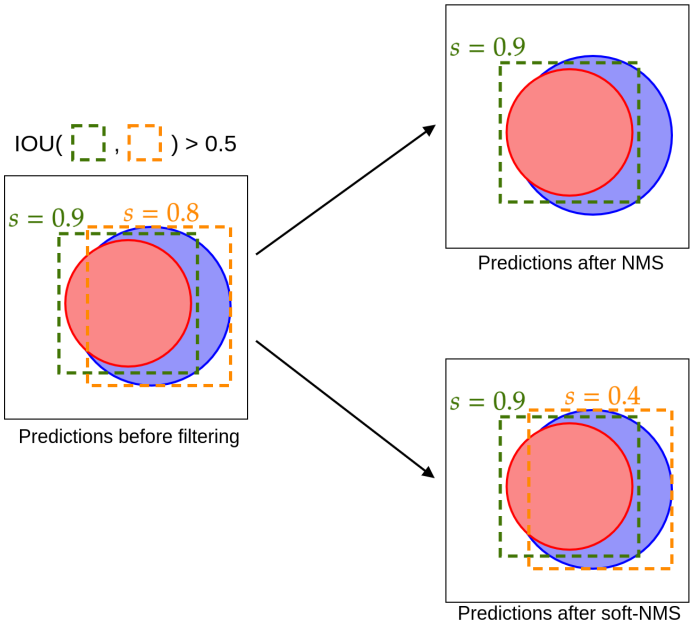

Figure 2.10: Illustration of the filtering process using the NMS algorithm (top right) and soft-NMS (bottom right).

where $p = \begin{bmatrix} p_1 & p_2 & ... & p_i & ... & p_c \end{bmatrix}$ is the vector of the estimated probabilities of belonging to each class i, which sum equals to 1, $p_t = \begin{bmatrix} p_{t_1} & p_{t_2} & ... & p_{t_i} & ... & p_{t_c} \end{bmatrix}$ is the vector evaluating the distance of p to the ground truth $z = \begin{bmatrix} z_1 & z_2 & ... & z_i & ... & z_c \end{bmatrix}$ ($z_i = 1$ if the object belong to class i and $z_i = 0$ else $\forall i = 1, ..., c$) and c the number of classes.

The CE loss is limited when the classes are unbalanced. As a matter of fact, since it sums the errors of all predictions, the network can choose to sacrifice a minor class performances to improve the overall performance. The FL, however, is designed to down-weight well classified examples to focus the network's training on the hard misclassified ones. Its equation is the following:

$$FL(p, z) = \sum_{i=1}^{c} (1 - p_{t_i})^{\beta} \log(p_{t_i}) \ , \tag{2.9}$$

where $\beta \in [0, 5]$ the down-weighting factor. The closer p_{t_i} is to 1 (good classification), the smaller $(1 - p_{t_i})$ will be, which down-weight the loss for well classified examples.

2.3.3 Evaluation metrics

Once the network is trained, the predictions are evaluated using specific metrics. Object detectors aim to increase the number of good predictions while decreasing the number of false predictions. To achieve this goal, two metrics are defined: the precision and the recall. The precision measures the quantity of correct predictions among all the predictions. The recall measures the quantity of correct predictions with regards to the

ground truth. To make good predictions, an object detector must have a good balance between its precision and its recall metrics. Their equations are the followings:

$$\text{precision} = \frac{TP}{TP + FP} \ ,$$

$$\text{recall} = \frac{TP}{TP + FN} \ ,$$

where TP stands for True Positives and is the number of results predicted positive by the model that are actually positive, FP stands for False Positives and denotes the number of results predicted positive by the model that are actually negative and FN stands for False Negatives and is the number of results predicted negative by the model that are actually positive.

In multiclass object detectors, the positive and negative predictions are determined by a threshold for the IOU between the ground truth and the predicted bounding boxes. If the IOU between the prediction and the ground truth is above the threshold, the prediction is considered positive, it is considered negative otherwise. To measure the quality of the predictions, the precision and recall are calculated while varying the IOU threshold. The precision recall-curve, which gives indications about the balance between the precision and recall of the predictions, is obtained by plotting the precisions with regards to their respective recalls. For more interpretability on this curve, the Average Precision (AP) metric is created. The AP is the precision averaged across all unique recall levels and has the following formula:

$$p_{\text{interp}}(r) = \max_{\bar{r}:\bar{r}\geq r}(p_s(\bar{r}))$$

$$AP = \frac{1}{11} \sum_{r\in[0,0.1,...,1]} p_{\text{interp}}(r)$$

where $r \in [0, 0.1, ..., 1]$ is the set of eleven equally spaced possible recall levels, $\bar{r}$ is all recall values that exceed a given one r, $p_{\text{interp}}(r)$ is the maximum precision for which recall is greater or equal to r and $p_s(\bar{r})$ is the precision at the corresponding recall $\bar{r}$. This formula is illustrated Figure 2.11.

The AP is computed for each class separately. In order to measure the overall performance of object detectors, the Mean Average Precision (mAP) over all the possible classes is computed. The formula of the mAP is the following:

$$mAP = \frac{1}{c} \sum_{i=1}^{c} AP_i \ , \tag{2.10}$$

where c is the number of classes and AP_i is the AP of class i.

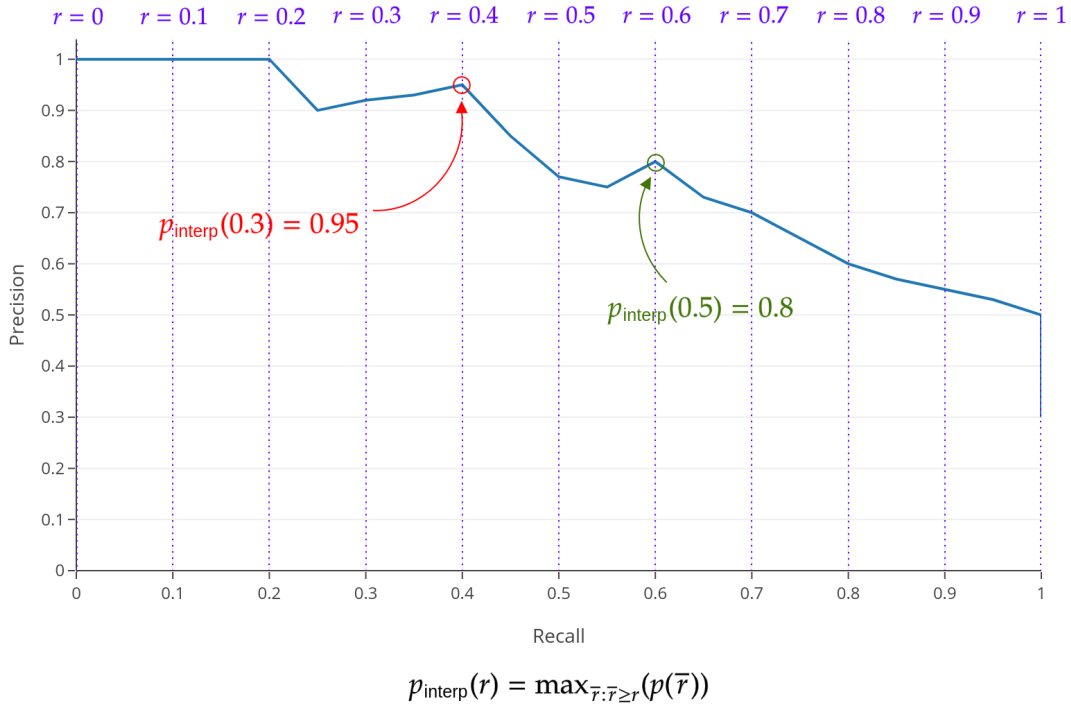

Figure 2.11: Illustration of the computation of $p_{\text{interp}}(r)$ on the precision-recall curve.

2.4 Cycle-Consistent Generative Adversarial Networks

CycleGAN [78] aim to achieve the image-to-image translation task. Given two domains X and Y, unpaired image-to-image translation is the task of learning the mapping functions $M_{XY} : X \rightarrow Y$ and $M_{YX} : Y \rightarrow X$ using unpaired samples $x_i \in X$ with $i \in [1..N]$ and $y_j \in Y$ with $j \in [1..M]$.

The two mapping models, M_{XY} and M_{YX}, are learnt by combining the objective function of the standard Generative Adversarial Networks (GAN) [66] with a Cycle-Consistency loss function. The adversarial cost related to the GAN serves for training the models to generate samples that match the target domain distribution, while the Cycle-Consistency cost ensures that the learned models are able to correctly reconstruct an original image (of the source domain) from a generated one.

Formally a GAN is composed of a generative model $G : Z \rightarrow X$ which maps a known distribution p_Z, usually normal or uniform, to the unknown distribution p_X of the samples and a discrimination model $D : X \rightarrow [0, 1]$. Both the generator and the discriminator are fully CNN. The generator G attempts to fool the discriminator D, which in turn tries to distinguish a real sample from a sample generated by the model G. Learning a GAN amounts to solve the following problem:

$$G^*, D^* = \arg\min_G \max_D \mathcal{L}_{\text{GAN}}(D, G) \ ,$$
$$\text{with } \mathcal{L}_{\text{GAN}}(D, G) = \mathbb{E}_{x \sim p_X}\Big[\log(D(x))\Big] + \mathbb{E}_{z \sim p_Z}\Big[\log(1 - D(G(z)))\Big] \ ,$$

where $\mathbb{E}$ refers to the expectation.

For its part, CycleGAN learns the two models M_{XY} and M_{YX} by using unpaired real samples $x \in X$ and $y \in Y$ respectively drawn according to the (unknown) distributions p_X and p_Y as input. It also learns two discrimination networks $D_X : X \to [0,1]$ and $D_Y : Y \to [0,1]$ able to detect generated samples from real ones in the domains X and Y respectively. CycleGAN relies on the Least-Squares variant of GAN [79] and considers the following adversarial costs:

$$\mathcal{L}_{\text{GAN}}(D_Y, M_{XY}) = \mathop{\mathbb{E}}_{y \sim p_Y}\left[(D_Y(y)-1)^2\right] + \mathop{\mathbb{E}}_{x \sim p_X}\left[D_Y(M_{XY}(x))^2\right] \ ,$$
$$\mathcal{L}_{\text{GAN}}(D_X, M_{YX}) = \mathop{\mathbb{E}}_{x \sim p_X}\left[(D_X(x)-1)^2\right] + \mathop{\mathbb{E}}_{y \sim p_Y}\left[D_X(M_{YX}(y))^2\right] \ .$$

In order to ensure the cyclic consistency, i.e. both the compositions $M_{XY} \circ M_{YX}$ and $M_{YX} \circ M_{XY}$ are identity functions, a ℓ_1 reconstruction error term is devised for the mapping models:

$$\mathcal{L}_{\text{reco}}(M_{XY}, M_{YX}) = \mathop{\mathbb{E}}_{y \sim p_Y} ||y - M_{XY}(M_{YX}(y))||_1 + \mathop{\mathbb{E}}_{x \sim p_X} ||x - M_{YX}(M_{XY}(x))||_1 \ .$$

Gathering all these elements leads to the objective function:

$$\begin{aligned}\mathcal{L}_{\text{CycleGAN}}(D_X, D_Y, M_{XY}, M_{YX}) = \mathcal{L}_{\text{GAN}}(D_Y, M_{XY}) + \\ \mathcal{L}_{\text{GAN}}(D_X, M_{YX}) + \eta \mathcal{L}_{\text{reco}}(M_{XY}, M_{YX}) \ ,\end{aligned} \tag{2.11}$$

where $\eta > 0$ is an hyper-parameter that controls the influence of the reconstruction term. Training a CycleGAN consists in solving, via alternate gradient descent, the following minmax problem:

$$M_{XY}{}^*, M_{YX}{}^*, D_X{}^*, D_Y{}^* = \arg \min_{\substack{M_{XY} \\ M_{YX}}} \max_{\substack{D_X \\ D_Y}} \mathcal{L}_{\text{CycleGAN}}(D_X, D_Y, M_{XY}, M_{YX}) \ . \tag{2.12}$$

The full learning procedure of a CycleGAN is sketched in Algorithm 2 and illustrated in Figure 2.12.

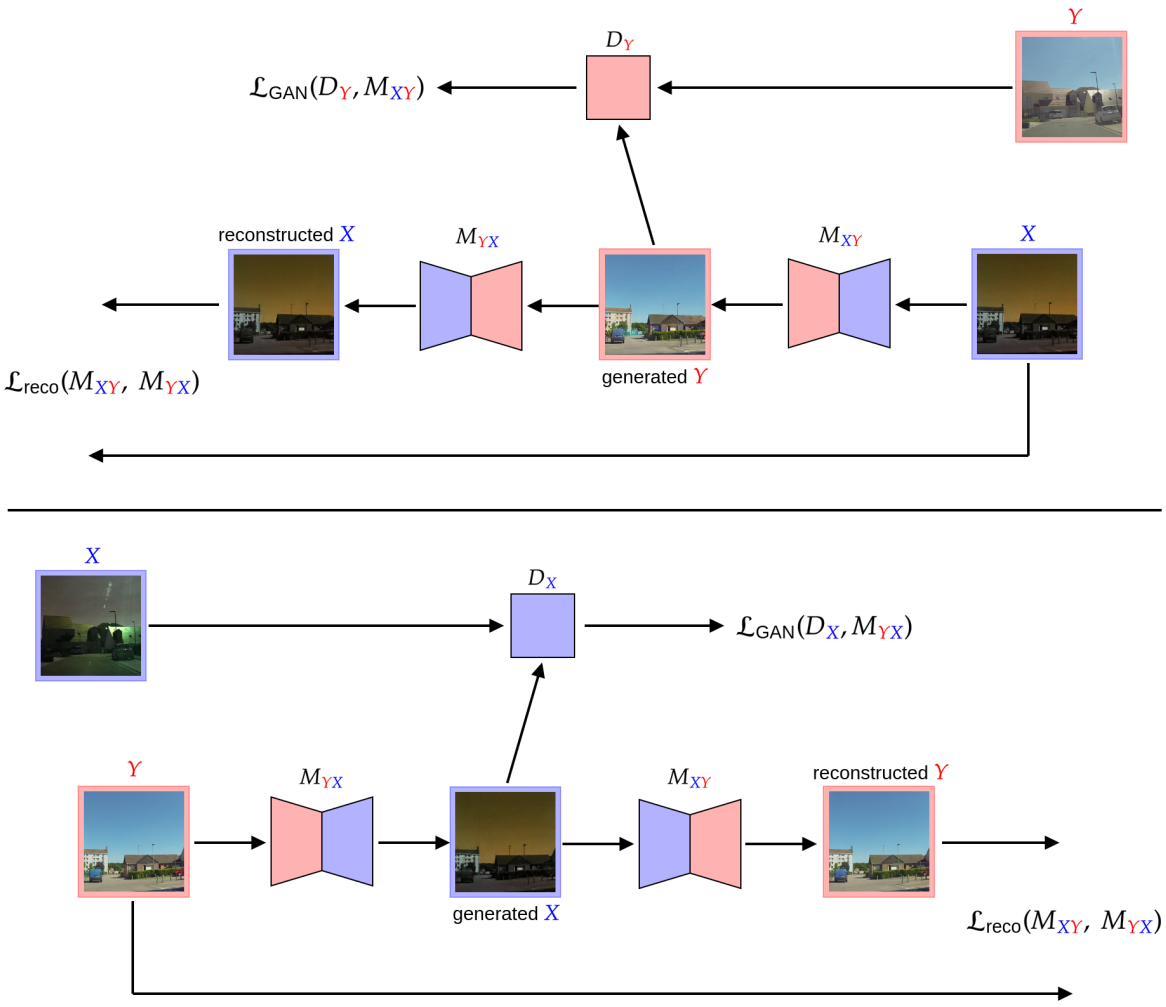

Figure 2.12: Illustration of the image-to-image translation process using a CycleGAN. On top row, the translation from domain X to domain Y is illustrated. On bottom row, the translation from domain Y to domain X is illustrated.

Algorithm 2 CycleGAN training algorithm

Require: X and Y two unpaired datasets, M_{XY} and M_{YX} the mapping networks, D_X and D_Y the discrimination models, m the mini-batch size

while stopping condition is not met **do**

sample a mini-batch $\{x_i\}_{i=1}^{m}$ from X

sample a mini-batch $\{y_i\}_{i=1}^{m}$ from Y

update D_X by stochastic gradient descent of

$\sum_{i=1}^{m}(D_X(x_i)-1)^2+(D_X(M_{YX}(y_i)))^2$

update D_Y by stochastic gradient descent of

$\sum_{i=1}^{m}(D_Y(y_i)-1)^2+(D_Y(M_{XY}(x_i)))^2$

sample a mini-batch $\{x_i\}_{i=1}^{m}$ from X

sample a mini-batch $\{y_i\}_{i=1}^{m}$ from Y

update M_{XY} by stochastic gradient descent of

$\sum_{i=1}^{n}(D_Y(M_{XY}(x_i))-1)^2+\eta(||x_i-M_{YX}(M_{XY}(x_i))||_1$

$+||y_i-M_{XY}(M_{YX}(y_i))||_1)$

update M_{YX} by stochastic gradient descent of

$\sum_{i=1}^{n}(D_X(M_{YX}(y_i))-1)^2+\eta(||x_i-M_{YX}(M_{XY}(x_i))||_1$

$+||y_i-M_{XY}(M_{YX}(y_i))||_1)$

end while

2.5 Summary

This chapter gives the necessary background on Deep Learning to understand the experiments carried out in this thesis. The basic concepts of Deep Learning are first described, starting from the functioning of an artificial neuron. The combination of several artificial neurons to constitute a MLP is then detailed. This concept comes with the backpropagation algorithm, enabling the network to adjust its parameters to their optimal value during the training process. The description of the convolution operation and how it is integrated to convolution layers is also given. Convolution layers are used to CNN, which are the most efficient architectures to process images. From these basic concepts, the elements constituting object detectors are presented. The different architectures performing object detection are first described. The loss functions used in object detectors, quantifying the error, are also detailed, followed by the evaluation metrics, giving indications on the quality of the predictions. Finally, the CycleGAN, performing image-to-image translation, is explained. Deep architectures combined with multimodal data are widely used in the autonomous driving fields. Their ability to provide an accurate and relevant road scene analysis in real time make them a powerful tool to achieve this task. The different deep architectures of the state of the art are reviewed in the following chapter.

Chapter 3

Literature review

Contents

3.1 Introduction

Object detection is a fundamental task to perform an accurate road scene analysis. Recent progresses, such as the ChauffeurNet [80] architecture behind the Waymo car or the HydraNet [81] architecture behind Tesla Autopilot, enable a high autonomy when the vision is clear. However, some improvements still need to be done when the visibility is altered. To overcome this limitation, non-conventional sensors are introduced to see beyond color-based vision, yet without guarantee an accurate road scene analysis in every situation.

In this chapter, we first go through the use of polarimetric imaging in the literature. The applications of this modality give an intuition on its use to enhance road object detection in adverse weather. The role of the non-conventional sensors currently used in the literature to overcome an altered visibility is then reviewed. The specificities of infrared imaging, LiDAR and Radar, enhancing road scenes analysis in complex situations, as well as their limits are detailed.

To improve road object detection, these non-conventional modalities are often coupled with deep architectures. Following this stream, the different off-the-shelf object detectors of the literature are described to provide an overview of each pipeline. Since

the last part of this thesis focuses on multimodal fusion, this chapter explains the different fusion schemes by sketching the different architectures of the literature. Finally, different datasets are presented to visualize the performances of the mentioned pipelines, performing color-based common object detection or providing different modalities to allow autonomous navigation. These datasets are used in the literature as benchmarks, enabling a fair comparison between different architectures performing the same task. The literature reviewed in this chapter lays the foundations of the work achieved in this thesis, including the datasets constitution, road object detection in polarimetric scenes in adverse weather and multimodal color-based and polarimetric fusion.

3.2 Non-conventional modalities

Non-conventional modalities are becoming more and more popular in autonomous navigation. As a matter of fact, they provide information complementing conventional color-based vision, which is highly affected by visibility changes and lighting conditions. In this section, we first review polarimetric imaging, a non-conventional modality describing objects by their physical properties. The use of this modality in different fields is detailed to understand the intuition behind its application in autonomous navigation. The non-conventional sensors currently used to enhance road scene analysis in complex situations are then detailed, as well as their limits.

3.2.1 Polarimetric imaging

In the polarization formalism described in section 1.2, it is mentioned that this modality is able to describe objects by their reflective properties which are object specific. Thanks to the strong features provided by polarimetric imaging, it is widely used in the biomedical field. As a matter of fact, since defective cells do not have the same composition as healthy ones, polarization enables pathological diagnosis, such as cancer, at an early stage [19, 82, 83, 84, 85, 86, 87, 88].

Another application of this non-conventional modality consists in enhancing perception under water [89, 90, 91, 92, 93, 94, 95, 96], where color-based sensors fail to distinguish the different elements of the scene. The interpretation of satellite images is also improved using Synthetic-Aperture Radar (SAR) polarimetric imaging [97, 98, 99, 100, 101, 102, 103], due to the different reflective properties of land surfaces, buildings and rivers among others. Polarimetric imaging is also popular in dehazing algorithms [20, 104, 105, 106, 107, 108] since the light reflected from the haze particles can be easily distinguished from the light reflected from the rest of the scene.

Polarimetric imaging also provides an accurate depth map [21, 109, 110, 111, 112] of a scene, which has proved to enhance indoor autonomous navigation. In the same vein, polarization helps 3D objects reconstruction [113, 114, 115, 116, 117] by using the information provided by the angle of polarization of the reflected wave.

These applications have lead to recent interest in this modality to enhance autonomous driving. Huber *et al.* [118] use polarimetric images to detect glare in road scenes, induced by the presence of water or ice on the road. Fan *et al.* [22] employ polarimetric images to detect cars with a higher accuracy than conventional images.

Blanchon *et al.* [119] apply polarization properties to perform an efficient segmentation of highly reflective areas on the road such as puddles. Li *et al.* [120] operate on polarimetric images to perform road segmentation in a scene regardless of the illumination conditions to enhance autonomous navigation.

These applications, i.e. the dehazing properties of polarimetric images and their ability to detect highly reflective surfaces, are encouraging towards the use of this modality to enhance road scene analysis in adverse weather conditions.

3.2.2 Road scenes analysis in complex situations

Complex situations, such as strong illumination or low visibility, are very challenging for autonomous driving since conventional sensors do not provide reliable information in such conditions [9, 10]. Some non-conventional sensors, such as LiDAR, Radar or infrared imaging have proved to be more efficient than the regular RGB camera to address road scene analysis when the visibility is altered. In this section, the different modalities used to improve road scene analysis are reviewed, starting from the sensors used to enhance night vision, followed by the means to overcome adverse weather conditions or strong illuminations.

Night vision Images provided by color-based cameras vary a lot with the scene illumination as they are based on human vision. During nighttime, the color or shape of road objects is altered since the lighting of the scene is provided by streetlights or headlights of other vehicles if any. Hence, the provided information is not reliable, which is an issue to guarantee the safety of the most vulnerable road users.

To overcome this issue, infrared imaging is introduced, especially to enhance pedestrians, cyclists or motorcyclists detection [11, 121, 122, 123, 124, 125, 126, 127] since it relies on an entity's temperature, which is usually higher than the background for living beings. Animals can be another obstacle in road scenes that are efficiently detected using infrared imaging [128, 129, 130]. However, infrared imaging often suffer from low resolution and pedestrians can be confounded by hot parts of vehicles [124].

Radar is another sensor that provides information regarding the geometry of the scene or the distance and speed of other road users in real time, no matter the lighting conditions. It is able to efficiently estimate the road curvature in very low visibility conditions [131, 132, 133]. It is also able to efficiently detect potholes [134] or the speed of other road users when the visibility is very low [135]. However, Radars are affected by the thermal noise of the external environment [15], as a consequence, the detections include noise.

LiDAR can also accurately analyze road scenes at nighttime since they provide a 3D representation of the scene that is not affected by the scene illumination. They therefore enable to efficiently detect road users [16, 136, 137, 138] or animals [139]. LiDAR also perform recognition of roads in low visibility [140, 43, 141]. However, LiDAR lack of precision when detecting long distance objects since they often suffer of poor resolution [142].

Adverse weather conditions and strong illuminations Adverse weather conditions are also challenging situations an autonomous vehicle must address. As a matter of fact, weather changes, such as fog or rain alter the scene visibility, as well as the color or shape of objects. On the other hand, a bright sun induces strong reflections on the scene, a sharp backlight and glare. These perturbations highly affect conventional color-based sensors and result in a non reliable road scene description [9].

Infrared imaging also finds applications in road scene analysis in adverse weather conditions. It enables a detection of road users at a larger range than visible cameras under fog, rain and snow [12, 9, 143, 144, 145, 146]. It also enables to detect traffic signs and landmarks more efficiently under fog, rain and snow [12, 147] and the presence of water and ice on the road [148, 149, 150, 151]. However infrared imaging shows limits in adverse weather conditions such as heavy fog or rain because they can modify the thermal footprint of vulnerable road users such as pedestrians, cyclists and motorcyclists [13].

LiDAR is another sensor robust to lighting changes. This property implies that the previously reviewed tasks to enhance night vision are still valid to overcome strong reflections and glare. However, LiDAR is weakly reflective on wet road surfaces [42], causing a lack of measures. On the other hand, adverse weather causes undesired measurement points due to snowflakes or drops misinterpretations [17, 152, 153].

Similarly to the LiDAR sensor, Radar are invariant to illumination conditions, hence, their improvements to enhance night vision are still valid in strong illuminations and invariant to glare. On top of that, Radars perform a more reliable road users detection under snow, rain and fog than visible cameras [14, 154, 155, 156]. However, Radar are likely to be affected by the thermal noise of adverse weather [15].

Summary of non-conventional sensors In this section, we reviewed the different sensors that can be an added value to enhance the scenes' perception in complex situations. However, even if infrared imaging, LiDAR and Radar overcome color-based perception in adverse weather or during nighttime, they all show limits when the visibility is very low. This limitation reminds that autonomous navigation in complex situations is still a grand challenge to address. Hence, other non-conventional modalities need to be explored as an alternative to the ones used in the literature.

3.3 Object detection

Performing an accurate object detection is paramount to provide a reliable road scene analysis. Deep architectures nowadays have shown outstanding performances and are the most adapted to achieve this task. There are plenty of different pipelines detecting objects since this research field is getting more and more popular. This section first focuses on the most competitive architectures for common object detection. Then, the fusion architectures aiming to fuse different modalities to perform autonomous navigation are described. Finally, the different datasets used as benchmarks to evaluate the performances of the different architectures are detailed.

3.3.1 Deep architectures

In this section, the deep architectures performing object detection are reviewed. The origin of the detection task is first presented through the early deep architectures. Off-the-shelf one stage and two stages object detectors, as well as their enhanced version are also detailed and summarized in Figure B.1.

Two stages detectors

Two stages detectors consist in dividing the detection task into two subtasks, which are finding ROI first and then classify each of these ROI. In this section, the early methods are first reviewed, leading to the concept of backbone networks. Finally, the different end-to-end two stages object detectors are presented.

• **Early methods** DetectorNet [157] is the first attempt to address the object detection problem. Object are detected using a coarse-to-fine mask regression approach followed by a regression step. One DetectorNet is trained per class contained in the dataset which makes it computationally expensive. Its 7 layers architecture, including 5 convolutional and 2 fully connected layers, inspired several other early two stages object detectors. Among them, we can cite R-CNN [158] and Spatial Pyramid Pooling Network (SPPNet) [159], which introduce the concept of priors by using the selective search algorithm [160] to extract 2000 region proposals, each one processed by a 7 layer architecture before being classified by a linear Support Vector Machine (SVM) [161]. DeepMultiBox [162] is a less computationally expensive approach since it predicts a fixed number of bounding boxes (100 or 200), which represent potential objects, using a DetectorNet-based architecture. A score is associated to each bounding box, enabling the NMS algorithm [75] to reduce the proposals before classifying them by a similar CNN. OverFeat [163] similarly to DeepMultiBox predicts the four coordinates of a bounding box, before classifying it with a similar architecture.

• **From early methods to backbones of end-to-end architectures** Deeper architectures later provide more accurate alternatives to the 7 layer architecture, following the early two stages principles. We can cite Visual Geometry Group network (VGG) [164], stacking 13 (VGG16) and 16 (VGG19) convolutional layers, providing higher level features to describe an image. GoogLeNet [165] rethinks the convolution operation by introducing the Inception block, enabling an optimized deeper architecture while keeping the computational budget constant.

These architectures are nowadays used as backbones of end-to-end object detectors and their performances are evaluated on ImageNet [63]. Among them, we can find Residual Network (ResNet) (ResNet-50, ResNet-101, ResNet-152) [166] which use residual information of previous layers to ease the training of deep architectures. Their variant, ResNeXt [167], increases the cardinality of the network instead of going deeper for better performances, while Densely Connected Convolutional Networks (DenseNets) connect each layer to every other layer in a feed forward fashion. MobileNets [168] provide light weight DNN by dividing a standard convolution layer into a depthwise convolution layer followed by a pointwise convolution layer. Squeeze-and-Excitation

Networks (SENets) [169] generalize extremely well across challenging datasets, since they adaptively recalibrate channel-wise feature responses by explicitly modeling interdependencies between channels. Neural Architecture Search Networks (NASNets) [170] propose a method that learns the model architecture directly on the dataset of interest. Finally, EfficientNets [171] balance the network depth, width and resolution according to the computational resources to increase performances. An illustration of the core operations of these architectures can be found in Figure B.2.

• **End-to-end architectures** Fast R-CNN [65] marks an important milestone in the object detection field. It is the first architecture trained end-to-end with a multi-task loss, performing both bounding boxes classification and regression, enabling a faster and more accurate detection. Multiple ROI priors and an image are input to the network, which is composed of a VGG16 backbone and fully connected layers. The output contains a score for each bounding box, which is a softmax probability for each class and for the background, and four refined coordinates.

Faster R-CNN [23] improves the previously described architecture by introducing the concept of RPN. This fully convolutional network enables to share full-image convolutional features with the detection pipeline. It acts as a sliding window on feature maps which produces 9 anchor boxes at each location as priors. A score indicating if the anchor contains an object is associated to each anchor box. By merging RPN and Fast R-CNN, the resulting ROI are then refined and classified before being filtered by the NMS algorithm. The whole pipeline is trained end-to-end, and enables to process several frames per second since the region proposal step is nearly cost-free.

The Faster R-CNN pipeline is nowadays used as a basis to two stages object detectors. Some modifications have lead to improvements on the detection task on the MS COCO detection challenge [4]. We can cite Region-based Fully Convolutional Network (R-FCN) [172] which provides a more accurate and efficient detection by replacing the costly fully connected layers by convolutional ones. Faster R-CNN combined with SNIPER [173] processes context regions around ground truth instances, adaptively generated based on the scene complexity and at a larger scale than regular feature maps, instead of every pixel in an image, enabling a higher detection accuracy. Path Aggregation Network (PANet) [174] identifies that the way information propagates in neural networks plays an important role in its performances and proposes a Bottom-up Feature Pyramid to propagate efficiently semantically strong features. Cascade R-CNN [24] introduces a multistage detection sub-network, with different IOU thresholds, to reduce the false positive detection rate, later improved by Hybrid Task Cascade (HTC) [175], performing cascade refinement jointly on the classification and regression tasks. Global Context Network (GCNet) [176] introduces the non-local convolution operation (see Figure B.2), capturing long-range dependencies by aggregating information from other positions to a query position. TridentNet [177] uses dilated convolutions [178] to generate scale-specific feature maps with a uniform representational power. Grid R-CNN [179] replaces the offset regression branch by a grid-guided mechanism providing probability heatmaps giving the location of the bounding boxes. Libra R-CNN [180] revisits the standard training process by balancing the information flow in each resolution of the FPN (see Figure B.3), leading to more discriminative features.

D2Det [181] improves the localization of the bounding boxes by predicting multiple offsets for an object proposal. Dynamic R-CNN [182] adapts automatically the IOU threshold used for the label assignment criteria and the parameter of the regression loss function, based on the statistics of proposals during training. Faster R-CNN combined with Task-aware Spatial Disentanglement (TSD) [183] bridges the gap between sensitive location for localization and classification, to remedy the spacial misalignment in the sibling heads hurting the training process. More recently, DetectoRS [184] combines Recursive FPN (see Figure B.3) with switchable Atrous convolutions (dilated convolutions), enabling to look twice or more at the input features at different scales.

One stage detectors

One stage detectors, unlike two stages detectors, enable to make the region proposals and classification tasks in a single forward pass. This pipeline design leads to faster predictions but sometimes at the cost of less accurate predictions.

The single-shot object detection paradigm is first formulated by the MSC-MultiBox architecture [185]. The intuition behind this network is to predict both the four coordinates and the confidences score for each of the 11 anchor boxes priors at each scale. However, even if this architecture provides a higher computational efficiency, its performances do not overcome the two stages methods of the state of the art. DenseBox [186] remedies this limitation by introducing a fully convolutional network, based on VGG19. This architecture takes an image pyramid, i.e. the same image at different scales (see Figure B.3), sets an anchor box at every four pixels of each image of each scale, then refines each bounding box while predicting a detection score indicating if the box contains or not an object. The final predictions are obtained by performing a NMS filtering step.

The YOLO network [1] marks a milestone in single-shot object detectors and is illustrated in Figure 3.1. This architecture enables to perform object detection at 45 frames per second (fps). It divides the input image into a $S \times S$ grid (usually 7×7), used to predict both the class probability map of each cell and a fixed number of bounding boxes, i.e. the four coordinates and its confidence score of containing an object. The final detections are determined by linking the predicted bounding boxes to their respective class from the class probability map. YOLO9000 [187] improves this pipeline by incorporating batch normalization to converge faster and by using anchor boxes priors using the k-means algorithm [188]. It enhances small objects detection and overall accuracy by predicting the anchor boxes offsets instead of the bounding boxes from scratch and performing multi-scale training. YOLOv3 [189] introduces a deeper network with a multilabel approach by using independent logistic classifiers in order to address more complex domains such as the Open Images dataset [190]. It achieves higher performances by predicting bounding boxes at 3 different scales using a FPN [74]. The Adaptively Spatial Feature Fusion (ASFF) [191] enhances YOLOv3 by palliating the inconsistency across the scales of the FPN, by learning weights assessing a spatial importance to each feature map. YOLOv4 [192] achieves an even higher accuracy by including features to improve CNN, such as weighted residual connections and Cross-Stage-Partial-connections [193] to its backbone.

Another branch of one stage detectors is introduced by SSD [2], a fully convolu-

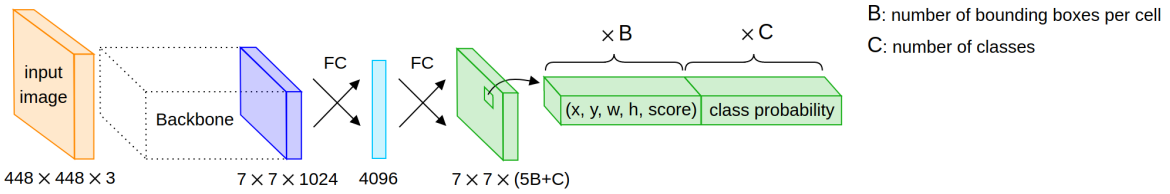

Figure 3.1: The architecture of YOLO [1]. Here FC stands for Fully Connected layer.

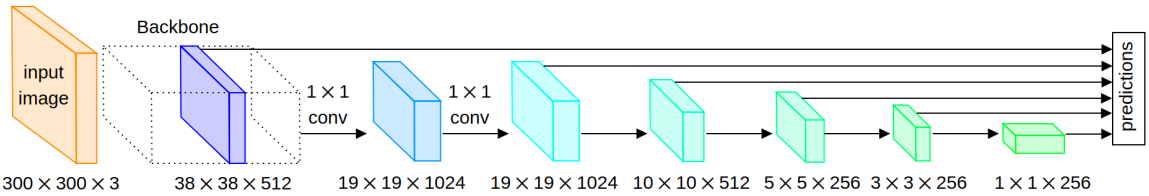

Figure 3.2: The architecture of SSD [2].

tional architecture which outperformed Faster R-CNN on MS COCO, illustrated in Figure 3.2. It originally uses VGG16 as a backbone, extended by 10 convolution layers used to perform multi-scale prediction. It sets prior anchor boxes at equally spaced locations on each feature map, for which coordinates offsets and a softmax probability on all the classes are predicted. A filtering step, using the NMS algorithm is performed to obtain the final predictions. Deconvolutional Single Shot Detector (DSSD) [194] adds deconvolutional layers to introduce additional large scale context in object detection, enhancing small objects detection. RefineDet [195] improves this architecture by introducing inter-connected anchor refinement and an object detection module, reducing the search space for the classifier by filtering out negative anchors. M2Det [196] introduces the concept of Multi-Level Feature Pyramids, an enhanced version of FPN (see Figure B.3), providing richer feature maps to improve object detection at different scales.

The RetinaNet network [3] is another off-the-shelf one stage architecture, illustrated in Figure 3.3. It is the first architecture that couples its backbone, originally a ResNet-50, with FPN, enhancing multi-scale prediction. Similarly as the SSD architecture, anchor boxes are initialized at each pyramid level. A regression and a classification subnetworks respectively predict the offset coordinates and softmax probabilities for each anchor box in a fully connected fashion. Once again the NMS algorithm enables to get the final bounding boxes. This architecture also introduces the FL (see equation (2.9)), designed to focus the training process on hard misclassified examples. An enhanced version of RetinaNet includes the Soft-Anchor-Point Object Detection algorithm (SAPD) [197], assigning attention weights to anchor boxes to avoid suppressing the boxes with a more precise location but a lower score. The Feature Selective Anchor-Free (FSAF) module [198], however, consists in plugging a convolution layer in parallel of each subnet providing feature maps, respectively activated by regions containing objects and the class it belongs, to reinforce the detection of small objects missed by anchor boxes. Fully Convolutional One Stage object detector (FCOS) [199] on the other

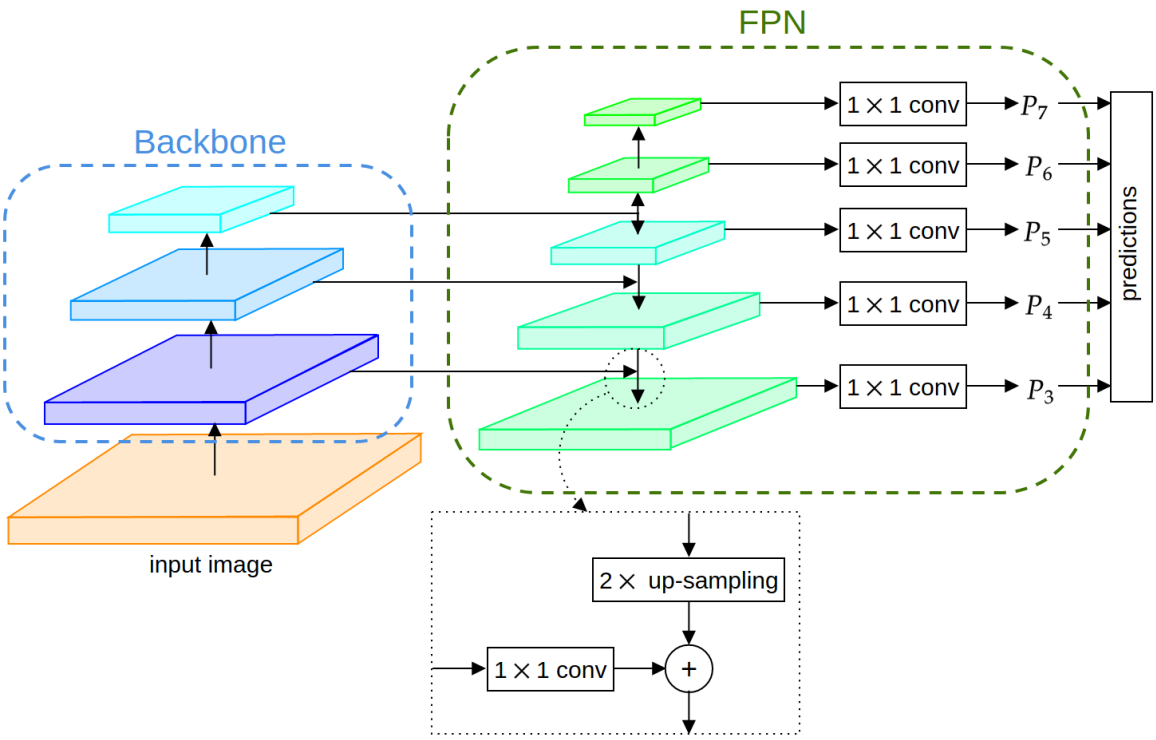

Figure 3.3: The architecture of RetinaNet [3].

hand is fully convolutional and anchor box free, performing a per-pixel bounding box prediction, coupled with a center-ness operation, suppressing the low quality detected bounding boxes. The Multiple Anchor Learning (MAL) approach [200] generates an anchor bag for each object at each feature map pyramid level, to jointly optimize classification and localization subnetworks. More recently, FreeAnchors [201] proposes a learning-to-match approach for object detection, by constituting bags of anchors for each object and selecting the most relevant one using the maximum likelihood approach instead of an IOU threshold.

Another family of single shot detectors comes with the design of EfficientDet [202]. This pipelines follows the principle of most other single shot object detectors, i.e. a backbone providing the image features with anchor boxes refined and classified. This architecture involves a Bi-directional Feature Pyramid Network (BiFPN) (see Figure B.3), allowing easy and fast multi-scale feature fusion. It also includes a component scaling method that uniformly scales the resolution depth and width of the networks to enable an optimized architecture.

More recently, a new family of detectors has emerged, based on the transformers encoder-decoder's architecture [203]. The DEtection TRansformer (DETR) [25] views object detection as a direct set prediction problem, removing anchor generation and the NMS post-processing step. Its set-based global loss forces unique predictions via bipartite matching. This pipeline is composed of a CNN backbone giving a set of image features, flatten as a sequence and fed to a transformer encoder. This sequence is fed to a transformer decoder, which uses a fixed number of learnt object queries to determine

where to look for an object in the encoded sequence. Each object query is transformed into an output embedding by the decoder, which are independently processed by a feed forward network predicting if it contains an object and if so, predicts the class and the coordinates of the bounding box. Deformable DETR [204] introduces a deformable attention module to focus on a small set of key points sampling instead of the whole image, to ease the convergence. Swin Transformer [205] produces a hierarchical feature representation of the image, enabling to model various scales of an instance.

Objects as a set of points

Another paradigm of object detection is introduced by the CornerNet architecture [26]. It gets rid of priors by detecting objects as a pair of key points, the top left and the bottom right corners of the bounding box. This architecture is based on 2 Hourglass networks [206] put end-to-end as a backbone, enabling to capture both global and local features of an image. The backbone is extended by two prediction modules, respectively for the top left and the bottom right corners. Each prediction module is composed of the localization sub-module, containing the heatmap locating the corners of the different object categories, the offset sub-module to adjust the corners' position and the embedding sub-module determining if two corners describe the same object. A post-processing algorithm is required to filter and assemble the set of corners. CenterNet [27] improves the accuracy by predicting a triplet of key points, including the center of the bounding box instead of just two corners. ExtremeNet [28] on the other hand predicts four extreme points (top most, bottom most, left most and right most) and one center point to refine the prediction. Matrix Nets [207] combine CornerNet with an enhanced version of the FPN, providing a scale and aspect ratio aware architecture (see Figure B.3). Corner Proposal Network for Object Detection (CPNDet) [208] uses the basis of CornerNet to constitute a two stages object detector, constituting ROI from the extracted corners and assigning a label to each proposal by a standalone classification stage.

RepPoints [29] is another approach aiming to represent objects as a set of sample points used for localization and recognition in a multi-stage fashion. It uses a FPN as a backbone, where each pyramidal level predicts a set of RepPoints. Each set of RepPoints is given to an afterwards pipeline containing several regression steps refining the previous proposition. The penultimate set of RepPoints obtained after multiple regressions is then classified while a last regression step is performed. The final propositions are obtained after using the NMS filter as a post-processing step. RepPoints v2 [123] improves this architecture by using a verification branch, predicting a Corner Point and a foreground heatmaps, reinforcing the regression step.

3.3.2 Deep fusion for autonomous navigation

The previous section reviewed the different object detectors. However, they process one modality at the time, mainly RGB color-based images. To efficiently describe a road scene, several modalities are needed to provide further information on the scene [209, 210]. These past few years, people have come up with deep architectures aiming to optimally fuse the different modalities. In this section, the different fusion schemes

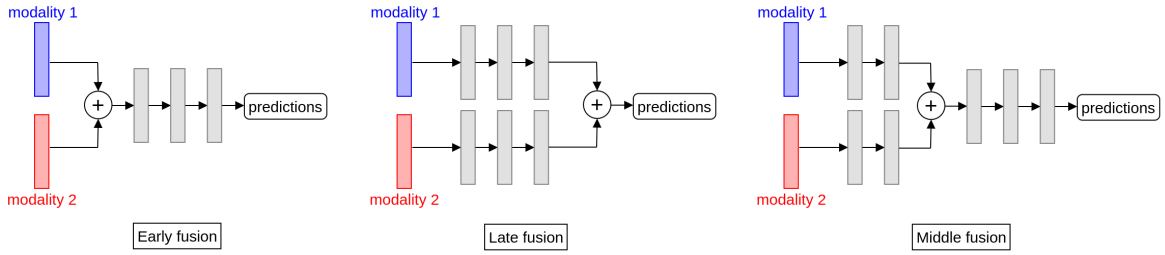

Figure 3.4: Illustration of the different fusion schemes.

and the resulting architectures for autonomous navigation purposes are described. The three different fusion pipelines are reviewed, including Early, Late and Middle fusion, which are sketched in Figure 6.1.

Early fusion

Early fusion consists in fusing raw or pre-processed data at the early stage of the network, enabling to fully exploit raw data by learning their joint representation. It results in lighter architectures and thus requires lower computation resources. However, fusing the modalities at an early stage results in inflexible networks, that need to be retrained completely when one modality needs to be replaced by another. On top of that, this pipeline is sensitive to sensor breakdown and data misalignment.

Because of the drawbacks of this fusion scheme, it is not often used for autonomous navigation. Liu *et al.* [30] explore an Early, a Middle and a Late fusion schemes, based on the Faster R-CNN network. These architectures fuse RGB and thermal images to enhance pedestrian detection and compare their performances. However, the Early fusion pipeline does not provide the best results. In the same vein, Wagner *et al.* [211] also compare the Early and Late fusion pipelines to fuse RGB and thermal images to enhance pedestrian detection and leaded to similar conclusions.

Pfeuffer *et al.* [212] use Faster R-CNN as a basis to build an Early, a Middle and a Late fusion pipelines to fuse RGB images with a LiDAR depth map to enhance road object detection in adverse weather. Once again, the Early fusion scheme provides lower results than the two others. Sindagi *et al.* [213] propose Multimodal Voxel Network (MVX-Net), fusing RGB images and LiDAR Front View (VF) maps at an early stage based on VoxNet [214]. PointFusion is explored, which consists in projecting the 3D point cloud onto the image using a known calibration matrix. They also explore VoxelFusion, which involves projection of 3D voxels[1], onto the RGB images at a later stage which results in a lower accuracy.

Late fusion

Late fusion consists in processing each modality separately and combining the decision output of each domain. Late fusion has a high flexibility since only one network needs

[1] A voxel is to a pixel what 3D is to 2D. While a point cloud contains continuous observation of the scene, the voxel mesh is a dense volume modeling these observations in a discrete space.

to be retrained to replace one modality. It is also robust to sensor breakdown, since the other modalities provide valid predictions to backup the whole pipeline. However, this pipeline is computationally expensive because it requires multiple networks.

Mees *et al.* [215] fuse RGB images with a depth map, using two Fast R-CNN in the Late fusion fashion to enhance indoor pedestrian detection. The modalities are fused in a supervised way using a gating network finding the adapted weight of each modality. Takumi *et al.* [216] combine the detection of RGB, Near Infrared (NIR), Middle Infrared (MIR) and Far Infrared (FIR) modalities using four YOLO architectures to enhance road object detection. Shin *et al.* [217] present RoarNet, which first extracts a set of 2D bounding boxes on RGB images and then finds one or several corresponding areas in the 3D point cloud, which are used to perform 3D object detection. Following the same principle, Du *et al.* [218] projects the LiDAR point cloud towards the corresponding RGB image to refine the point cloud area on which the 3D object detection is made. Wang *et al.* [219] also use 2D regions detected on RGB images to generate frustums, i.e. portions of the pyramid generated from the 2D region, containing LiDAR point-clouds of the corresponding area. Each frustum is processed using a PointNet [220] and the results are concatenated and fed to a fully convolutional network which classifies and predicts the coordinates of the 3D bounding box.

Middle fusion

Middle fusion is a compromise between early and late fusion. It consists in first processing each modality separately and combining them or sharing connections at intermediate layers of the networks. This property of the network allows to learn different representation of the cross modalities at different scales, resulting in a highly flexible network. However, even if Middle fusion is the best of both worlds, it requires a lot of neural architecture search to find the optimal way to fuse intermediate layers. In practice, Middle fusion is mainly used since it achieves the best detection results.

Even though Middle fusion is often used to perform 3D object detection, since most of them are designed to process LiDAR point clouds, some of them perform exclusively 2D object detection. Schneider *et al.* [221] combine RGB image and their depth maps using an architecture composed of an adapted GoogLeNet, processing RGB data, inter-connected with an adapted Network in Network (NiN) [222], processing depth data. This architecture provides fused representations of these two modalities, modeling complex intra-domain dependencies, enabling to enhance 2D object detection. Guan *et al.* [223] propose an illumination-aware multispectral deep neural network, fusing RGB and thermal images, to enhance pedestrian detection. The two modalities are first processed separately and fed into a illumination-aware network that attributes the optimal weight to each modality with regards to the lighting conditions. Chadwick *et al.* [224] fuse Radar and RGB images to enhance distant vehicle detection. The two modalities are processed separately first, using ResNet-18 blocks, and concatenated to be processed a second time, using ResNet-18 blocks, to perform 2D detection at several scales. Bijelic *et al.* [225] fuse stereo RGB images, gated NIR images, FIR images, Radar and LiDAR front view point clouds to enable 2D detection under fog without seeing fog during the training process. An entropy is estimated for each modality and

fed at several stages of the corresponding SSD network. The pipelines processing each modality are inter-connected to provide intra-modalities representations. Nabati *et al.* [32] propose a Radar Region Proposal Network (RRPN) that fuses RGB images with Radar signals to increase the speed and accuracy of 2D detections. They first map the Radar coordinates towards the RGB image and use these points to generate anchor boxes on the RGB image to reduce the priors. The regions are then processed by an off-the-shelf object detector to get the final detections. Kim *et al.* [226] propose a deep fusion architecture to enhance 2D car detection when randomly inducing light variations or occlusion on the scene. This architecture is composed of two SSD adapted for each modality, containing interconnected layers performing a Gated Information Fusion on the two modalities to perform object detection at several scales.

The previous paragraph reviewed the architectures performing 2D detection, now we describe the pipelines achieving 3D object detection. Qi *et al.* [227] introduce Frustum PointNets, fusing RGB information with the corresponding LiDAR point cloud. They use a 2D object detector to extract 2D regions from RGB images, which are used to extract the corresponding frustums to perform Point Cloud segmentation to find the 3D instances. Xu *et al.* [228] present PointFusion that leverages both RGB images and LiDAR FV point cloud information to perform 3D object detection. A ResNet-50 and a PointNet are respectively used to process separately the cropped RGB image and its corresponding LiDAR point cloud. The processed data are fed to a dense Fusion network, providing an optimal intra-domain representation to perform 3D object detection. Ku *et al.* [229] propose an Aggregate View Object Detection (AVOD), fusing LiDAR Bird Eye View (BEV) point clouds with RGB images to perform 3D object detection. Each modality is first processed separately by a fully convolutional network and the obtained features are cropped and resized before being fused to perform object detection. The proposals and the separately processed features are fused a second time to provide the final detections. Wang *et al.* [230] fuse BEV LiDAR point clouds with RGB images using two RetinaNet with interconnected intermediate layers to perform 3D object detection. Chen *et al.* [31] propose a Multi-View 3D network (MV3D), combining RGB images, LiDAR BEV point clouds and LiDAR FV point clouds to perform 3D object detection. The first stage consists in processing separately the three modalities and predicting the 3D ROI from the LiDAR BEV map. These 3D ROI are projected towards the other modalities. A deep fusion network is used to combine region-wise features for each ROI in order to predict its class and its oriented 3D box regression. Casa *et al.* [33] propose IntentNet to predict road users' intention and their respective 3D bounding boxes from the raw LiDAR BEV point cloud and the map of the environment. The two modalities are processed separately by a fully convolutional backbone and concatenated to be processed by another backbone in order to perform 3D object detection and predict their intention. Liang *et al.* [231] fuse RGB images and BEV LiDAR point clouds to perform an accurate 3D object detection. The RGB image is processed by several ResNet-18 networks in order to extract multi-scale feature maps combined with a multi-scale fusion operation. The fused maps are then processed by continuous fusion layers and added at different stages of the LiDAR BEV pipeline to perform 3D object detection. Dou *et al.* [232] propose SEG-VoxelNet, which fuses RGB images with LiDAR BEV point clouds. The network first segments the RGB

Dataset	PASCAL VOC 2007 [235]	PASCAL VOC 2012 [236]	ILSVRC [63]	MS COCO [4]	Open Images V4 [190]
Images	9,963	11,540	476,688	123,287	1,910,098
Classes	20	20	200	91	600
Instances	24,640	31,561	534,309	886,284	15,440,132
Instances/image	2.47	2.73	1.12	7.19	8.08

Table 3.1: Large scale datasets for common object detection summary.

input while another branch aligns the LiDAR point cloud towards the RGB image. The segmentation and the alignment maps are then combined and fed to another network to perform 3D object detection. Liang *et al.* [233] fuse BEV LiDAR point clouds with RGB images and a depth map to perform 2D and 3D objects detection. They densely fuse layers of the two backbones of each modality at several scales to process intra-modal representations of data during the whole training process. Vora *et al.* [234] proposes PointPainting, fusing 3D LiDAR FV point cloud with RGB images. The RGB images are first segmented and a PointPainting operation is made to paint the corresponding LiDAR points into the segmented class. An off-the-shelf 3D object detector is then applied to the painted map to perform object detection.

3.3.3 Datasets

In the previous sections, we reviewed the object detectors of the literature as well as the different fusion architectures. Their performances are either evaluated on a dataset specifically hand-crafted for the task but more often on suitable public datasets used as benchmarks. In this section, the different datasets used to evaluate the object detector performances are first presented. Then, the ones used for road object detection are detailed. The performances of the different pipelines on these datasets are also given.

Large scale datasets for common object detection

Over the years, several datasets have been constituted to address the object detection challenge at a large scale. Because these datasets contain diverse images with several common classes, they are used as benchmarks to evaluate the performances of object detectors. The different datasets' properties are summarized up in Table 3.1.

PASCAL VOC The Pattern Analysis, Statistical Modelling and Computational Learning Visual Object Classes (PASCAL VOC) [235] was organized annually from 2005 to 2014, providing a benchmark to evaluate the performances of object detection algorithms. Even though this dataset was enlarged along the years, the PASCAL VOC 2007 [237] and the PASCAL VOC 2012 [236] datasets are the ones used as benchmarks to evaluate object detectors.

PASCAL VOC 2007 is composed of 9,963 images containing 24,640 annotated objects as bounding boxes. The objects belong to 20 classes, including person, animal, vehicle and indoor objects. PASCAL VOC 2012 contains 11,540 images containing 31,561 annotated objects as bounding boxes from the same 20 classes.

ILSVRC The ILSVRC [63] started in 2010 and was originally designed to perform large scale image classification. In 2012, several images are labelled with bounding boxes to perform object detection. The latest version of the dataset is composed of 476,688 images with 534,309 instances labelled with bounding boxes. The objects are from 200 categories, including vehicles, person, food, animals, furniture and household electrical appliances among others.

MS COCO In 2014, a larger dataset, the MS COCO [4] is released, challenging object recognition. Complex everyday scenes containing common objects in their natural context are collected for this purpose. MS COCO is composed of 123,287 images with 886,284 instances labelled with bounding boxes. The objects are from 91 categories, including accessories, vehicles, road signals, animals, sport, food and cutlery, furniture and household electrical appliances.

Open Images The Open Images Dataset [190] is the latest large scale dataset for multilabel object detection. The images have been collected without a predefined list of class names to enable natural class statistics and avoid design bias. The Open Images V4 [238] is the version of this dataset used as a benchmark to evaluate object detector performances. It contains 1,910,098 images with 15,440,132 instances annotated with bounding boxes. The objects belong to 600 classes, which are grouped into parent classes, enabling multilabel detection. The classes include vehicles, food, animals, furniture, person, buildings and road signs among others.

Performances of the different architectures In the previous paragraphs, we presented the properties of the baseline datasets for object detection at a large scale. We now review the performances of the different architectures presented in section 3.3.1. These performances are reported in Table 3.2, with regards to the Graphic Processing Unit (GPU) or Tensor Processing Unit (TPU) used to run the tests.

Datasets for road object detection

We previously reviewed the large datasets used to address common object detection at a large scale. In this section, we focus on datasets for road object detection. Several datasets are used for road scene analysis, but only the ones labelled with at least 2D or 3D bounding boxes are mentioned in this section. Therefore, AppoloScape [239], Mapillary [240] or Cityscapes [241] are not detailed since they perform semantic or instance segmentation. The datasets' properties are summarized up in Table 3.3.

Daimler Monocular Pedestrian Detection Enzweiler *et al.* [242] present the Daimler Monocular Pedestrian Detection dataset in 2008, a benchmark for pedestrian detection. It is composed of daytime images of road scenes in a restricted area, acquired with a color-based camera. It contains 21,790 images with 56,492 labelled pedestrians.

Caltech Pedestrian Dollár *et al.* [243] introduce the Caltech Pedestrian dataset in 2009, a pedestrian detection benchmark. This dataset provides daytime images of

Pipeline	Performances on datasets (mAP)				Memory usage	fps
	PASCAL VOC 2007	PASCAL VOC 2012	ILSVRC	MS COCO		
DetectorNet [157]	29.7	-	-	-	-	-
R-CNN [158]	58.5	53.3	-	-	test: 1 GPU*	0.08
SPPNet [159]	59.2	-	-	-	test: 1 GTX Titan GPU	2.62
DeepMultiBox [162]	29.2	-	-	-	test: GPU*	0.5
OverFeat [163]	-	-	29.9	-	test: 1 K20x GPU	0.5
Fast R-CNN [65]	66.9	65.7	-	19.7	test: 1 K40 GPU	3.3
Faster R-CNN [23]	69.9	67.0	-	21.5	test: 1 K40 GPU	5
R-FCN [172]	79.5	77.6	-	34.9	train: 8 K40 GPU, test: 1 K40 GPU	5.9
SNIPER [173]	-	-	-	46.1	train: 8 GPU V100, test: 1 GPU V100	5
PANet [174]	-	-	-	47.4	train: multi-GPU*	-
Cascade R-CNN [24]	-	-	-	42.8	train: 8 Titan Xp GPU test: 1 Titan Xp GPU	7.1
HTC [175]	-	-	-	47.1	train: 16 Titan Xp GPU, test: 1 Titan XP GPU	2.1
GCNet [176]	-	-	-	48.4	train: 8 GPU∗	-
TridentNet [177]	-	-	-	48.4	train: 8 GPU*	-
Grid R-CNN [179]	-	-	-	43.2	train: 32 Nvidia Titan Xp GPU	-
Libra R-CNN [180]	-	-	-	43.0	train: 8 GPU*	-
D2Det [181]	-	-	-	50.1	train: 8 GPU*	-
Dynamic R-CNN [182]	-	-	-	49.2	test: 1 RTX 2080TI GPU	13
TSD [183]	-	-	-	51.2	train: 16 GPU*, test: 1 Tesla V100 GPU	4.9
DetectoRS [184]	-	-	-	51.3	-	3.9
YOLO [1]	63.4	57.9	-	-	test: 1 Titan X GPU	45
YOLO9000 [187]	76.8	73.4	-	21.6	-	67
YOLOv3 [189]	-	-	-	33.0	test: 1 Titan X GPU	**78**
ASFF [191]	-	-	-	43.9	train: 4 Tesla V100 GPU, test: 1 Tesla V100 GPU	29.4
YOLOv4 [192]	-	-	-	43.5	test: 1 Tesla V100 GPU	65
SSD [2]	71.6	74.9	**43.4**	26.8	test: 1 Titan X GPU	19
DSSD [194]	81.5	80.0	-	33.2	train: 4 P40 GPU, test: 1 Titan X	6.6
RefineDet [195]	**85.8**	**86.8**	-	41.8	train: 4 M40 GPU, test: 1 Titan X GPU	24.1
M2Det [196]	-	-	-	41.0	train: 4 Titan X GPU, test: GPU V100	11.8
RetinaNet [3]	-	-	-	39.1	train: 8 GPU*, test: 1 M40 GPU	5
SAPD [197]	-	-	-	43.1	train: 8 GPU*, test: 1 GTX 1080Ti	6.1
FSAF [198]	-	-	-	44.6	train: 8 GPU*, test: 1 Titan X GPU	2.8
FCOS [199]	-	-	-	44.7	-	-
MAL [200]	-	-	-	47.0	train: 8 GPU*	-
FreeAnchor [201]	-	-	-	47.3	train: 8 Tesla V100 GPU	-
EfficientDet [202]	-	-	-	52.2	train: 32 TPU, test: 1 Titan V GPU	3.8
DETR [25]	-	-	-	44.9	train: 16 V100 GPU, test: 1 V100 GPU	10
Deformable DETR [204]	-	-	-	52.3	test: 1 Tesla V100 GPU	19
Swin Transformer [205]	-	-	-	**58.7**	train: 8 GPU*, test: 1 V100 GPU	15.3
CornerNet [26]	-	-	-	42.1	train: 10 Titan X GPU, test: 1 Titan X GPU	4.1
CenterNet [27]	84.5	-	-	47.0	train: 8 Tesla V100 GPU, test: 1 Tesla P100 GPU	3.7
ExtremeNet [28]	-	-	-	43.7	train: 5 GPU*	3.1
MatrixNet [207]	-	-	-	47.8	train: 10 Titan Xp GPU	-
CPNDet [208]	-	-	-	41.6	train: 8 Tesla V100 GPU, test: 1 Tesla V100 GPU	26.2
RepPoints [29]	-	-	-	46.5	train: 4 GPU*	-
RepPoints v2 [123]	-	-	-	52.1	train: 8 GPU*, test: 1 Titan XP GPU	3.8

Table 3.2: Summary of large scale datasets for common object detection. The MS COCO style AP [4] is used for the performances on MS COCO. *: The GPU is not specified. The best detection scores and the highest fps are in bold. The reported scores and their associated fps are the highest in the original paper. Note that the V100 GPUs perform faster computations.

road scenes from a color-based camera. It is composed of 250,000 images, labelled with 350,000 2D bounding boxes of pedestrians.

KITTI The KITTI dataset [36] marks an important milestone in the autonomous driving field. It was released in 2012 and is nowadays still a benchmark to evaluate algorithms aiming to address challenging tasks such as 2D or 3D road object detection. It is the first large dataset providing outdoor scenes with data from multiple calibrated sensors, including 4 color-based cameras and 1 LiDAR. It is composed of 14,999 images with 80,256 2D bounding boxes and their 3D equivalent, from 8 different classes including car, pedestrian, van, truck, cyclist and tram.

Dataset	Sensor	Time of the day	Weather	Several cities	Images	Classes	2D labelled objects	3D labelled objects
Daimler [242]	1 RGB camera	Day	Clear	✘	21,790	1	56.492	✘
Caltech [243]	1 RGB camera	Day	Clear	✘	250,000	1	350,000	✘
KITTI [36]	4 RGB cameras 1 LiDAR	Day	Clear	✘	14,999	8	80,256	80,256
KAIST [35]	1 RGB camera 1 thermal camera	Day Night	Clear	✘	95,328	3	103,128	✘
BDD100K [244]	1 RGB camera	Day Night Dawn	Clear Overcast Snowy Rainy Cloudy Foggy	✓	100K	10	1,841,435	✘
Waymo [37]	5 RGB cameras 5 LiDAR	Day Night Dawn	Clear	✓	390,000	4	11.8M	12.6M
nuScenes [245]	6 RGB cameras 1 LiDAR 5 Radars 1 GPS 1 IMU	Day Night	Clear Rainy	✓	40,000	23	✘	1.4M
UA-DETRAC [246]	1 RGB camera	Day Night Dawn	Clear Rainy Cloudy	✘	140,000	4	1.21M	✘
Argoverse [247]	360°RGB images from 7 cameras 2 stereo RGB cameras 1 long range LiDAR	Day	Clear	✘	100 segments	17	✘	300,000
PandaSet	5 Wide-Angle RGB cameras 1 Long-Focus RGB camera 1 Mechanical Spinning LiDAR 1 Solid-State LiDAR	Day Night Dawn	Clear	✓	48,000	28	✘	1.4M

Table 3.3: Summary of datasets for road object detection.

KAIST Hwang *et al.* [35] propose the KAIST dataset in 2015, a multispectral pedestrian benchmark. This dataset provides daytime and nighttime images from a color-based and a thermal cameras. This dataset is composed of 95,328 images, labelled with 103,128 2D bounding boxes from 3 classes, including person, people and cyclist.

BDD100K In 2018, Yu *et al.* [244] released the Berkeley Deep Drive dataset containing 100K images (BDD100K) dataset, a larger dataset with more heterogeneous data than the KITTI dataset. As a matter of fact, it contains outdoors scenes in several places, under several weather conditions and at different times of the day. It is composed of 100,000 images with 1,841,435 2D bounding boxes from 10 classes including car, sign, light, person and bike among others.

Waymo Open dataset In 2020, Sun *et al.* [37] introduce the Waymo Open dataset. It contains a large number of data, collected in several places at different time of the day, using 5 color-based cameras and 5 LiDAR. This dataset contains 390,000 images labelled with 12.6 millions 3D bounding boxes and 11.8 millions 2D bounding boxes from 4 classes including vehicles, pedestrian, cyclists and signs.

nuScenes In 2020, Caesar *et al.* [245] present the nuScenes dataset. It contains a large number of road scenes in several places and different traffic conditions. The acquisitions campaigns were made at different time of the day under several weather conditions, using several synchronized sensors, including 6 color-based cameras, 1 LiDAR, 5 Radars, 1 Global Positioning System (GPS) and 1 Inertial Measurement Unit (IMU). This dataset is composed of 40,000 images labelled with 1.4 million of 3D bounding boxes from 23 classes including pedestrian, car, bus, animal, bicycle and motorcycle among others.

UA-DETRAC Wen *et al.* [246] propose the UA-DETRAC dataset in 2020. This dataset provides road scenes from different points of view, at different times of the day and under different weather conditions using a color-based camera. The dataset is composed of 140,000 of images, labelled with 1.21 million of 2D bounding boxes from 4 different classes, including van, bus, cars among others.

Argoverse Chang *et al.* [247] propose the Argoverse dataset in 2019. This dataset aims at tracking 3D road users using 360°images from 7 color cameras, 2 stereo color cameras and 3D point clouds from 1 long range LiDAR. This dataset is composed of more than 10,572 objects to track from 100 segments from 15 to 60 seconds length, labelled with 300,000 3D bounding boxes from 17 classes including vehicle, pedestrian, bicycle, motorcycle and animal among others.

PandaSet In 2020, the PandaSet dataset[2] is presented. This dataset provides road scenes from different points of view, at different times of the day and under different weather conditions using 5 Wide-Angle color cameras, 1 long focus color camera, 1 Mechanical Spinning LiDAR and 1 Solid-State LiDAR. This dataset is composed of 48,000 images labelled with 1.4 million of 3D bounding boxes from 28 classes including pedestrian, car, bicycle and motorcycle among others.

Performances of fusion architectures In the previous paragraphs, we review the properties of the baseline datasets for road object detection. We now review the performances of the different fusion architectures presented in section 3.3.2. The performances of the different architectures are summarized up in Table 3.4.

3.4 Summary

In this chapter, we first reviewed non-conventional modalities. The applications of polarimetric imaging in the literature are first listed to understand the intuition behind its use to enhance road scene analysis in adverse weather. The other non-conventional modalities of the literature, including LiDAR, infrared imaging and Radar, addressing challenging road situations, such as low visibility or adverse weather, are then detailed. Their applications to circumvent complex situations and their limits are described.

Object detection being a fundamental task to perform an accurate road scene analysis, the different architectures performing common object detection are then explained. Different paradigms are used to enable object detection. The two stages detectors are first sketched as object detectors are historically designed according to this paradigm. They are followed by one stages detectors, designed to increase the speed of the latter. Another paradigm consists in seeing object detection as a set of points in order to get rid of the computationally expensive bounding boxes priors.

Following the description of these pipelines, the different fusion schemes are detailed. They are often based on off-the-shelf object detectors or their backbones. The fusion schemes can be divided in three categories, the Early, the Late and the Middle fusion.

[2]https://scale.com/open-datasets/pandaset

The Early fusion scheme consists in fusing the desired modalities at the entry of the network. While it fully exploits the raw data of the two modalities, it is very sensitive to sensor breakdown or data misalignment. Late fusion, on the contrary, consists in processing each modality separately and combining their decision. While it is robust to sensor breakdown, the whole pipeline is computationally expensive. For this reason, the Middle fusion emerged, combining the best of these two worlds by starting to process each modality of the network and fusing them halfway. It has shown accurate results while saving some computational resources.

To compare the performances of these architectures, people have come up with different benchmarks. The final section of this chapter consists in reviewing the different datasets used to evaluate common object detectors. A summary of the detection scores achieved by the above-mentioned architectures on these benchmarks is also given. These results are followed by a review of the popular datasets used for road object detection, mainly composed of multimodal data. The performances of the different fusion schemes, achieved on the different benchmarks are also reminded.

Dataset	Architecture	Fusion	Modalities	Detection	$mAP_{0.5}$	$mAP_{0.7}$	car $AP_{0.5}$	car $AP_{0.7}$	pedestrian $AP_{0.5}$	cyclist $AP_{0.5}$	MR
KAIST	ConvNet [30]	Early	RGB thermal	2D	-	-	-	-	-	-	40.34
	ConvNet [30]	Middle	RGB thermal	2D	-	-	-	-	-	-	36.44
	ConvNet [30]	Late	RGB thermal	2D	-	-	-	-	-	-	40.77
	CaffeNet-based [211]	Early	RGB thermal	2D	-	-	-	-	-	-	52.20
	CaffeNet-based [211]	Late	RGB thermal	2D	-	-	-	-	-	-	42.32
	IADTN + IAMSS [223]	Middle	RGB thermal	2D	-	-	-	-	-	-	**28.93**
KITTI	Faster R-CNN-based [212]	Early	RGB LiDAR FV	2D	**76.2**	-	77.6	-	50.9	**72.9**	-
	Faster R-CNN-based [212]	Middle	RGB LiDAR FV	2D	75.6	-	78.0	-	51.1	71.4	-
	Faster R-CNN-based [212]	Late	RGB LiDAR FV	2D	75.8	-	77.7	-	**51.6**	71.2	-
	Frustum PointNets [220]	Middle	RGB LiDAR FV	2D	-	-	-	90.78	-	-	-
	AVOD [229]	Middle	RGB LiDAR FV	2D	-	-	-	89.73	-	-	-
	MV3D [31]	Middle	RGB LiDAR FV LiDAR BEV	2D	-	-	-	90.53	-	-	-
	R-DML [226]	Middle	RGB LiDAR FV	2D	-	-	**98.69**	-	-	-	-
	Multi-task multi-sensor [233]	Middle	RGB LiDAR FV	2D	-	-	-	**91.82**	-	-	-
	MVX-Net (VoxelFusion) [213]	Early	RGB LiDAR FV	3D	-	82.3	-	-	-	-	-
	MVX-Net (PointFusion) [213]	Early	RGB LiDAR FV	3D	-	**85.5**	-	-	-	-	-
	RoarNet [217]	Late	RGB LiDAR BEV	3D	83.71	-	-	-	-	-	-
	CNN [218]	Late	RGB LiDAR BEV	3D	87.69	57.63	-	-	-	-	-
	Frustum PointNets [220]	Middle	RGB LiDAR FV	3D	-	-	-	81.2	-	-	-
	Frustum ConvNet [219]	Late	RGB LiDAR BEV	3D	**89.02**	-	85.88	-	52.37	79.58	-
	PointFusion [228]	Middle	RGB LiDAR FV	3D	-	-	-	77.92	33.36	49.34	-
	AVOD [229]	Middle	RGB LiDAR FV	3D	-	-	-	84.41	50.8	64.0	-
	MV3D [31]	Middle	RGB LiDAR FV LiDAR BEV	3D	-	-	**96.02**	71.29	-	-	-
	Continuous Fusion [231]	Middle	RGB LiDAR BEV	3D	-	-	-	86.32	-	-	-
	SEG-VoxelNet [232]	Middle	RGB LiDAR BEV	3D	-	-	-	86.32	-	-	-
	Multi-task multi-sensor [233]	Middle	RGB LiDAR FV	3D	-	-	-	86.81	-	-	-
	PointPainting [234]	Middle	RGB LiDAR FV	3D	69.86	-	-	**92.45**	**58.7**	**83.91**	-

Table 3.4: Performances of the fusion architectures on KITTI and KAIST. The rows in red and in blue are respectively the architectures performing 3D and 2D object detection. MR stands for "missing rate". In bold, the best detection score for each task (column). $AP_{0.5}$ and $AP_{0.7}$ respectively stand for AP with a IOU threshold of 0.5 and 0.7.

Chapter 4

The datasets

Contents

4.1 Introduction

Road objects detection is a fundamental step to achieve a reliable road scene analysis. Based on the literature reviewed in Chapter 3, DNN are the best solution to accomplish this task in real time with a high accuracy. However, providing an accurate road scene analysis when the visibility is altered is one of the challenging research problems, limiting the deployment of autonomous cars and ADAS at a larger scale.

Using non-conventional modalities, combined with conventional ones are yet the best solutions to enhance road scene analysis when the conditions are not optimal [12]. Polarimetric imaging is a rich modality that enables to characterize objects not only by their color or shape but also regarding their physical information, invariant to visibility changes [18]. Thanks to this property, this non-conventional imaging could provide complementary information to color-based features to characterize road objects in every situation.

Deep architectures require rather large and diverse datasets to provide reliable results. Some adapted datasets, containing polarimetric images of road scenes in adverse weather conditions, are needed to carry out the experiments of this thesis. Such datasets are not publicly available and the first stage of this thesis is dedicated to their constitution. This chapter focuses on the different steps to constitute the polarimetric and multimodal datasets aiming to detect road objects in adverse weather. The different acquisition setups are first described, followed by the datasets constitution, including data sorting and labelling. We also go through the encoding of polarimetric images for machine learning, by describing the different polarimetric data formats constituted for the experiments. Finally, it is important to note that data acquisition, data sorting and data labelling, are expensive tasks. In order to circumvent this limitation, we address the problem of generating polarimetric images from color-based ones. It enables to provide a polarimetric version of flagship datasets of the literature, such as KITTI [36] or BDD100K [244], to carry out further experiments.

4.2 The acquisitions

Prior this work, there were no large and diverse enough public datasets containing polarimetric images of road scenes, labelled for object detection. This is the reason why data are collected, sorted and annotated for this purpose. In this section, the different acquisition campaigns are first described, followed by the constituted datasets.

4.2.1 Polarimetric acquisitions

The first acquisition campaign aims to collect polarimetric images of road scenes. The intuition behind this campaign is to palliate the lack of publicly available polarimetric data to analyze road scenes at a large scale. To this end, it is important to take several parameters into account. First the data must be both numerous and diverse enough to cover as many road situations as possible. Secondly, the viewing point of road scenes must be close to the driver's one.

To cope with all these requirements, a Polarcam Four Dimensions (4D) Technology polarimetric camera (see section 1.2 for further information) is placed behind the windshield at the height of the driver's eye, constituting an embedded system. This acquisition setup is similar to the one used for the BDD100K [244], which is a reference in autonomous driving. The acquisitions are made while driving in order to capture realistic road scenes. A large enough area is covered nearby Rouen-Normandy during this campaign. It provides the necessary data variability to constitute a reliable dataset. The data collected during this acquisition campaign are within the purple area drawn

on the map in Figure 4.1. It is important to note that this acquisition campaign is made under sunny weather. Examples of road scenes collected during this campaign are shown in Figure 4.2.

4.2.2 Multimodal acquisitions

To carry out further experiments, it is necessary to get multimodal, paired, color-based and polarimetric data. The same requirements apply to this acquisition campaign, i.e. getting a large and diverse dataset from a viewpoint similar to driver's one.

To satisfy the need of getting multimodal information of the same scene, a color-based sensor, a Basler camera later replaced by a GoPro to increase the performances, is placed next to the polarimetric camera. The obtained acquisition setup is placed behind the windshield at the height of the driver's eye. This embedded system is shown Figure 4.3.

To increase the diversity of the acquired multimodal road scenes, a larger area nearby Rouen-Normandy is explored. Road scenes of highways, cities, small villages, parking and academic areas are collected. To increase the variability of the scenes, the acquisitions are made under various weather conditions, including sunny, cloudy and foggy. The circuit of this acquisition campaign is represented by the grey (cloudy weather), the green (cloudy weather) and the blue (sunny weather) lines in Figure 4.1. The red star in the same figure indicates the place where the foggy scenes are acquired. Examples of paired multimodal scenes are shown in Figure 4.4

Another acquisition campaign is made in order to get more adverse weather scenes. Because it is difficult to capture outdoor scenes in adverse weather, the acquisitions are made in a simulation platform. This platform is located at the Cerema Center in Clermont-Ferrand[1]. It provides night/day simulation as well as fog from 15 meters visibility and drizzle to dense rain. The advantage of such a platform is that the exact visibility of each scene is known unlike in real scenes. Examples of paired multimodal scenes acquired at the Cerema platform are shown in Figure 4.5.

4.2.3 Datasets constitution

Once the images acquired, they need to be sorted and labelled to constitute relevant datasets to perform the needed experiments. In this section, the choices regarding the data sorting and labelling are detailed as well as the final datasets features.

Data sorting and labelling

Before giving further details on the sorting task, it is important to review the polarimetric and color-based sensors properties. The polarimetric camera has a frame rate of 25 fps while the color-based sensor has a frame rate set to 30 fps during the acquisition campaigns. Regarding the sensors' properties, the polarimetric camera has a standard lens and has a resolution of 500×500 pixels. As for the Basler color-based camera, it has a standard lens and a resolution of 720×480 pixels. Finally, the GoPro has a

[1]More information can be found at `https://www.cerema.fr/en`

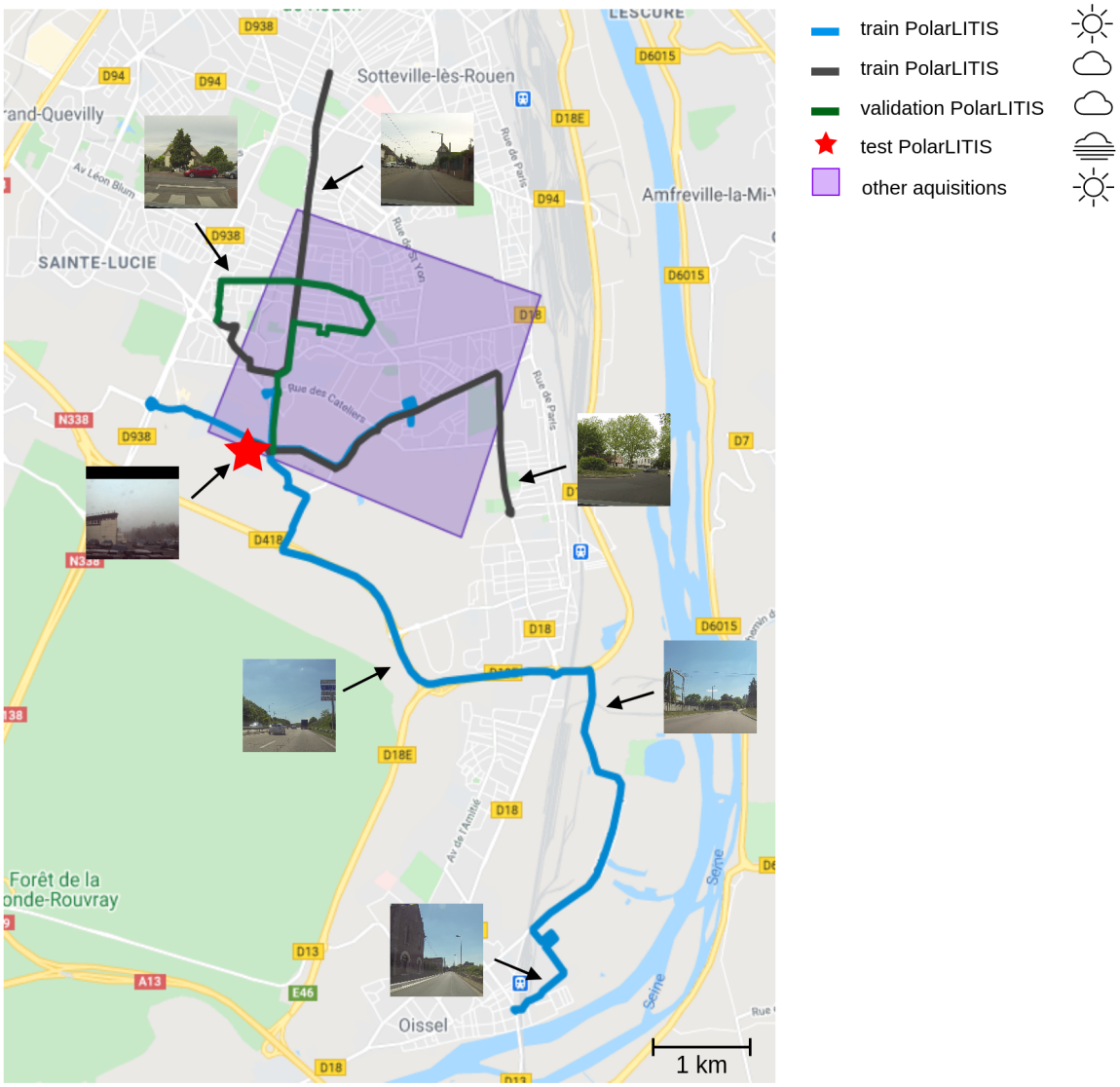

Figure 4.1: Data acquisition circuits. The purple area indicates the acquisitions of polarimetric images only. The blue (train, sunny), grey (train, cloudy) and green (validation, cloudy) as well as the red star (test, foggy) indicate the circuit of the multimodal acquisitions. Note that the training, validation and testing sets of the multimodal dataset cover different areas.

Figure 4.2: Examples of road scenes captured during the polarimetric acquisition campaign. Here (I_0, I_{45}, I_{90}) are placed as the (R, G, B) format.

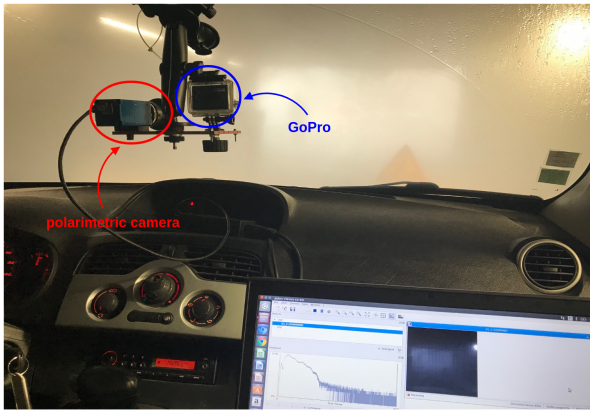

Figure 4.3: Embedded acquisition setup.

Figure 4.4: Examples of road scenes captured during the multimodal acquisition campaign. First row contains the the polarimetric version of the scenes, represented by the intensity I_0, and second row is their RGB equivalent.

Figure 4.5: Examples of road scenes captured during the multimodal acquisition campaign at the Cerema tunnel. First row contains the the polarimetric version of the scenes, represented by the intensity images $I = (I_0, I_{45}, I_{90})$, and second row is their RGB equivalent.

Sensor	Modality	Frame rate	Type of lens	Resolution	Post processing	Final resolution
Polarcam	$(I_0, I_{45}, I_{90}, I_{135})$	25 fps	standard	500×500 pixels	$\emptyset$	500×500 pixels
Basler	(R, G, B)	30 fps	standard	720×480 pixels	Crop width fill top with 0	557×557 pixels
GoPro	(R, G, B)	30 fps	fisheye	3648×2736 pixels	Crop image	906×945 pixels

Table 4.1: Overview of the different sensors properties as well as their post processing to get the closest multimodal pair of images.

fisheye lens and a resolution of 3648×2736 pixels. In order to get the closest content possible between two multimodal paired images, the color-based images are processed. Regarding the images from the Basler camera, their width is cropped to 557 pixels to get the closest content possible to their polarimetric equivalent. Their height is filled with 0 at the top of the image (sky) to get squared-shaped images without altering the original objects' shapes which result in a 557×557 image. As for the GoPro camera, the edges are cropped to get a 906×945 pixels image. Note that this process also reduces the deformation caused by the fisheye lens, accentuated at the edges of the image, since the interesting content of the raw color-based images is mostly located at their center. The properties of the different sensors are summarized up in Table 4.1.

Since the two acquisitions campaigns, polarimetric only and multimodal, have two different purposes, the data sorting is done accordingly in the following ways:

- Polarimetric only: since this acquisition campaign aims to explore the relevance of polarimetric features to describe road scenes, one out of 25 frames are kept to constitute the final dataset. This enables to get a diverse and various enough dataset aiming to characterize road scenes by their polarimetric features;

- Multimodal acquisitions (sunny and cloudy): this acquisition campaign aims to cover the wide range of diversity encountered in road scenes. To achieve this goal, one out of 50 frames and one out of 60 frames are kept respectively regarding the polarimetric and color-based (GoPro) sensors to constitute the final dataset.

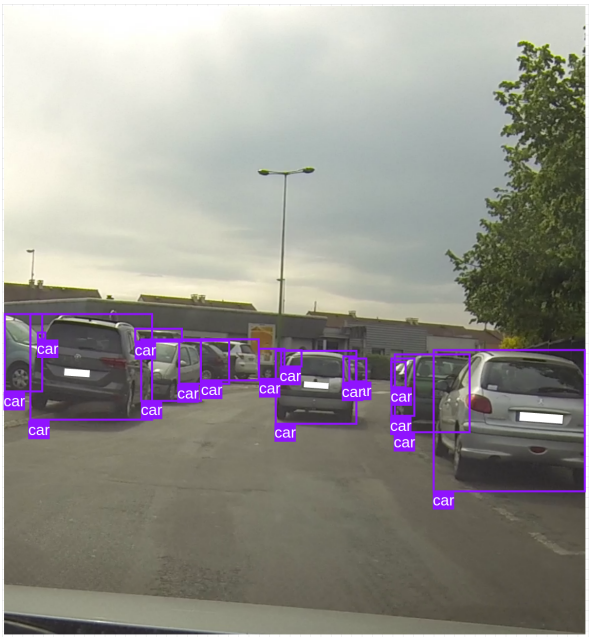

Figure 4.6: Illustration of the labelling precision.

- Multimodal acquisitions (foggy and Cerema tunnel): these acquisition campaigns aim to collect road scenes in adverse weather. Since the acquisitions are made in restricted areas, the images are selected manually to maximize their variability.

Note that, for the polarimetric acquisitions and the multimodal acquisitions (sunny and cloudy), the choices are made according to the different sensors' frame rates. Synchronizing the cameras and a well-chosen camera calibration and sensor placement could have avoided such image selection.

Bounding boxes are used to label images since the dataset aims to perform object detection. Four classes are selected to annotate the most frequently encountered objects in road scenes, which are 'car', 'person', 'bike', and 'motorbike'. The 'car' class contains all four-wheels motor vehicles, including cars, vans, trucks and buses. The 'person' class contains all kinds of road users, including pedestrians, cyclists and bikers, except car drivers. The 'bike' class contains regular and electric bikes without including the cyclists. The class 'motorbike' contains the motorbikes without including the bikers. Every object is labelled in the images, including semi-occluded objects (i.e. people behind parked cars) and mostly occluded objects (i.e. parts of windshields corresponding to cars parked behind many others). Figure 4.6 illustrates the precision of the labels.

Datasets properties

To carry out the different experiments of this thesis, six different datasets are constituted. The first dataset is exclusively composed of polarimetric images in sunny weather. It explores the relevance of using polarimetric features to characterize road scenes. The second dataset studies the behavior of polarimetric features under adverse weather. Its training and validation sets exclusively contain polarimetric images

Dataset	Purpose	Modalities	Labels	Weather Conditions
1	Exploration of polarimetric features[2]	polarimetric	Bounding boxes	sunny
2	Study of polarimetric features under fog[3]	polarimetric color-based	Bounding boxes	sunny foggy
3	Comparison and fusion of multimodal features under fog[4]	polarimetric color-based	Bounding boxes	sunny cloudy foggy
4	Generation of polarimetric images[5]	polarimetric color-based	$\emptyset^{\dagger}$	sunny cloudy
5	Evaluation of generated polarimetric images[6]	polarimetric	Bounding Boxes	sunny cloudy
6	Study of polarimetric features in several weather conditions[7]	polarimetric color-based	Bounding Boxes	fog (several densities) dense rain

Table 4.2: Overview of the different datasets used in this work. †: data generation is an unsupervised training process which does not require labels.

in sunny weather and its testing set is composed of paired multimodal color-based and polarimetric images under fog. The third dataset, named the PolarLITIS dataset, contains paired multimodal polarimetric and color-based images. It compares how color-based and polarimetric features vary with the weather conditions. It is also used for multimodal fusion to improve road scenes description in adverse weather conditions. Its training set is composed of sunny and cloudy scenes, its validation set of cloudy scenes and its testing set of foggy scenes. The training, validation and testing sets of the PolarLITIS dataset cover different areas as shown in Figure 4.1 to minimize the risk of over-fitting. The fourth dataset aims to generate polarimetric images from color-based ones and contains unpaired multimodal images in sunny and cloudy weather. The fifth dataset only contains polarimetric features in sunny and cloudy weather and evaluates the quality of the generated polarimetric images. Finally, the sixth dataset is composed of paired multimodal color-based and polarimetric images. The acquisitions are made in the Cerema tunnel, simulating road scenes under fog and rain. Such a device gives the exact visibility distance of fog and rain. Eleven different weather conditions are contained in this dataset, including foggy scenes with respectively 15m, 20m, 25m, 30m, 35m, 40m, 45m, 50m, 60m and 70m and tropical rain. This dataset conducts a further analysis of the impact of polarimetric features to characterize road scenes when the visibility is altered. The overall information of these six datasets are summarized up in Table 4.2 and their size, as well as the number of instances of each class can be found in Table 4.3.

[2]The dataset can be downloaded at: `https://doi.org/10.5281/zenodo.5547728`
[3]The dataset can be downloaded at: `https://doi.org/10.5281/zenodo.5547750`
[4]The dataset can be downloaded at: `https://doi.org/10.5281/zenodo.5547760`
[5]The dataset can be downloaded at: `https://doi.org/10.5281/zenodo.5547768`
[6]The dataset can be downloaded at: `https://doi.org/10.5281/zenodo.5547796`
[7]The dataset can be downloaded at: `https://doi.org/10.5281/zenodo.5547801`

Dataset	Properties	Train	Validation	Test
1	Weather conditions	sunny	sunny	sunny
	Number of images	$1005 \times 0.8^*$	$1005 \times 0.2^*$	231
	Instances of 'car'	$7743 \times 0.8^*$	$7743 \times 0.2^*$	645
	Instances of 'person'	$503 \times 0.8^*$	$503 \times 0.2^*$	153
	Instances of 'bike'	0	0	4
	Instances of 'motorbike'	$19 \times 0.8^*$	$19 \times 0.2^*$	0
2	Weather conditions	sunny	sunny	foggy
	Number of images	$2221 \times 0.8^*$	$2221 \times 0.2^*$	509
	Instances of 'car'	$11687 \times 0.8^*$	$11687 \times 0.2^*$	9265
	Instances of 'person'	$1488 \times 0.8^*$	$1488 \times 0.2^*$	442
	Instances of 'bike'	$4 \times 0.8^*$	$4 \times 0.2^*$	12
	Instances of 'motorbike'	$21 \times 0.8^*$	$21 \times 0.2^*$	0
3	Weather conditions	sunny/cloudy	cloudy	foggy
	Number of images	1640	420	509
	Instances of 'car'	6061	2102	9265
	Instances of 'person'	527	134	442
	Instances of 'bike'	39	7	7
	Instances of 'motorbike'	14	5	0
4	Weather conditions	sunny/cloudy	$\emptyset^\dagger$	$\emptyset^\dagger$
	Number of images	2485	$\emptyset^\dagger$	$\emptyset^\dagger$
	Instances of 'car'	$\emptyset^\dagger$	$\emptyset^\dagger$	$\emptyset^\dagger$
	Instances of 'person'	$\emptyset^\dagger$	$\emptyset^\dagger$	$\emptyset^\dagger$
	Instances of 'bike'	$\emptyset^\dagger$	$\emptyset^\dagger$	$\emptyset^\dagger$
	Instances of 'motorbike'	$\emptyset^\dagger$	$\emptyset^\dagger$	$\emptyset^\dagger$
5	Weather conditions	sunny/cloudy	sunny/cloudy	sunny/cloudy
	Number of images	3861	1248	509
	Instances of 'car'	19587	3793	2793
	Instances of 'person'	2049	294	161
	Instances of 'bike'	16	35	3
	Instances of 'motorbike'	52	4	5
6	Weather conditions	$\emptyset^\ddagger$	$\emptyset^\ddagger$	foggy/rainy
	Number of images	$\emptyset^\ddagger$	$\emptyset^\ddagger$	484
	Instances of 'car'	$\emptyset^\ddagger$	$\emptyset^\ddagger$	461
	Instances of 'person'	$\emptyset^\ddagger$	$\emptyset^\ddagger$	718
	Instances of 'bike'	$\emptyset^\ddagger$	$\emptyset^\ddagger$	152
	Instances of 'motorbike'	$\emptyset^\ddagger$	$\emptyset^\ddagger$	0

Table 4.3: Datasets properties. $*$: four-fifth of the train/validation set are used for training purposes and the remaining one-fifth for validation purposes. †: data generation is an unsupervised training process using unpaired data which does not require neither validation and testing sets nor labels. ‡: this dataset is used for evaluation purposes only.

4.3 Encoding images for machine learning

In section 4.2, the different datasets constituted for experimental purposes are presented. However, since these datasets do not contain enough images to perform an efficient training from scratch, it is paramount to use networks pre-trained on larger RGB datasets for the different experiments. To this end, three channels images containing different polarimetric features are constituted, leading to six different data formats. As seen in section 1.2, several polarimetric features can be computed from the four acquired intensities I_0, I_{45}, I_{90} and I_{135}. Six data formats, resulting in different polarimetric features combinations, are constituted to provide different information of the same scene. In order to get more homogeneous images, each channel of the polarimetric images is normalized between 0 and 255. This normalization is chosen so that the polarimetric images are coded in the same range as the 8 bits RGB images to be processed by neural networks in the same way. The different data formats, which have a physical meaning, and the intuition behind them are presented in this section.

4.3.1 Intensities images

This data format gathers three intensities $I_{\alpha_i, i=1:3}$ associated to three angles of the linear polarizer $\alpha_{i,i=1:3}$. The choice of three angles instead of four comes from equation (1.8), implying that theoretically the fourth channel can be deduced from the three others. The intensities I_0, I_{45} and I_{90} thus contain all the necessary information for the learning process and I_{135} would be redundant. This data format is referred to as intensities images $I = (I_0, I_{45}, I_{90})$. An example of such coding is illustrated in Figure 4.7.

4.3.2 Stokes images

The linear Stokes vector is a rich polarimetric feature that directly describes the reflected light wave. Knowing that, the three Stokes parameters are chosen to constitute another data format. This data format is referred as the Stokes images $S = (S_0, S_1, S_2)$ and is illustrated in Figure 4.8.

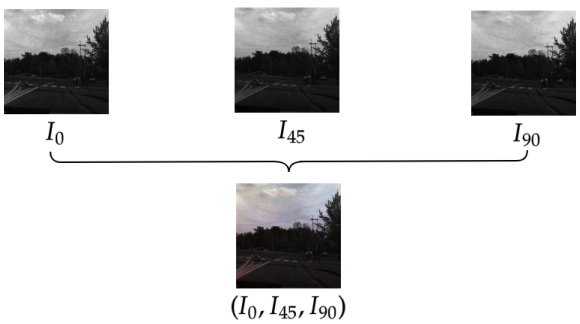

Figure 4.7: Example of an intensity image. I_0, I_{45} and I_{90} are placed respectively as the RGB configuration.

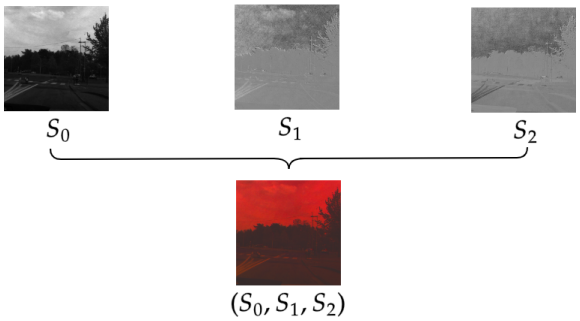

Figure 4.8: Example of a Stokes image. S_0, S_1 and S_1 are placed respectively as the RGB configuration.

4.3.3 Pauli inspired images

This data format is a mix between the polarimetric intensities I and the linear Stokes vector S. It is inspired by the Pauli decomposition [248] of the polarimetric information contained in polarization-encoded SAR images. The Pauli decomposition has shown high performances in image classification [249], [250]. Unlike, the four intensity images I_0, I_{45}, I_{90} and I_{135} obtained in linear polarization, the polarimetric SAR intensities are encoded as I_{HH}, I_{HV}, I_{VH} and I_{VV}, which respectively refer to orientations H (horizontal) and V (vertical) of the received and the transmitted light wave. For instance, the crossed polarization I_{HV} represents the intensity of a horizontal transmitted light by a source and a vertical received light wave by the sensor. The original Pauli decomposition is carried out for $I_{HH} - I_{VV}$, I_{HV} and $I_{HH} + I_{VV}$ components placed respectively as the RGB configuration. According to the rotation of the polarizer, similarities between I_{HH} and I_0 are noticed, even if the incident light is not known, meaning that the polarizer oriented at an angle of 0° is assimilated to an horizontal polarization filter. In the same way, I_{VV} can be assimilated to I_{90} for the vertical filter. I_{VH} is assimilated to I_{45} as it corresponds to the mean orientation between the horizontal and the vertical filters. From equation (1.9), the Stokes parameters S_0 and S_1 are calculated as $S_0 = I_0 + I_{90}$ and $S_1 = I_0 - I_{90}$. Following the Pauli inspired format, the data is encoded as Pauli $= (I_0 - I_{90}, I_{45}, I_0 + I_{90})$ which means Pauli $= (S_1, I_{45}, S_0)$ respectively coded as the RGB configuration. An illustration of this data format can be found in Figure 4.9.

4.3.4 HSV images

From the polarimetric features, it is possible to obtain the HSV format of the scene [251]. This equivalence between the HSV encoding and the polarization parameters is based on the intuition that the angle of polarization ϕ corresponds to the Hue channel, the degree of polarization ρ to the Saturation and the total intensity S_0 to the Value of each pixel. The HSV data format is encoded as HSV $= (\phi, \rho, S_0)$. An illustration of this data format can be found in Figure 4.10. It is important to note that, to fulfill the HSV format, the channel containing ϕ is normalized between 0 and 180.

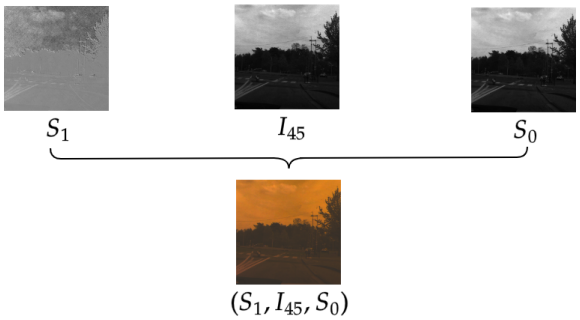

Figure 4.9: Example of a Pauli inspired image. S_1, I_{45} and S_0 are placed respectively as the RGB configuration.

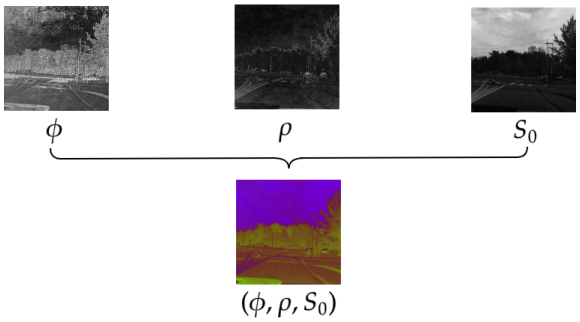

Figure 4.10: Example of a polarimetric HSV image. ϕ, ρ and S_0 are placed respectively as the RGB configuration.

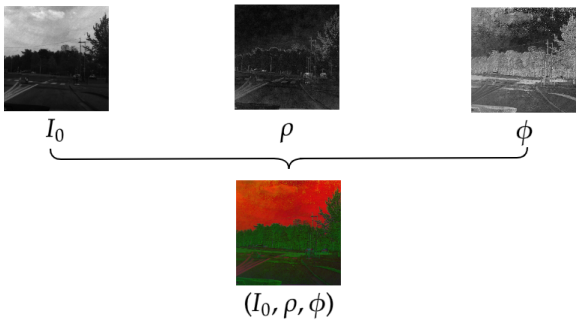

Figure 4.11: Example of a polarimetric pseudo-HSV image. I_0, ρ and ϕ are placed respectively as the RGB configuration.

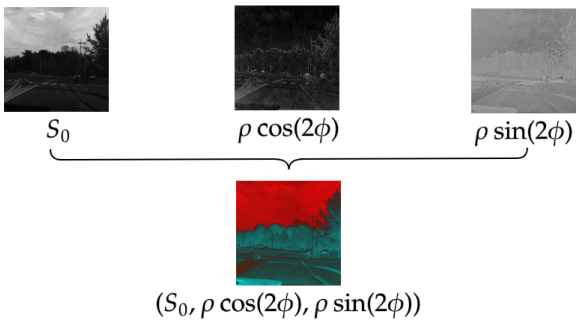

Figure 4.12: Example of a Poincaré inspired image from its polarimetric features. S_0, $\rho\cos(2\phi)$ and $\rho\sin(2\phi)$ are placed respectively as the RGB configuration.

4.3.5 pseudo-HSV images

The pseudo-HSV images, which are the first version of the HSV images, are used in early experiments. In this data format, I_0, ρ and ϕ are respectively placed as the RGB configuration, leading to pseudo-HSV = (I_0, ρ, ϕ). An illustration of this data format can be found in Figure 4.11. It is important to note that, because $\phi \in [\frac{-\pi}{2}, \frac{\pi}{2}]$ and with regards to the HSV format, the channel containing ϕ is normalized between 0 and 180.

4.3.6 Poincaré inspired images

This data format is inspired by the representation of the Stokes vector, normalized by its first component S_0, in the Poincaré sphere [252]. In the case of linear polarization, the Stokes vector is of dimension three instead of dimension four in a general Stokes formalism. The Stokes vector normalized by its first element is no longer represented in a Poincaré sphere, but in a unitary circle. This representation is sketched in Figure 1.5 and illustrates equations (1.10) and (1.11). The projection of ρ on the abscissa x and the ordinate y axis of the unit circle results in two components, $\rho\cos(2\phi)$ and $\rho\sin(2\phi)$. These components are used to constitute this new data format, in which S_0, $\rho\cos(2\phi)$ and $\rho\sin(2\phi)$ are placed as the RGB configuration resulting in Poincaré = (S_0, $\rho\cos(2\phi)$, $\rho\sin(2\phi)$). An illustration of this data format can be found in Figure 4.12.

To sum up, six polarimetric data formats, encoded for machine learning, are consti-

Data format	Channel 1	Channel 2	Channel 3
I	I_0	I_{45}	I_{90}
S	S_0	S_1	S_1
Pauli	S_1	I_{45}	S_0
HSV	ϕ	ρ	S_0
pseudo-HSV	I_0	ρ	ϕ
P	S_0	$\rho\cos(2\phi)$	$\rho\sin(2\phi)$

Table 4.4: Summary of the different polarimetric data formats. Here I, S, Pauli, HSV, pseudo-HSV and P stand respectively for intensities images, Stokes images, Pauli inspired images, HSV images, pseudo-HSV images and Poincaré inspired images.

tuted. Table 4.4 recaps the content of each channel of the polarimetric data formats.

4.4 Data generation

As mentioned previously, constituting a dataset is an expensive task. Both the data acquisition, data sorting and data annotation are time-consuming tasks that limit the sizes of the constituted datasets. This section explains how performing image-to-image translation enables to dispose of large polarimetric datasets, labelled for road object detection.

4.4.1 Motivations

The datasets presented in section 4.2, are the first multimodal polarimetric and color-based public datasets for road object detection. However, the sizes of the constituted datasets are limited by the expensive labelling task. Disposing of larger datasets would enable to explore further the behavior of polarimetric features in road scenes. Conventional data augmentation methods often palliate the lack of color-based data during the training process of DNN. This method is excluded when learning polarimetric features since the augmented images do not satisfy the physical admissibility constraints presented in equations (1.2) and (1.4). This is the reason why the best solution is to formulate the problem of polarimetric image generation as a CycleGAN learning problem (see section 2.4) under physical constraints. The CycleGAN algorithm achieves unpaired image-to-image translation with only a few images and the added constrains ensure that the generated images are valid. This method allows to circumvent the expensive labelling step by transferring a source labelled dataset to one or multiple target domains [253] by keeping unchanged the shapes of the source image. In this section, the adapted CycleGAN algorithm is explained and the generation results are presented.

4.4.2 Proposed approach

As discussed above, the goal is to learn a generative model able to produce realistic polarization-based images from RGB images. For the sake, an image-to-image translation framework is adopted and extended to account for admissibility the constraints a polarimetric image must fulfill.

To generate a polarimetric image from an RGB image, the CycleGAN approach is proposed to learn the translation models M_{XY} and M_{YX} between X the domain of the polarimetric images and Y the RGB images domain (see section 2.4). Let $\hat{I} \in \mathbb{R}^4$ be the intensity vector associated to a pixel of a generated polarimetric image. To be physically admissible, each pixel has to satisfy the admissibility constraints (see equation (1.2)) and the calibration constraint (see equation (1.4)). These polarimetric constraints are referred by $\mathcal{C}_1$, $\mathcal{C}_2$ and $\mathcal{C}_3$ as follows:

$$\begin{aligned} \mathcal{C}_1 &: I = AS \ , \\ \mathcal{C}_2 &: {S_0}^2 \geqslant {S_1}^2 + {S_2}^2 \ , \\ \mathcal{C}_3 &: S_0 > 0 \ . \end{aligned}$$

By construction, S_0 is always positive as it represents the total intensity reflected from an object. As the last layer of the generation models customary uses the hyperbolic tangent as activation function, each output intensity $\hat{I}$ is within the range $]-1, 1[$ which is scaled to $]0, 255[$. Hence $\hat{S}_0 = \hat{I}_0 + \hat{I}_{90}$ (see equation (1.9)) is ensured to be strictly positive. Therefore, constraint $\mathcal{C}_3$ can be deemed satisfied for the generated polarimetric images. To handle the remaining constraints $\mathcal{C}_1$ and $\mathcal{C}_2$, one could resort to the Lagrangian dual of CycleGAN optimization problem (see equation (2.12)) subject to these constraints. However, this may be computationally expensive, as it requires to entirely optimize four neural networks (respectively the discrimination and the mapping network models) in an inner loop of a dual ascent algorithm. Moreover the overall optimization procedure may not be stable because of the minmax game involved in the CycleGAN learning.

In order to derive an efficient algorithm to learn CycleGAN under output constraints, a relaxation of the problem is introduced. Instead of strictly enforcing the constraints, a measure of how far the generated image pixels are from the admissibility domain is made, through additional cost functions to minimize. For the constraint $\mathcal{C}_1$, a ℓ_2 distance between the generated image $M_{YX}(y)$ and $A\hat{S}$ is proposed. It reads:

$$\mathcal{L}_{\mathcal{C}_1} = \mathop{\mathbb{E}}_{y \sim p_Y} ||M_{YX}(y) - A\hat{S}||_2 \ ,$$

with $\hat{S} = \begin{bmatrix} \hat{S}_0 & \hat{S}_1 & \hat{S}_2 \end{bmatrix}^\top$ the Stokes vector calculated from the generated image by M_{YX} using equation (1.9). Similarly, to enforce the constraint $\mathcal{C}_2$, a rectified linear penalty $\mathcal{L}_{\mathcal{C}_2}$ is considered. It is defined by:

$$\mathcal{L}_{\mathcal{C}_2} = \mathop{\mathbb{E}}_{y \sim p_Y} \max\left({\hat{S}_1}^2 + {\hat{S}_2}^2 - {\hat{S}_0}^2, 0\right) \ .$$

The loss $\mathcal{L}_{\mathcal{C}_1}$ translates the respect of the acquisition conditions according to the calibration matrix A while $\mathcal{L}_{\mathcal{C}_2}$ is related to the physical admissibility constraint on the deduced Stokes vectors from the generated image.

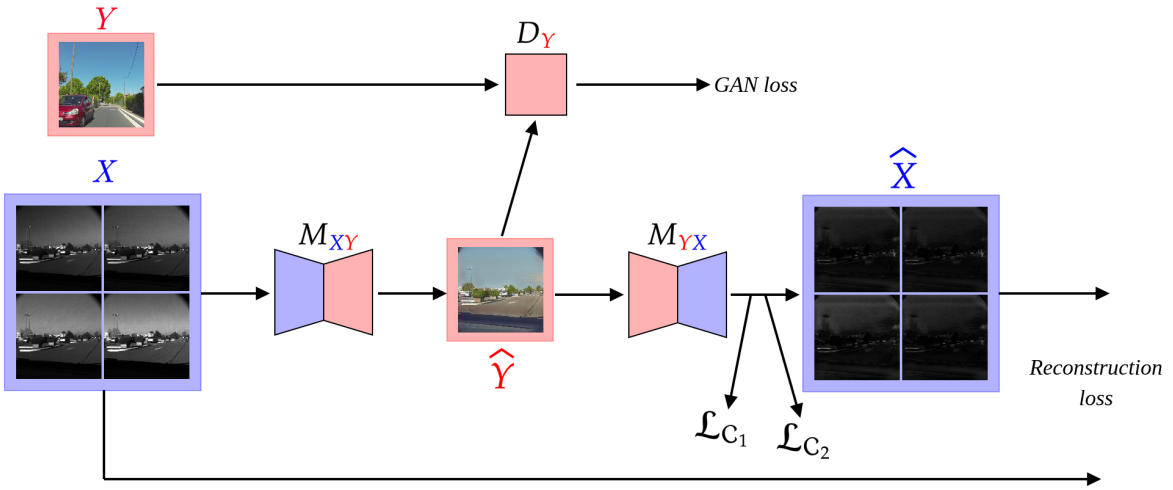

Figure 4.13: Overview of the CycleGAN training process extended with $\mathcal{L}_{\mathcal{C}_1}$ and $\mathcal{L}_{\mathcal{C}_2}$.

Gathering all these elements, the CycleGAN under physical constraints is trained by optimizing the following objective function:

$$\mathcal{L}_{\text{final}} = \mathcal{L}_{\text{CycleGAN}} + \mu\mathcal{L}_{\mathcal{C}_1} + \nu\mathcal{L}_{\mathcal{C}_2} \quad . \tag{4.1}$$

The non-negative hyper-parameters μ and $\nu \in \mathbb{R}^+$ control respectively the balance of admissibility and calibration constraints according to the CycleGAN loss $\mathcal{L}_{\text{CycleGAN}}$ (see equation (2.11)). As the values of $\mathcal{L}_{\mathcal{C}_1}$ and $\mathcal{L}_{\mathcal{C}_2}$ are computed pixel-wisely, their averages over the whole image in the objective function are considered. The training principle of the proposed generative model is illustrated in Figure 4.13.

4.4.3 Experimental evaluation

Hereafter, the experimental setup, including the image generation procedure and its evaluation, is presented.

Polarimetric images generation using CycleGAN

To conduct the experiments, 2485 unpaired images from each domain (RGB and polarimetry, see row 5 of Table 4.3) are selected. Example instances are shown in Figure 4.14 and Figure 4.15 for polarimetric and RGB images respectively. The polarimetric images are of dimension $500 \times 500 \times 4$. The latter dimension is due to the four intensities acquired by the camera, namely I_0, I_{45}, I_{90} and I_{135}. The RGB images are of dimension $906 \times 945 \times 3$.

The extended CycleGAN is trained for 400 epochs on randomly cropped patches of size 200×200. As for the constraints, it is found experimentally that setting the hyper-parameters $\mu = 1$ and $\nu = 1$ in (4.1) provides the best performances. As for the original CycleGAN, the hyper-parameter η, controlling the reconstruction cost, is set to $\eta = 10$. The learning rate is decreased linearly from 2×10^{-4} to 2×10^{-6} during the 400 training epochs.

Figure 4.14: Examples of polarimetric images used to train the adapted CycleGAN. Only the intensities I_0 are shown here.

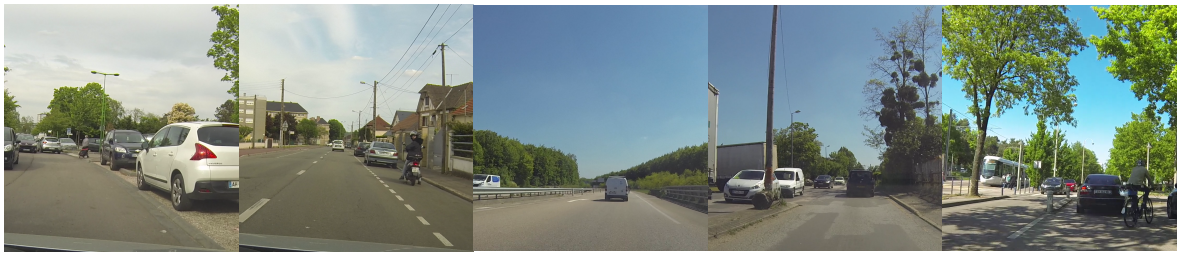

Figure 4.15: Examples of RGB images used to train the adapted CycleGAN.

To evaluate the effectiveness of the trained generative model, KITTI and BDD100K (only using daytime images since polarimetry fails to characterize objects during nighttime) are considered, which often serve as testbed in applications related to road scene object detection. The constrained-output CycleGAN trained is used to transfer RGB images from KITTI and BDD100K to the polarimetric domain. The resulting datasets are denoted respectively as Polar-KITTI and Polar-BDD100K. Since the CycleGAN architecture is fully convolutional, it has no requirement on the size of the input image. Therefore, even if the model was trained on 200×200 patches, it scales straightforwardly to the images of size 1250×375 from KITTI and of size 1280×720 from BDD100K datasets.

To assess wether or not fulfilling the physical constraints is paramount, a variant of Polar-KITTI and Polar-BDD100K are investigated: a standard unconstrained CycleGAN based on the same unpaired RGB/polarimetric images is learnt. It is worth mentioning that the so generated polarization-encoded images may not mandatory satisfy the feasibility constraints.

Evaluation of the generated images

To assert the ability of the generated Polar-KITTI and Polar-BDD100K datasets to preserve the relevant features for road scene applications, a detection network is trained following the setup in Figure 4.16. For this experiment, a RetinaNet-50 [3] (see section 5.2 for more information) pre-trained on the MS COCO dataset [4] is fine-tuned in three different settings. In the first setup, the detection model is fine-tuned based on the original RGB KITTI (or BDD100K) while the second experimental setting considers the fine-tuning on the generated polarimetric images from KITTI (Polar-KITTI) or BDD100K (Polar-BDD100K) datasets. The third experimental setting uses the unconstrained variant of the generated images from KITTI or BDD100K datasets. Af-

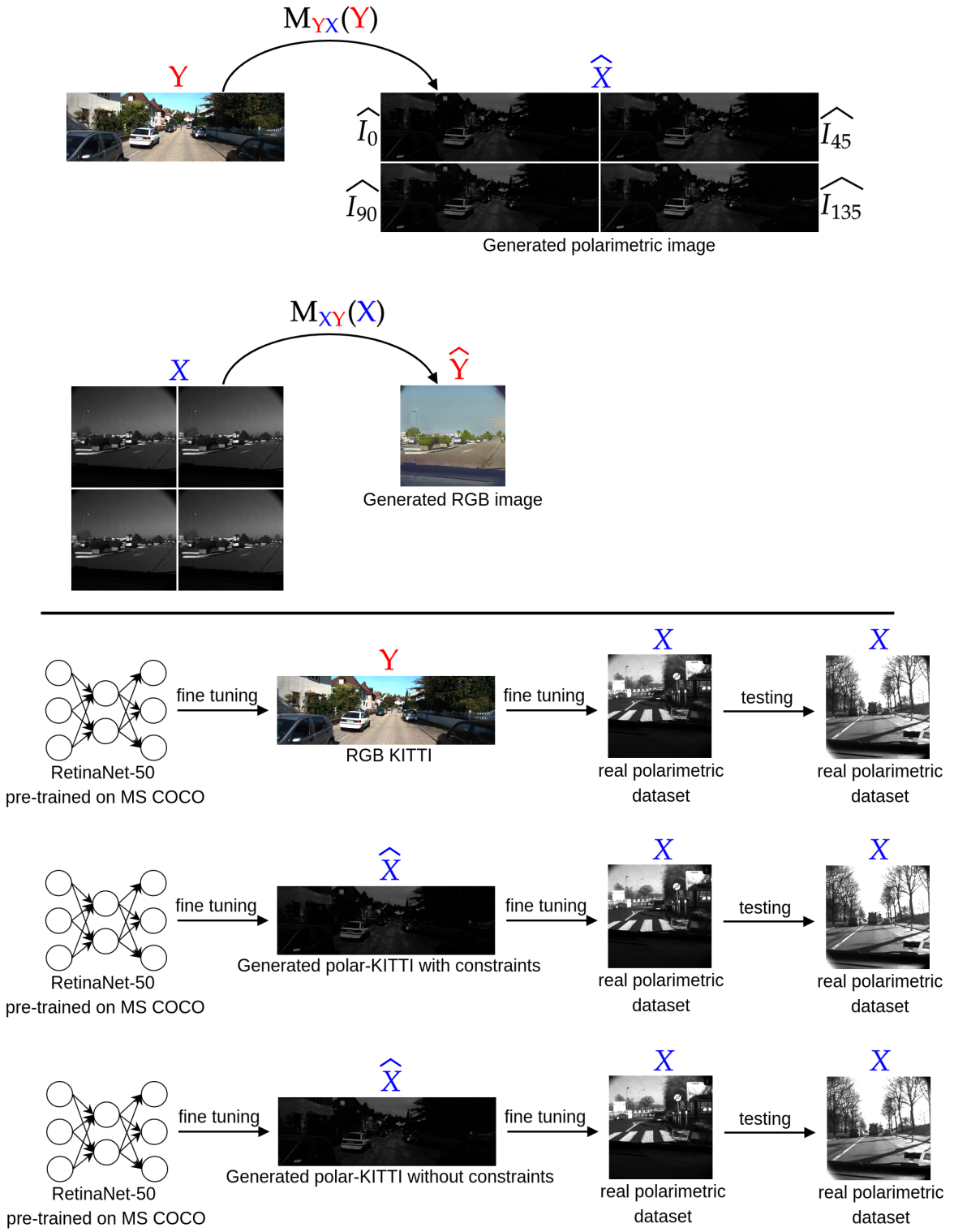

Figure 4.16: Setup of the detection evaluation experiment. The procedure is illustrated with the KITTI dataset and straightforwardly extends to the BDD100K dataset.

terwards the three final detection models are obtained by a last fine-tuning on the real polarimetric dataset (see row 5 of Table 4.3).

Overall, the trained CycleGAN and detection networks under these settings are evaluated in qualitative and quantitative ways. The end goal is to check: (i) the ability of the generated images to help learning polarimetry-based features for object detection, and (ii) the influence of respecting the polarimetric feasibility constraints on detection performances.

The visual quality of the generated images is measured by computing the classical Fréchet Inception Distance (FID) [254]. Computing this distance requires to extract visual features from each set of images (real and generated) using a pre-trained deep neural network (usually an Inception v3 [255] network pre-trained on ImageNet [63]) and to evaluate the Fréchet (or Wasserstein) distance between the distributions of these features, which are assumed to be Gaussian distributions. This distance is calculated using 509 images from each generated polarimetric dataset and from the test set as described in Table 4.3.

As feature extractor, because the classical Inception v3 network is not adapted to polarimetric images since it is trained on ImageNet [63], the convolutional part of a polarimetry-adapted RetinaNet detection network [256] is used, which has been trained on the MS COCO dataset and fine-tuned on a real polarimetric dataset.

In order to evaluate the improvements in the detection, the error rate evolution ER_o is computed. The improvement ER_o on the detection of the object o is given by:

$$ER_o = \frac{1 - AP_o^p - (1 - AP_o^{RGB})}{1 - AP_o^{RGB}} \ ,$$

where AP_o^{RGB} and AP_o^p respectively denote the average precision for object o detection in RGB and in polarimetric images. Note that a negative ER_0 means that AP_o^p is improved over AP_o^{RGB}.

4.4.4 Results and discussion

First an evaluation of wether the generated images are qualitatively coherent is made. For the sake, polarimetric images are generated from their RGB equivalent. An example of the generated polarimetric images is displayed in Figure 4.17.

As for the constraints, Table 4.5 shows how including them to the CycleGAN's loss helps generating images which better fulfill the physical polarimetric properties at the pixel scale. The errors related to the constraints $\mathcal{C}_1$ and $\mathcal{C}_2$ on generated images using this approach are consistent with the observed errors on the real images, whereas the unconstrained approach yields poor results. Obviously, constraint $\mathcal{C}_3$ is met for all generated images thanks to the tanh activation at the last layer of the generative models. Additionally, the obtained FID are of **6022.7** for the unconstrained CycleGAN and **4485.1** for our approach[8], which indicates that taking the constraints into account improves visual and physical quality of the generated samples.

[8]Note that the scale of the FID scores computed with the pre-trained RetinaNet is larger than when using a pre-trained Inception v3 network.

Figure 4.17: Examples of polarimetric image generation. From left to right: I_0, I_{45}, I_{90} and I_{135} ground truth, RGB image and I_0, I_{45}, I_{90} and I_{135} generated from RGB image.

Datasets	$\mathcal{C}$	Mean	Median
Real polar	$\mathcal{C}_1$	0.06 ± 0.04	0.04
	$\mathcal{C}_2$	$2.47 \pm 7.11\%$	0.48%
	$\mathcal{C}_3$	0%	0%
Generated polar no $\mathcal{C}$	$\mathcal{C}_1$	0.26 ± 0.19	0.23
	$\mathcal{C}_2$	$27.31 \pm 43.5\%$	2.15%
	$\mathcal{C}_3$	0%	0%
Generated polar with $\mathcal{C}$	$\mathcal{C}_1$	0.12 ± 0.04	0.12
	$\mathcal{C}_2$	$1.55 \pm 3.36\%$	0.14%
	$\mathcal{C}_3$	0%	0%

Table 4.5: Evaluation of the constraint fulfillment using the designed losses $\mathcal{L}_{\mathcal{C}_1}$ and $\mathcal{L}_{\mathcal{C}_2}$ at the pixel scale. Here, the column $\mathcal{C}$ indicates the evaluated constraint. $\mathcal{C}_1$ refers to the constraints $I = AS$, $\mathcal{C}_2$ to ${S_0}^2 \geqslant {S_1}^2 + {S_2}^2$ and $\mathcal{C}_3$ to $S_0 > 0$. The mean and the median of the percentage of pixels in an image that do not fulfill the constraints $\mathcal{C}_2$ and $\mathcal{C}_3$ are computed. Regarding the constraint $\mathcal{C}_1$, the mean and the median of $||I - AS||/(||I|| + ||AS||)$ is computed.

Databases used	Class	Test	ER_o	Databases used	Class	Test	ER_o
KITTI RGB	person	0.663	N/A	BDD100K RGB	person	0.736	N/A
+ real polar	car	0.785	N/A	+ real polar	car	**0.821**	N/A
mAP		0.724	N/A	mAP		0.778	N/A
Polar-KITTI no $\mathcal{C}$	person	0.673	-0.03	Polar-BDD100K no $\mathcal{C}$	person	0.720	0.06
+ real polar	car	0.786	-0.01	+ real polar	car	0.816	0.03
mAP		0.730	-0.02	mAP		0.768	0.05
Polar-KITTI with $\mathcal{C}$	person	**0.704**	-0.12	Polar-BDD100K with $\mathcal{C}$	person	**0.762**	-0.10
+ real polar	car	**0.794**	-0.04	+ real polar	car	0.815	0.03
mAP		**0.749**	-0.09	mAP		**0.789**	-0.05

Table 4.6: Comparison of the detection performance after the two successive fine-tunings. RetinaNet-50 pre-trained on MS COCO is the baseline of all the experiments. The first row refers to the RetinaNet-50 fine-tuned on KITTI or BDD100K RGB. The second row refers to the fine-tuning on Polar-KITTI or Polar-BDD100K without constraints while the bottom row represents the detection models fine-tuned on Polar-KITTI or Polar-BDD100K with the constraints. All these models are finally fine-tuned on the real polarimetric dataset.

Next, the benefit of the generated images is shown for an application example which is the object detection task. This enables to check if objects contained in the scene are globally physically coherent. A RetinaNet-based detection model is learnt according to the setups described in Section 4.4.3 and the obtained detection performances in term of mAP are summarized in Table 4.6. The bike and motorbike detection performances are not evaluated as the polarimetric dataset does not contain enough objects of these two classes.

As can be seen in Table 4.6, using the generated polarimetric images improves the detection performance in real polarimetric images. The improvement is substantial for car and pedestrian detection. An improvement of 4% for car detection is achieved and of 12% for pedestrian detection which leads to a global improvement of 9% in the detection, using Polar-KITTI with constraints. Similarly for Polar-BDD100K dataset, an improvement of 10% for pedestrian detection is noticed which leads to an increased mAP of 5% (pedestrians and cars). However, regarding BDD100K similar detection performances are obtained either for RGB or polarimetric images and this is due to the fact that generated images using CycleGAN do not perform well on small objects. To verify that, an analysis of the detection scores, while varying the minimal area of the bounding boxes from which objects are taken into account, is made. The obtained results are shown for the training including the Polar-BDD100K and the RGB BDD100K in Figure 4.18. They illustrate that, when the minimal area of bounding boxes increases, the AP of car regarding the training including Polar-BDD100K, overcomes the one including RGB BDD100K. Even if the results are limited by a lower quality of small objects in the generated images in this specific case, the generated polarimetric images help improving the overall detection results.

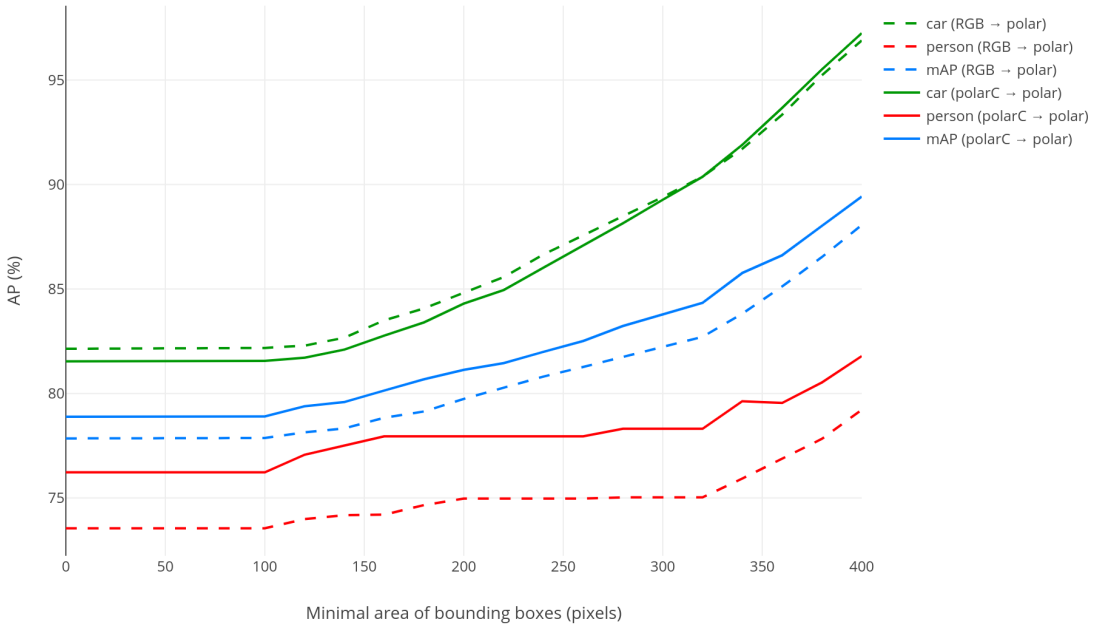

Figure 4.18: Evolution of the average precision when setting a minimal area of the bounding boxes to be detected. PolarC refer to the generated polarimetric images under the admissibility physical constraints. Here green lines refer to the evolution of cars' detection, blue lines to the evolution of the mAP and red lines to the evolution of person's detection. The dashed lines refer to the training including the BDD100K RGB and the solid lines to the training including Polar-BDD100K.

4.5 Summary

This chapter presents the polarimetric and multimodal datasets constituted to carried out all the required experiments within this thesis. The polarimetric and multimodal (polarimetric and color-based) acquisitions setups, as well as the different acquisition circuits, are sketched. The properties of the different sensors are given, which are used to sort the acquired frames, in order to maximize the diversity of the different datasets. The post-processing operations, enhancing the collected images are detailed followed by the annotation technique. The intuition behind every experiment carried out in this thesis is also explained before giving the properties of the constituted datasets. In order to enable deep architectures to process polarimetric images, we come up with six polarimetric data formats, encoded for machine learning. These polarimetric data formats include different polarimetric features combinations, giving different information on the scene. Even though the constituted datasets are the first publicly available datasets for road object detection, they are limited by their sizes as we could not afford dedicating more time to their constitution. To overcome this limitation, an image-to-image translation pipeline is designed, enabling to generate polarimetric images from RGB ones. The CycleGAN architecture, the best solution to this problem, is modified so that the generated images meet the physical admissibility constraints of the polarimetric images. The designed pipeline enables to generate polarimetric versions of flagship color-based datasets for road object detection, such as KITTI or BDD100K and can be extended to similar datasets. The next chapter focuses on the experiments carried out using these datasets, demonstrating the impact of polarimetric features to enhance road object detection in adverse weather.

Chapter 5

Polarimetric imaging for adverse weather conditions

Contents

5.1 Introduction

The previous chapter presents the different acquisition campaigns aiming to collect polarimetric images of road scenes in several weather conditions. Prior this work, there were no large enough and publicly available datasets, containing polarimetric images for road object detection. Data collection is therefore paramount to study the impact of polarimetric features combined with deep architectures on road scene analysis. Using the constituted datasets, this chapter focuses on the experiments demonstrating the added value of polarimetric features for road object detection in foggy scenes.

A first experiment is conducted on polarimetric sunny road scenes. It aims to find a deep architecture suited for road object detection and verifies that it is adapted to process polarimetric images. It also demonstrates that the constituted dataset performs

an efficient training without over-fitting, with detection scores in the same range as the state of the art.

The second experiment gives a first intuition on the utility of polarimetric features to enhance road scene analysis under fog. To this end, two deep architectures are trained separately on unpaired polarimetric and RGB datasets, containing road scenes in sunny weather. The obtained architectures are tested respectively on road scenes under fog. The experimental results show that polarimetric features seem more invariant to visibility changes than color-based ones.

The third experiment demonstrates and confirm this first intuition. Using the paired multimodal PolarLITIS dataset, the training processes are conducted on the same basis. On top of that, this experiment explores further polarimetric data formats. The evaluation task demonstrates reliably that, unlike color-based features, the polarimetric characteristics learnt in good weather condition can be used to detect road objects under fog. This property can be a real asset to enhance autonomous driving, since being able to describe a scene despite unexpected visibility changes would improve road users' safety.

5.2 Learning polarimetric features using Deep Neural Networks

Before evaluating the impact of polarimetric imaging for road scene understanding, it is necessary to find an adapted architecture to process it. As a reminder, this work is the first attempt to combine polarimetric imaging with deep architectures for road object detection. The ability of such architectures to perform efficiently object detection in polarimetric road scenes is thus unknown. This experiment aims to evaluate if color-based and polarimetric features achieve similar performances in detecting road objects. To this end, a DNN is trained on polarimetric road scenes and its performances are compared to the state of the art.

In this section, the properties of the required architecture to perform this experiment are first given. Once the adapted DNN presented, the experimental setup in which it is included is sketched. The experimental results are then exposed and discussed.

5.2.1 Experiments

In this section, the experiments carried out are detailed. The required features, motivating the choice of an adapted deep architecture for road object detection in polarimetric images, are first exposed. Based on these information, the experimental setup performing this task is then described.

Choosing an adapted architecture

The first step of this experiment consists in selecting the best architecture to perform road object detection on polarimetric images. To justify the technical choices, it is important to remind the properties of the polarimetric dataset. It is composed of 1236 images with 4 unbalanced classes (see row 1 of Table 4.3 for more information).

As a matter of fact, because the acquisitions are made on road scenes in real traffic conditions, cars are encountered more often than other road users. To avoid neglecting the underrepresented classes during the training process, the architecture must use the FL (see section 2.3 for more information). The FL is known to enable an efficient training from an unbalanced dataset.

The class unbalancement is not the only particularity of the used dataset. Since it is composed of 1236 images, it is paramount to use an architecture able to perform an efficient training on small datasets without relying on data augmentation. Indeed, the polarimetric images responding to physical constraints (see section 1.2), the images must remain unchanged during the training process. As a matter of fact, applying a rotation, a translation or a distortion to a polarimetric image do not guarantee the respect for the physical constraints at the pixel scale for the resulting image.

The last parameter to take into account is the nature of the task. As a matter of fact, object detection must be performed in real time by autonomous vehicles. This is the reason why the network must be both accurate and able to process several frames per second.

Considering all these technical constraints, RetinaNet [3] is best suited for this experiment. It is originally designed to use the FL and it does not rely on data augmentation to perform an efficient training unlike the SSD architecture [2]. On top of that, it processes up to 14 fps while reaching an accurate detection. Finally, this architecture is able to detect small objects better than the YOLO [1] architecture. Small objects mostly refer to distant road users and detecting them the soonest possible would enable to anticipate the decisions, directly impacting road users' safety.

Since RetinaNet is originally designed to use ResNet-50 [166] as a backbone, most of its performances in the state of the art are made using this architecture. Our experiment aims to compare the performances of polarimetric features to the state of the art. It is thus paramount to keep this pipeline to perform a fair comparison. From now on, this architecture is referred to as RetinaNet-50.

Experimental setup

At this stage, the RetinaNet-50 network is selected for the experiments, as well as the polarimetric dataset detailed in the first row of Table 4.3. As can be seen in this table, the training/validation set is composed of 1005 images of road scenes under good weather conditions. As a matter of fact, a deep architecture can not be trained on such a small dataset from scratch without over-fitting. This is the reason why the RetinaNet-50 is first trained on the MS COCO dataset [4], which is rather large since it contains more than 200k labelled images, to perform a reliable and efficient training. On top of that, MS COCO contains various color-based road scenes under good weather conditions and similar road objects as the polarimetric dataset and a RetinaNet-50 pre-trained on this dataset is publicly available.

Once the RetinaNet-50 is pre-trained on MS COCO, it is fine-tuned on the polarimetric dataset. The relevance of polarimetric features to describe a road scene is evaluated using the intensities images $I = (I_0, I_{45}, I_{90})$ (see section 4.3 for more information). Four fifths of the training/validation set are used for the training purpose and the remaining fifth is used for the validation purpose. The obtained network is

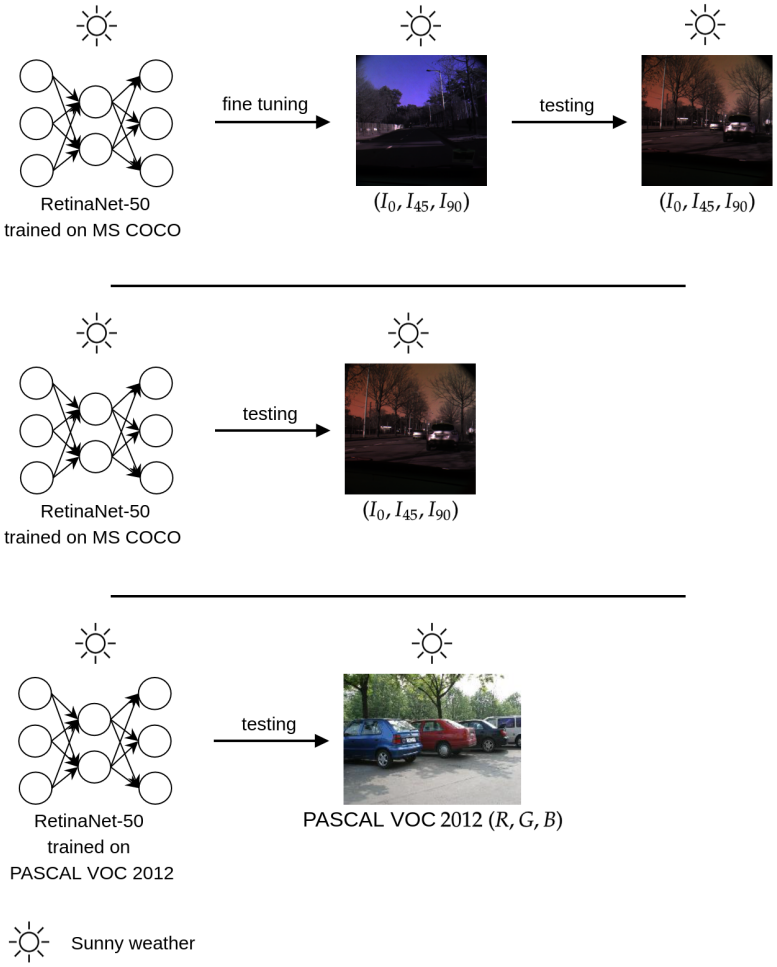

Figure 5.1: Experimental setup. The first row refers to RetinaNet-50 pre-trained on MS COCO, fine-tuned on $I = (I_0, I_{45}, I_{90})$ and tested on I. The second row is the RetinaNet-50 trained on MS COCO and tested on I. The third row is the RetinaNet-50 trained and tested on PASCAL VOC 2012, providing state of the art performances on this dataset.

then evaluated on the testing set. The different road objects' detection scores are respectively compared to the state of the art performances on PASCAL VOC 2012 [235] achieved with the same deep architecture. Unlike other datasets for which only the overall detection performance is available publicly, PASCAL VOC 2012 is the only one providing the score of each detection task with the RetinaNet-50 architecture. Similarly to MS COCO, PASCAL VOC 2012 contains road scenes in good weather with the desired road objects. To visualize the relevance of fine-tuning the network on polarimetric data, the RetinaNet-50 pre-trained on MS COCO is also evaluated on the polarimetric testing set. An illustration of this experimental setup can be found in Figure 5.1.

As for the the technical details, the network is trained during 50 epochs to ensure its convergence and without conventional data augmentation to avoid altering the physical features of polarimetric images. The weights used for evaluation are the ones associated

(a) Detections before fine-tuning on I (b) Detections after fine-tuning on I

Figure 5.2: Detection results using RetinaNet-50

to the lowest validation loss. The original parameters of the RetinaNet network are kept, i.e. an Adaptive moment estimation (Adam) optimizer [257] with a 10^{-5} learning rate.

5.2.2 Results and discussions

Before discussing the obtained results, it is important to note that, because there are less than 20 instances of classes bike and motorbike, they are not taken into consideration for the evaluation task. The formula used to compute the mAP can be found in equation (2.10). In this experiment, $n = 2$ and $AP_i \in \{AP_{\text{person}}, AP_{\text{car}}\}$.

The obtained results are summarized up in Table 5.1. As can be seen in this table, the detections results on polarimetric images are improved after fine-tuning the network on polarimetric features. This shows that the training process is done correctly and that a deep architecture is able to learn polarimetric features. The added value of fine-tuning the network on polarimetric features is illustrated in Figure 5.2.

As for the comparison with the state of the art, we can notice that the car detection score achieved on PASCAL VOC 2012 is of the same order as the one obtained on polarimetric images. As for the pedestrian detection, the score achieved on polarimetric images is not of the same order as the one achieved in the state of the art at this point. This could be due to the fact that the testing set contains a majority of partially and mostly occluded objects of the person class, unlike in the PASCAL VOC dataset. Some further experiments need to be carried out on a larger dataset, containing more instances of the person class, with a good balance between partially occluded and non-occluded objects. Such an experiment would enable to draw a more reliable conclusion regarding polarimetric pedestrian detection.

Class name	State of the art	AP no FT	AP FT
car	89.0	84.5	90.0
person	91.1	33.3	34.8
mAP	90.1	58.9	62.4

Table 5.1: Detections results using RetinaNet-50. The first column is the state of the art detection results on PASCAL VOC 2012. The second column is the detection results on I without fine-tuning the network on the polarimetric dataset. The third column is the detection results on I after fine-tuning the network on polarimetric features.

5.3 Polarimetric imaging under fog

In section 5.2, the ability of deep architectures to learn polarimetric features is studied. The faculty of polarimetric features to characterize road objects, especially cars, under good weather conditions, is demonstrated. However, this experiment must be reproduced on a larger dataset to extend the obtained results to other road users, such as pedestrians.

Since polarimetric features are known to be invariant to strong illuminations or low visibility (see section 1.2 for more details), they could be a great asset to characterize road scenes when the visibility is altered. This is the reason why the experiments in this section are focused on road scenes under fog. As a matter of fact, color-based features vary a lot with the visibility conditions. Deep color-based architectures, trained on road scenes in good weather conditions, fail to efficiently detect road objects when the visibility is altered. Since polarimetric features are invariant to luminosity changes, they should be able to characterize road objects whatever the weather condition.

In the previous section, only three polarimetric intensities, associated to three rotation angles of the polarizer, are studied. However, the different polarimetric features respectively provide specific complementary information to describe objects. Some polarimetric parameters could be more invariant to visibility changes and thus more adapted than others to describe road scenes in every situation. This is the reason why this experiment also studies the impact of other polarimetric features on road scene analysis.

To summarize up, this experiment investigates the ability of different polarimetric features, learnt in good weather conditions, to describe road objects in adverse weather conditions. To this end, the experimental setup is first described. The results are then exposed before being discussed.

5.3.1 Experimental setup

As a reminder, the goal of this experiment is to evaluate the faculty of polarimetric features to provide information invariant to visibility conditions. To this end, the polarimetric features describing road objects in good weather conditions are extracted using a deep architecture. This deep architecture is then evaluated exclusively on foggy road scenes. If such an architecture provides good detection scores, the invariance of

polarimetric features to weather conditions is proved.

For all of the experiments, the RetinaNet-50 network is used. The reasons for this choice are mentioned in section 5.2. The dataset used for this experiment is detailed in the second row of Table 4.3. The training/validation set contains 2221 images and is thus larger than the one used in the experiment of section 5.2. However, it still does not enable an efficient training from scratch on the polarimetric dataset without over-fitting the network. This is the reason why, once again, a RetinaNet-50 pre-trained on MS COCO is used as a basis for all of the experiments. The first experiment consists in fine-tuning separately this architecture on three different data formats, respectively the intensities images $I = (I_0,\ I_{45},\ I_{90})$, the Stokes images $S = (S_0,\ S_1,\ S_2)$ and pseudo-HSV = $(I_0,\ \phi,\ \rho)$ (more details can be found in section 4.3). The three obtained models are tested on their respective data format. At this stage there were no public dataset containing paired multimodal color-based and polarimetric road scenes for the object detection task. To overcome this limitation and to proceed to the fairest comparison possible, a network pre-trained on MS COCO, which mostly contains road scenes in good weather conditions, is tested on color-based foggy scenes. It is important to note that, the polarimetric and the color-based testing sets are paired and thus contain the same road objects. As mentioned previously, in order to study the ability of polarimetric features to describe a scene whatever the visibility conditions, the training and validation sets contain scenes in good weather conditions and the testing set contains scenes under fog. This experimental setup is illustrated in Figure 5.3.

As for the technical details, the RetinaNet-50 is trained on 50 epochs with an Adam optimizer [257] and a learning rate of 10^{-5}. The weights selected for the evaluation are the one corresponding to the lowest validation loss. Regarding the training/validation set, four fifths of this set are used for the training purpose and the remaining fifth for the validation task.

5.3.2 Results and discussions

Since there are less than 25 instances of the classes bike and motorbike in the training/validation and testing sets, they are not taken into consideration for the evaluation process. Similarly to the previous experiment, the formula used to compute the mAP can be found in equation (2.10). Here, $n = 2$ and $AP_i \in \{AP_{\text{person}}, AP_{\text{car}}\}$. The improvement of polarimetric scores towards RGB ones is quantified using the error rate evolution which has the following formula:

$$ER_o^d = \frac{1 - AP_o^d - (1 - AP_o^{\text{RGB}})}{1 - AP_o^{\text{RGB}}} \times 100 \ , \tag{5.1}$$

where ER_o^d is the percentage of error rate for the object $o \in \{'\text{person}', '\text{car}'\}$ and the polarimetric data format d, AP_o^{RGB} is the average precision for object o with the RGB data format while AP_o^d denotes the average precision after fine-tuning on the object o and the related polarimetric data format d.

The different detection scores can be found in Table 5.2. As can be seen in this table, I and S improve the detection results. The error rate is decreased by 45.9% for the person detection and by 19.4% for the car detection regarding I. As for S, the

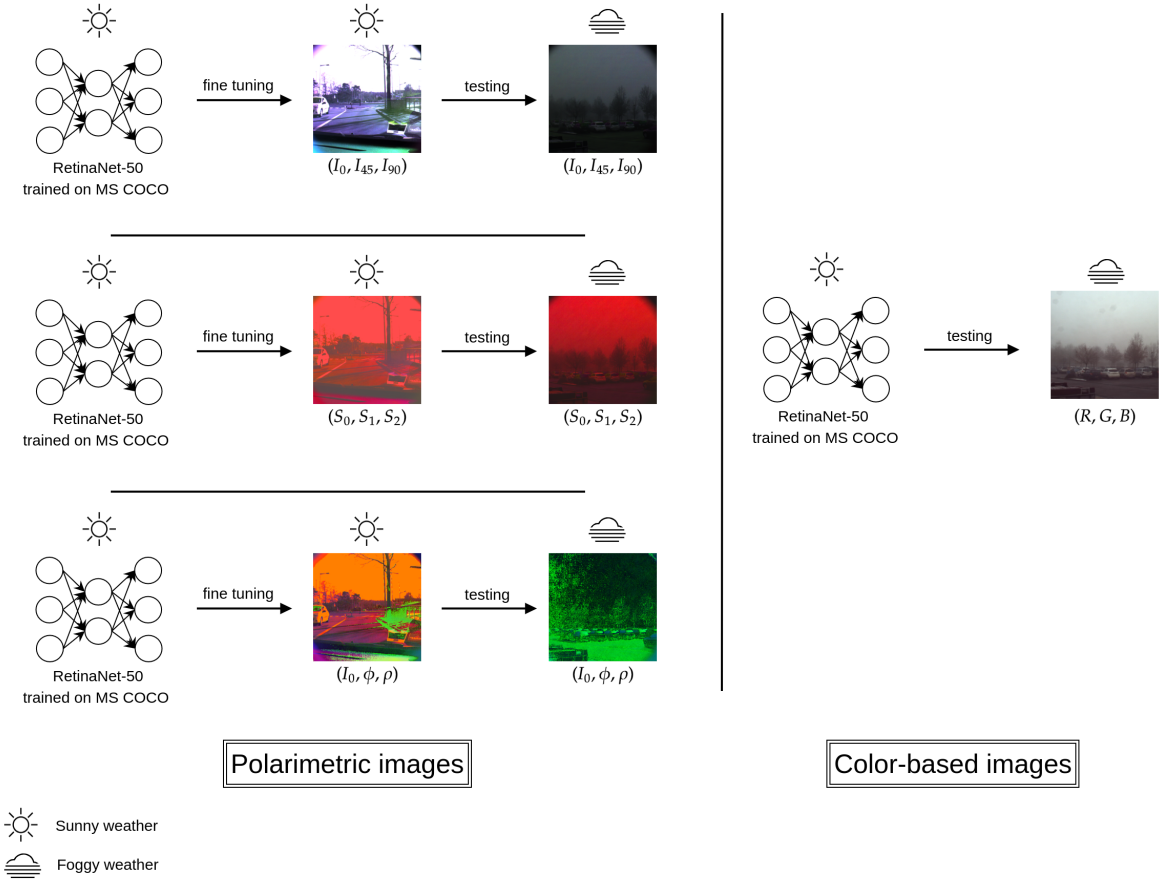

Figure 5.3: Experimental setup. The first column refers to polarimetric data formats, from top to bottom, I, S and pseudo-HSV. Three RetinaNet-50 pre-trained on MS COCO are fine-tuned and tested respectively on I, S and pseudo-HSV. The second column refers to the RGB images. The training set contains scenes in good weather conditions and the testing set contains foggy scenes.

Entries	Class name	AP no FT	AP FT	*ER*
RGB	person	82.54	✘	✘
	car	66.39	✘	✘
	mAP	74.47	✘	✘
I	person	**85.56**	**90.79**	-45.9
	car	60.64	**72.90**	-19.4
	mAP	73.10	**81.85**	-28.9
S	person	42.58	**82.77**	-1.3
	car	21.47	**69.75**	-10.0
	mAP	32.03	**76.26**	-7.0
pseudo-HSV	person	14.99	61.35	121.4
	car	18.89	**67.79**	-4.2
	mAP	16.94	64.57	38.8

Table 5.2: Comparison of the detection with RetinaNet-50 before and after fine tuning. AP no FT and AP FT stand respectively for Average Precision before Fine Tuning and Average Precision after Fine Tuning. In blue, the detection scores on the color-based images (RGB) and in bold all the scores that overcome them. The best score is in green.

error rate is respectively decreased by 1.3% and by 10% regarding the person and the car detection.

The obtained results demonstrate that two polarimetric data formats, I and S, improve object detection in road scenes under fog. As a matter of fact, the deep architectures are trained exclusively on road scenes under good weather conditions and tested exclusively on road scenes under fog. This confirms the intuition that, unlike color-based features, the polarimetric features are more invariant to weather condition changes. The physical information provided by polarimetric parameters is a reliable feature, describing an object when its color or shape are altered by the foggy conditions. Road object detection under fog is illustrated in Figure 5.4 for the three polarimetric data formats and and its RGB equivalent.

This experiment also demonstrates that people are well detected on polarimetric images. Unlike in the experiment carried out in section 5.2, the dataset used for the evaluation step is thought to contain a realistic representation of this class. Instead of having a majority of occluded objects, this dataset contains a good balance between occluded and non-occluded objects.

Finally, this experiment demonstrates that once again pre-trained color-based models can be used as a basis to ease the convergence towards polarimetric features. The different detection tasks are improved after fine-tuning the networks respectively on the three polarimetric data formats. These improvements are illustrated in Figure 5.5.

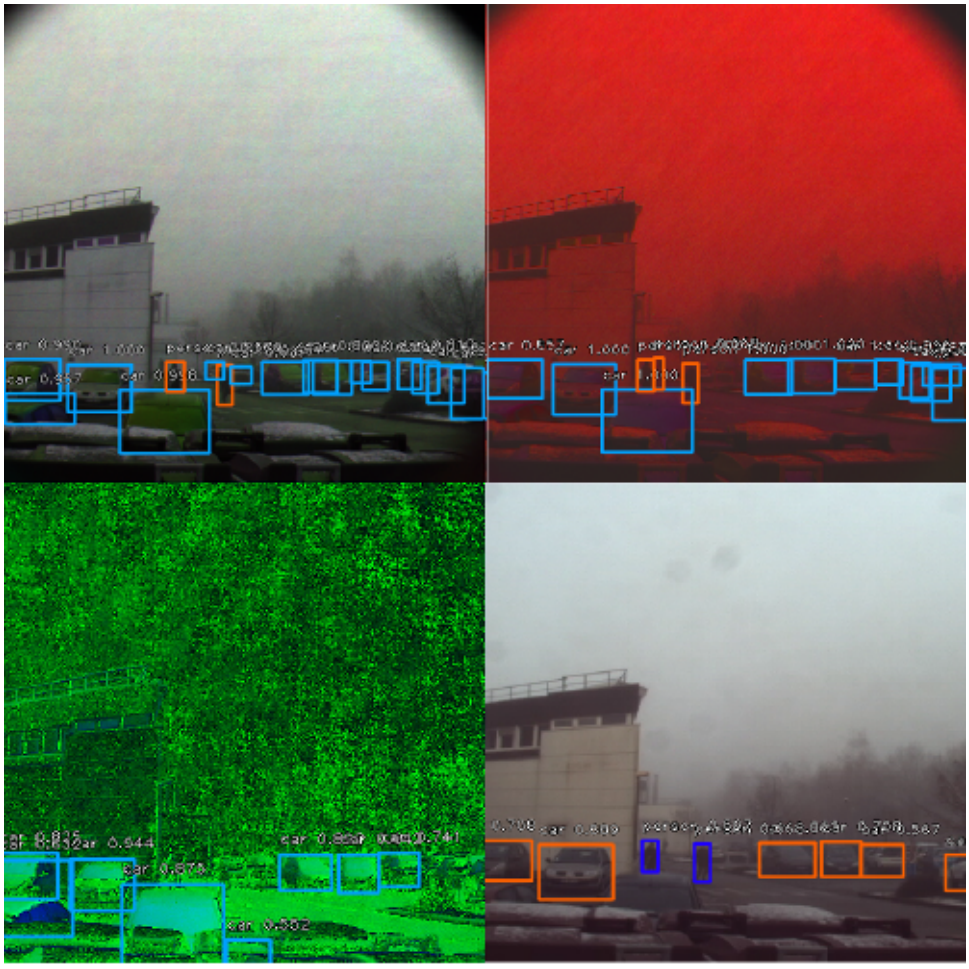

Figure 5.4: Detection results in foggy weather. On top left I, on top right S, on bottom left pseudo-HSV and on bottom right RGB.

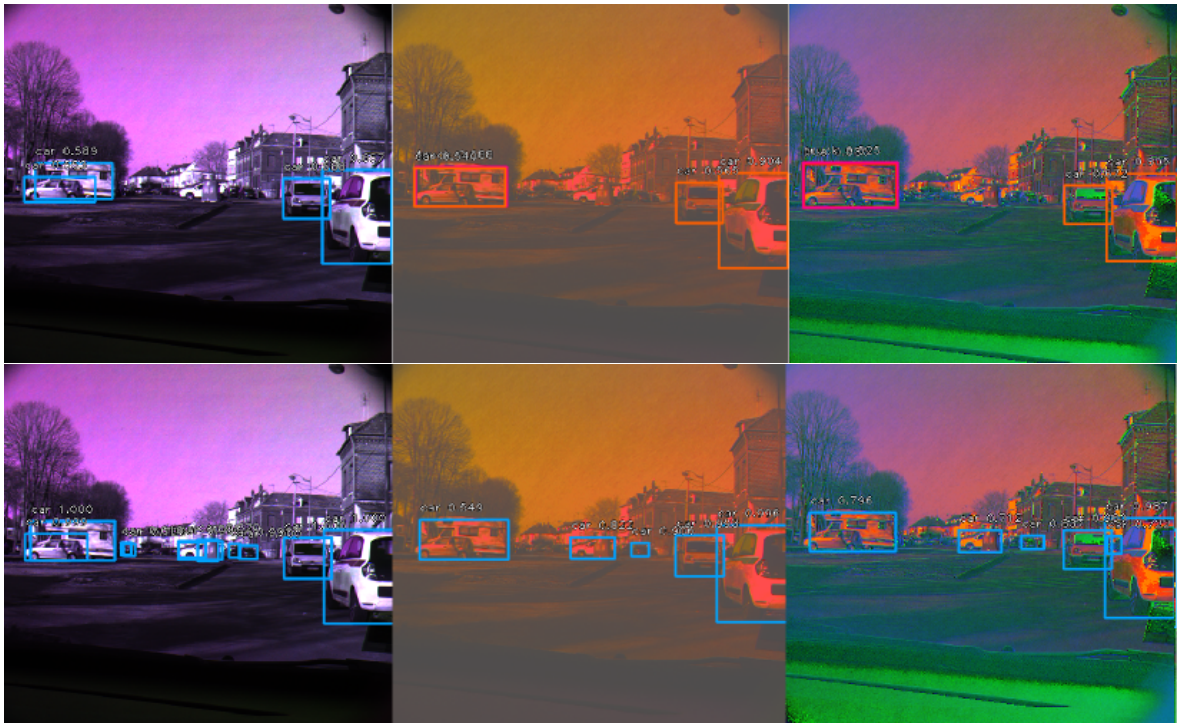

Figure 5.5: Detection results with RetinaNet-50. On top and bottom row, respectively the detection results before and after fine-tuning. From left to right: I, S and pseudo-HSV.

5.4 A deeper study of polarimetric features under fog

In section 5.3, the experiments carried out show encouraging results towards the added value of polarimetric imaging for road scene analysis under fog. However, these results are limited since the dataset used for the experiments does not contain paired multimodal color-based and polarimetric images. This is the reason why it is paramount to reiterate these experiments with a multimodal polarimetric and color-based dataset.

On the other hand, other polarimetric data formats need to be explored. The pseudo-HSV images used in the previous experiment are refactored to contain information homogeneous with the real color-based HSV format. Based on the literature, new polarimetric data formats, closer to the physics of a polarized light wave, are constituted and their impact on road scenes is studied. All these data formats are detailed in section 4.3.

Since only one deep architecture, the RetinaNet-50 network, is studied so far, other backbones to the RetinaNet network are used to reiterate the experiments.

To summarize up, this section aims to confirm the ability of polarimetric features to describe road scenes, no matter the visibility conditions. It also aims to study five polarimetric data formats for this purpose. Finally, other deep architectures are studied. To this end, the different experimental setups are first described and the results are exposed before being discussed.

5.4.1 Experimental setups

The first part of the experiment consists in confirming the ability of polarimetric features to analyze road scenes whatever the visibility conditions. To this end, the PolarLITIS dataset, described in the third row of Table 4.3, is used. This dataset contains 1640 images in the training set and 420 images in the validation set. The training set contains sunny and cloudy images while the validation set contains exclusively cloudy scenes. To evaluate the robustness of polarimetric features to weather changes, the testing set is exclusively constituted of foggy scenes.

It is important to note that, the PolarLITIS dataset contains paired multimodal polarimetric and color-based images. This property enables to perform a strong and reliable comparison between the polarimetric and the color-based evaluation. As a matter of fact, the color-based and polarimetric training processes are carried out on the same content, enabling an evaluation on these two modalities on the same basis.

Similarly to the experiments carried out in sections 5.2 and 5.3, a RetinaNet-50 pre-trained on MS COCO is used as a basis for the experiments. This network is fine-tuned respectively on five polarimetric data formats, respectively the intensities images $I = (I_0, I_{45}, I_{90})$, the Stokes images $S = (S_0, S_1, S_2)$, the HSV images HSV = (ϕ, ρ, S_0) the Pauli inspired images Pauli = (S_1, I_{45}, S_0) and the Poincaré inspired images $P = (S_0, \rho\cos(2\phi), \rho\sin(2\phi))$ and on RGB images to perform a fair comparison. This experimental setup is illustrated in Figure 5.6.

The second experiment consists in studying different deep architectures to extract physical features in polarimetric images. To this end, different backbones are used for the RetinaNet architecture, among which, ResNet-50, ResNet-101 [166], VGG16 and VGG19 [164]. Since, at this point, not all these deep architectures pre-trained on MS COCO were publicly available, another training process is designed. Training a DNN from scratch on such a large dataset can take several weeks and a lot of computing resources. This is the reason why this possibility is excluded. As all these backbones can be found pre-trained on ImageNet [63], they are used as a basis for this second experiment. Even if ImageNet is designed for object classification, it is proved that using a network that has converged into another domain makes transfer learning possible and efficient [258].

However, to perform object detection, a transfer learning step on a dataset designed for this purpose is paramount. The KITTI dataset [36] is used for this task since it contains road scenes in good weather conditions and its content is close to the PolarLITIS dataset. Moreover, because KITTI is not as large as MS COCO, it enables an efficient training in a reasonable amount of time. To summarize the second experiment, the four constituted architectures (i.e. RetinaNet using the four different backbones) are initialized with backbones pre-trained on ImageNet. These architectures are then used to perform transfer learning on KITTI. Finally, the obtained networks are fine-tuned on the PolarLITIS dataset on each of the five polarimetric data formats separately and on the RGB images before being tested on their respective data format. As a reminder, the training process is done using road scenes in good weather conditions and the obtained architectures are evaluated on road scenes under fog. This training process is illustrated for the intensities images in Figure 5.7 and can be extended to the others data formats.

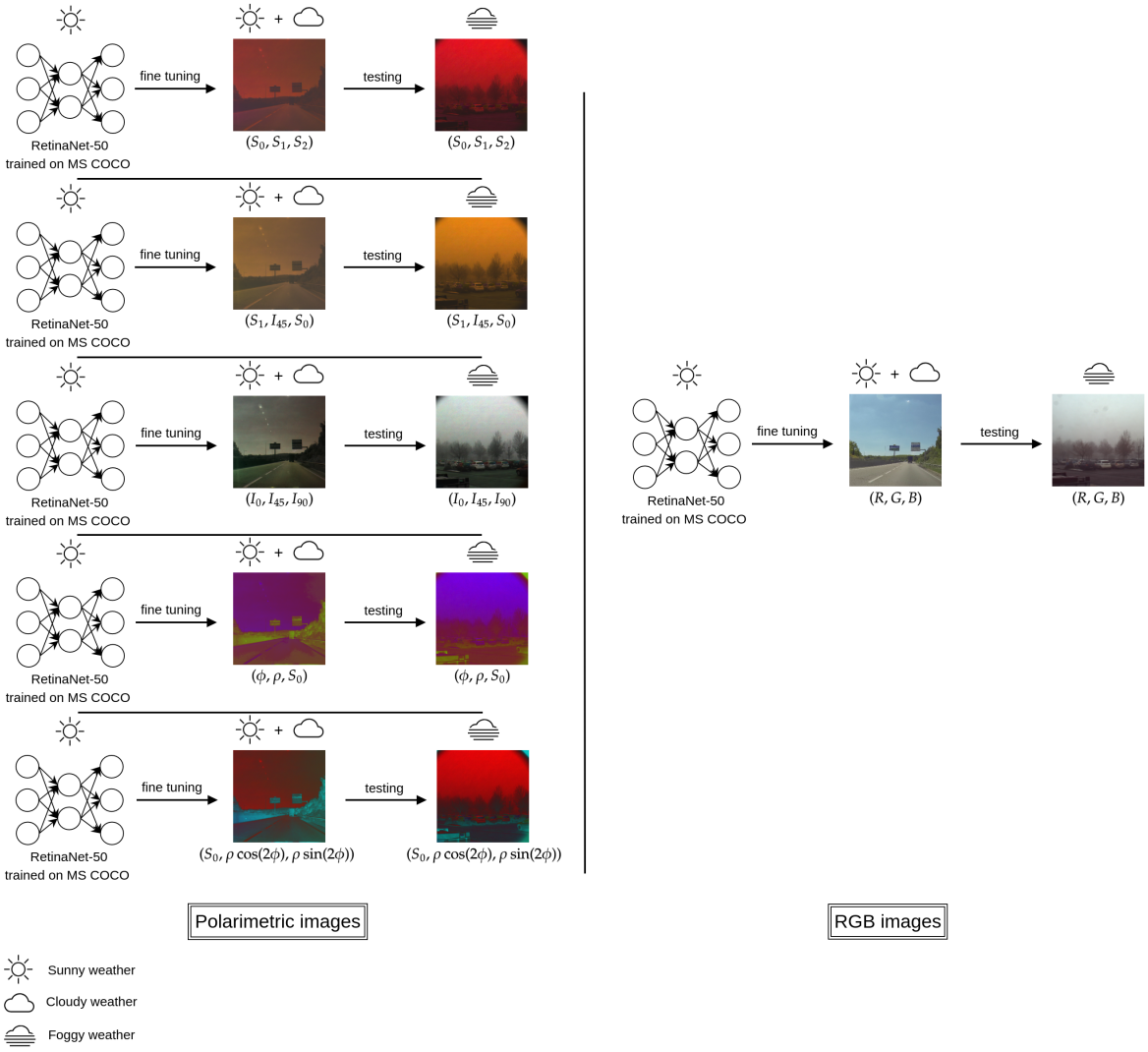

Figure 5.6: Experimental setup. Here, RetinaNet-50 pre-trained on MS COCO is fine-tuned on each data format separately.

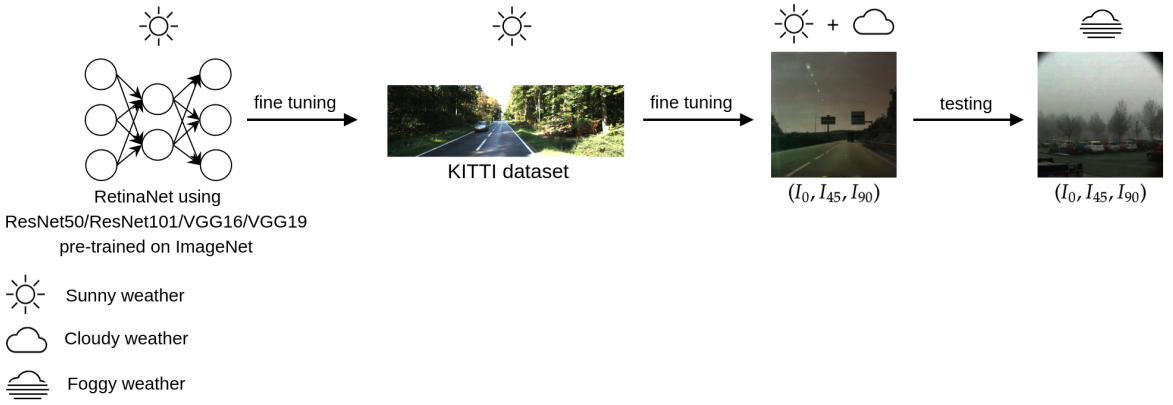

Figure 5.7: Experimental setup. Here, a backbone pre-trained on ImageNet is used as a basis. Transfer learning is then performed with this architecture on KITTI and then fine-tuned on the PolarLITIS dataset (polarimetric formats and RGB).

As for the training details, regarding the first experiment, the RetinaNet-50 is trained on 20 epochs on the PolarLITIS dataset. Regarding the second experiment, the four architectures are trained on 20 epochs on both the KITTI and the PolarLITIS datasets. For both experiments, the Adam optimizer using a learning rate of 10^{-5} is used. The optimal weights for each training process are the ones associated to the lowest value of the validation loss.

5.4.2 Results and discussions

Similarly to the experiments carried out in sections 5.2 and 5.3, since there are not enough instances of the classes bike and motorbike in the training set, they are excluded from the evaluation. The formula used to compute the mAP can be found in equation (2.10). Here, $n = 2$ and $AP_i \in \{AP_{\text{person}}, AP_{\text{car}}\}$. The error rate evolution is computed to quantify the improvements of the detection scores associated to the polarimetric data formats towards the ones associated to RGB following equation (5.1).

The different detection scores of the first experiment can be found in Table 5.3. As can be seen on this table, there are three polarimetric data formats that overcome the RGB modality regarding road object detection under fog. I, S and the Pauli inspired images respectively improve road object detection by 26.7%, 9.1% and 20.3%. This implies that the polarimetric intensities and the polarimetric Stokes parameters are more invariant to weather changes than the RGB images and therefore more reliable to describe road objects when the visibility is altered. Figure 5.8 shows an example of road object detection under fog on the same scene using the RGB modality and the five polarimetric data formats.

The different detection scores of the second experiment are summarized up in Table 5.4. This experiment confirms that I, S and the Pauli inspired images achieve the best detection scores. However, the obtained results using RetinaNet pre-trained on KITTI are not as good as the ones obtained using a RetinaNet pre-trained on MS COCO. This can be explained by the fact that MS COCO contains more than

Format	RGB	I	ER^I	S	ER^S	Pauli	ER^{Pauli}	HSV	ER^{HSV}	P	ER^P
car	73.28	**77.18**	-14.6	72.43	3.2	**75.09**	-6.8	30.86	158.8	39.10	127.9
person	80.97	**89.27**	-43.6	**86.01**	-26.5	**88.45**	-39.3	57.52	123.2	69.14	62.2
mAP	77.13	**83.23**	-26.7	**79.22**	-9.1	**81.77**	-20.3	44.19	144.0	54.12	100.6

Table 5.3: Comparison of the detection using RetinaNet-50 on the different polarimetric data formats and on RGB images. In blue, the RGB detection scores in percentage, in bold the polarimetric detection scores that overcome it and in green the best detection score.

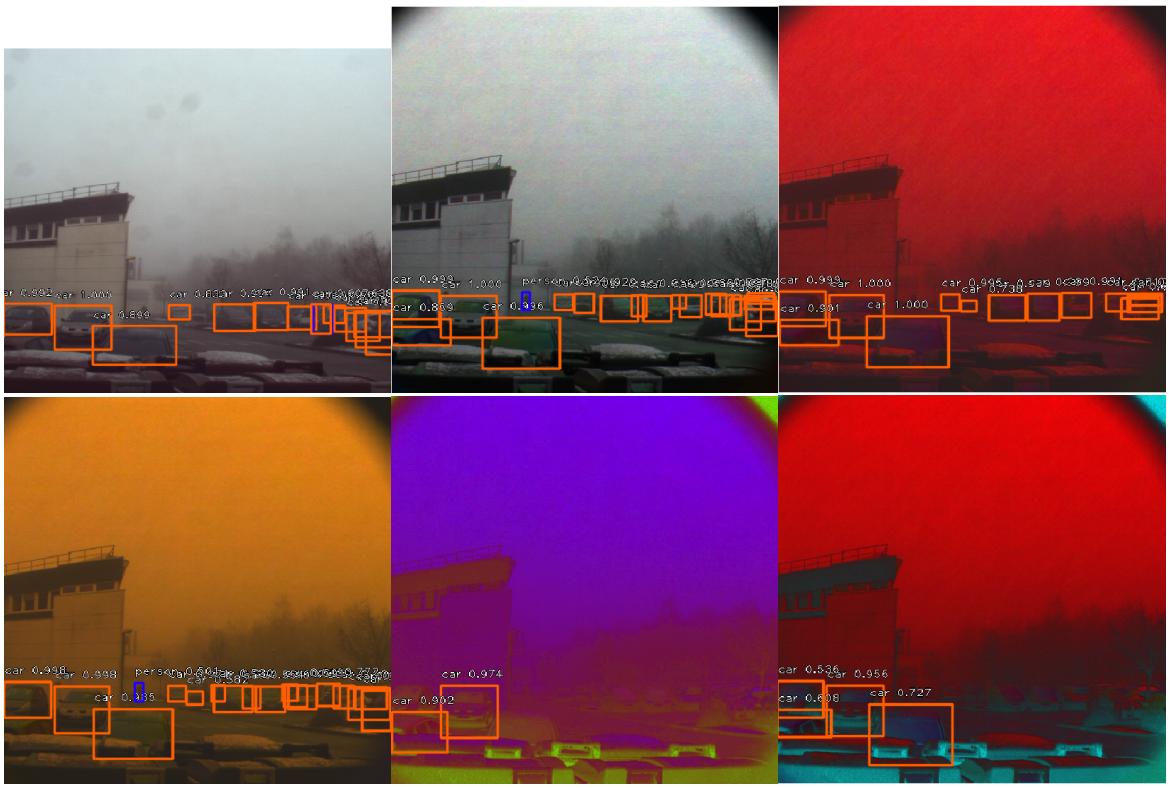

Figure 5.8: Detection using RetinaNet-50 on an RGB foggy scene and its polarimetric equivalent. From left to right and from top to bottom, RGB, I, S, Pauli, HSV and P.

Backbone	Classes	RGB	I	ER^I	S	ER^S	Pauli	ER^{Pauli}	HSV	ER^{HSV}	P	ER^P
VGG16	car	4.96	**68.09**	-66.4	**64.72**	-62.9	**60.28**	-58.2	**38.27**	-35.0	**38.47**	-35.3
	person	12.76	**72.59**	-68.6	**55.73**	-49.3	**59.50**	-53.6	**47.58**	-39.9	**56.88**	-50.6
	mAP	8.86	**70.34**	-67.5	**60.23**	-56.4	**59.89**	-56.0	**42.93**	-37.4	**47.68**	-42.6
VGG19	car	2.89	**64.46**	-63.4	**64.53**	-63.5	**60.76**	-59.6	**30.0**	-27.9	**41.10**	-39.3
	person	37.71	**73.72**	-57.8	**68.41**	-49.3	**49.48**	-18.9	**48.26**	-16.9	**49.92**	-19.6
	mAP	20.30	**69.09**	-61.2	**66.47**	-57.9	**55.12**	-43.7	**39.13**	-23.6	**45.51**	-31.6
ResNet50	car	22.43	**57.99**	-45.8	**63.16**	-52.5	**61.54**	-50.4	**26.65**	-5.4	**42.58**	-26.0
	person	33.95	**64.36**	-46.0	**64.25**	-45.9	**69.31**	-53.53	**36.11**	-3.3	**41.57**	-11.5
	mAP	28.19	**61.18**	-45.9	**63.71**	-49.5	**65.43**	-51.9	**31.38**	-4.4	**42.08**	-19.3
ResNet101	car	31.29	**59.51**	-41.1	**64.12**	-47.8	**64.90**	-48.9	21.26	14.6	30.49	1.2
	person	35.82	**63.94**	-43.8	**64.11**	-44.1	**69.20**	-52.0	19.37	25.6	25.39	16.3
	mAP	33.56	**61.73**	-42.4	**64.12**	-46.0	**67.05**	-50.4	20.32	19.9	27.94	8.5

Table 5.4: Comparison of the detection using RetinaNet with different backbones on the different data formats. In blue, the scores achieved by the RGB images, in bold all the scores that overcome them and in green the best detection scores.

220k images whereas KITTI only contains 7k images. RetinaNet pre-trained on MS COCO provides more general characteristics to describe road objects as it is larger than KITTI. The best results are achieved using a network pre-trained on a more general and larger dataset. We can conclude that the choice of the dataset on which the network is pre-trained is more relevant than the choice of the architecture itself.

The results of the two experiments enable to highlight the robustness of polarimetric features to weather changes. The physical features provided by polarimetric images to describe road objects are invariant to visibility changes. Polarimetric features are therefore a strong asset to analyze road scenes in every situation. Unlike color-based features, polarimetric features guarantee a reliable object detection in case of an unexpected alteration of the visibility, such as foggy weather, increasing road users' safety.

5.5 Summary

In this chapter, the ability of DNN to perform an efficient road object detection in polarimetric images is first explored. The RetinaNet architecture is chosen for this task, since it performs an efficient training on unbalanced datasets and processes images in real time. It proves that the car detection score achieved by polarimetric images is in the same range as the one achieved in the state of the art using RGB images. As a matter of fact, windshield and metallic bodywork are very reflective surfaces, more polarized than non-reflective ones, providing relevant features to characterize a car, regardless of its shape.

Following these encouraging results, the problem of road object detection under fog using polarimetric features is addressed. A second experiment uses a larger dataset containing sunny polarimetric images in its training set and foggy multimodal (polarimetric and color-based) images in its testing set. A deep architecture is trained respectively on three polarimetric data formats, I, S and pseudo-HSV, in good visibility conditions and tested on foggy scenes. The obtained results are compared with

the ones obtained in the same foggy RGB scenes, by using the same deep architecture trained on MS COCO. The obtained results give an intuition on the invariance of polarimetric features to foggy scenes, enabling a reliable person and car detection.

A final experiment confirms that polarimetric features characterizing road objects in good weather conditions are still valid to detect objects in foggy scenes, unlike color-based ones. The PolarLITIS dataset, containing multimodal polarimetric and color-based images, is used for this experiment. The polarimetric and RGB training processes are carried out in the same basis using this dataset this time. The experimental results prove that three polarimetric data formats, I, S and Pauli, enable good detection results in foggy scenes. Knowing that adverse weather conditions are not the most common in road scenes, providing information that describe an object, whatever the visibility conditions, makes autonomous vehicles more robust to unexpected changes. In the next chapter, several fusion schemes are explored to enhance road object detection under fog and the experiments are extended to a wide range of weather conditions.

Chapter 6

Polarimetric and color fusion

Contents

6.1 Introduction

The previous chapter demonstrates that polarimetric features are more robust than color-based ones, especially to weather changes. These results are based on the fact that a deep architecture trained on polarimetric road scenes in good weather conditions are able to detect road objects more efficiently than the same color-based architecture. However, these results could be improved by fusing the different modalities. On top of that, other adverse weather conditions need to be explored to generalize the obtained results to all kind of visibility alterations.

A first experiment explores several fusion schemes to find the most adapted one. These fusions schemes are tested with several polarimetric data formats and color spaces since they provide complementary information. This process enables to find the best modalities combination to enhance road scene analysis under fog.

The second part of this chapter validates all the experimental results of this thesis on other weather conditions. The best fusion schemes are applied to well chosen polarimetric and color-based features to enhance road object detection in several fog

densities and under tropical rain. This pipeline enables to extend the obtained results to other situations where the visibility is altered and to find its limitations.

6.2 Multimodal fusion

In Chapter 5, the added value of polarimetric features for road scenes analysis under fog is shown. However, when the visibility is altered, autonomous vehicles can not rely exclusively on one sensor's information to make their decisions. In fact, any perturbation in the sensor could alter the scene analysis and thus directly impact the final decision. Combining information from multiple sensors is a guarantee of safety since, in case of any perturbation in a sensor, the functional sensors can backup a broken one [34]. Moreover, information provided by multimodal sensors are complementary and enable to describe a road scene under different angles [259, 260]. Combining information from multiple and multimodal sensors is therefore paramount to reinforce the predictions when the visibility is altered [225].

Previous work [22] have shown that color-based features combined with polarimetric ones help enhancing car detection. However, the experiments are carried out under good weather conditions with a Deformable Part Model (DPM). This work focuses on the exploration of various fusion schemes between polarimetric and color features, using deep architectures. Several color spaces are explored to select the most relevant ones. Finally, two data registration methods are explored in order to palliate the problem of fusing non-stackable pixelwise images with an inconsistent offset.

6.2.1 The different fusion schemes

In order to improve object detection under fog, six fusion schemes are explored. The different fusion schemes are the followings and summed up in Figure 6.1:

1. Early fusion: the two images are stacked in order to create a six-channels image to be processed by the neural network,

2. Late fusion with naive NMS filter: the two images are processed separately by two different neural networks. The raw predicted bounding boxes are concatenated before being filtered by the NMS algorithm [75] (see algorithm 1),

3. Late fusion with naive soft-NMS filter: the two images are processed separately by two different neural networks. The raw predicted bounding boxes are concatenated before being filtered by the soft-NMS algorithm [261] (see algorithm 1),

4. Late fusion with double soft-NMS filter: the two images are processed separately by two different neural networks. The raw predicted bounding boxes of each modality are filtered separately a first time by the soft-NMS algorithm. The filtered boxes of each modality are then concatenated before being filtered a second time using the soft-NMS algorithm,

5. Late fusion with OR filter: the two images are processed separately by two different neural networks. The raw predicted bounding boxes of each modality are

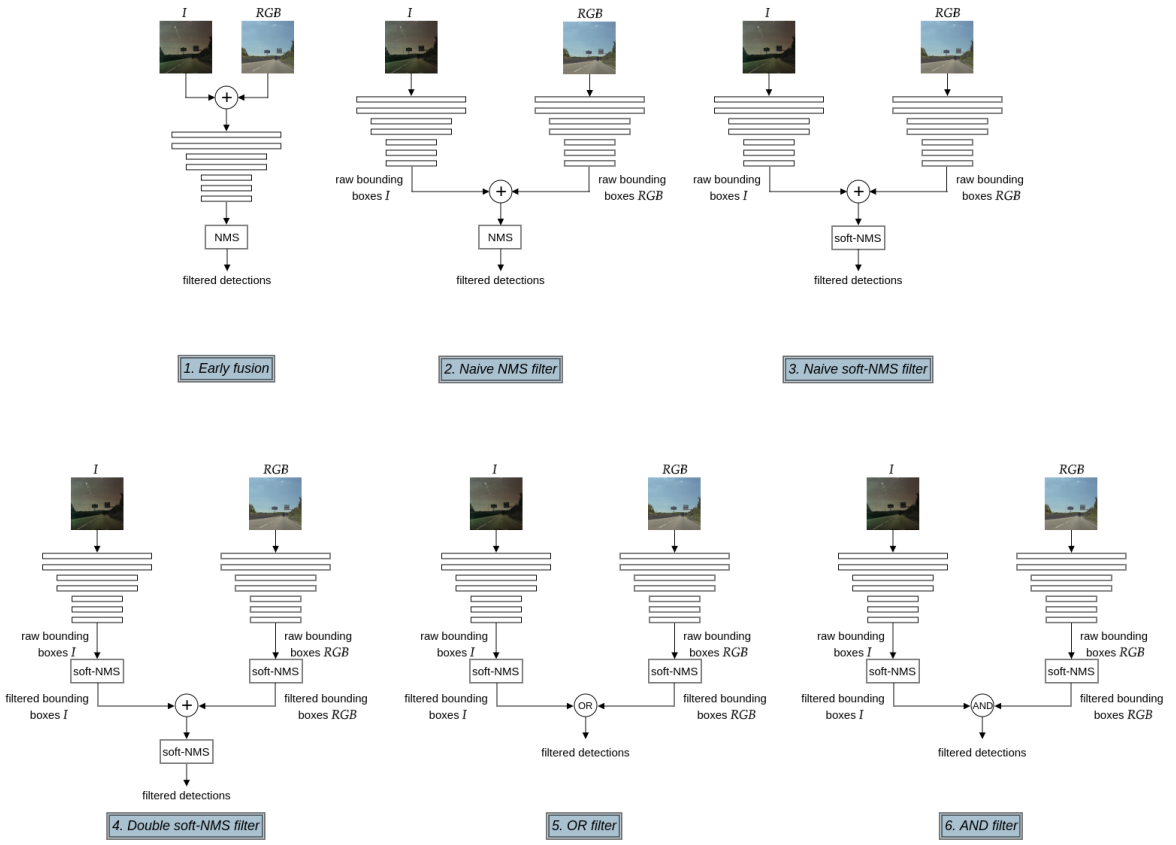

Figure 6.1: Illustration of the different fusion schemes. Here the fusion between RGB and intensities images $I = (I_0, I_{45}, I_{90})$ is illustrated (see Table 4.4 for more details) and can be extended to the other modalities combinations.

filtered separately a first time by the soft-NMS algorithm. Then an OR operation is made between these two sets of bounding boxes,

6. Late fusion with AND filter: the two images are processed separately by two different neural networks. The raw predicted bounding boxes of each modality are filtered separately a first time by the soft-NMS algorithm. Then an AND operation is made between these two sets of bounding boxes.

The PolarLITIS dataset is used for all the experiments. This dataset contains paired information from a color-based and a polarimetric sensor. The two paired images are therefore non-stackable pixelwise. The early fusion scheme is sensitive to data misalignment [34], which is the case for the color and polarimetric images in this dataset. Nevertheless, this fusion scheme is investigated to check the ability of the network to palliate the sensors' offset before image registration at first. In a second time, after image registration, the ability of this fusion scheme to address road object detection under fog is evaluated. However, early fusion relies on the information of both sensors to make an accurate prediction. An alteration of the signal of one of the sensors greatly impacts the predictions which makes it not robust to sensor breakdown.

Regarding the filters used for the late fusion purpose, five different configurations

Figure 6.2: From left to right, the detections on the intensities images $I = (I_0, I_{45}, I_{90})$, on the Pauli inspired images Pauli $= (S_1, I_{45}, S_0)$ and the fusion of these two modalities using a naive NMS filter. A loss of information can be noticed regarding the prediction of the fused modalities when cars are parked one behind another. This is due to the suppression of close bounding boxes when the naive NMS filter has to process too many of them.

are chosen. As mentioned in [261], when two predicted bounding boxes have a high IOU (see Figure 2.9), the NMS algorithm will only keep the one with the highest score . In the late fusion scheme, the number of predicted bounding boxes is doubled since each modality is processed separately before their concatenation. Thereby, in case of close objects (e.g. a line of cars parked behind one another), the NMS algorithm suppresses relevant bounding boxes as it is illustrated in Figure 6.2. Even if the soft-NMS filter is designed to palliate this problem, it shows limits when there are too many bounding boxes to process. This is the reason why a third late fusion scheme is proposed, the Double soft-NMS filter. This filter reduces the number of bounding boxes to process by filtering each prediction separately a first time before concatenating them. The lightened bounding boxes concatenation is filtered a second time to get the final predictions. It is worth noticing that, all the hyper-parameters used in the experiments are found experimentally, i.e. several values are tested and the optimal ones are selected. For the soft-NMS filter, the hyper-parameter σ of the Gaussian penalty function is set to $\sigma = 1$. As for the Double soft-NMS function, the hyper-parameters σ_{polar} and σ_{color} of the Gaussian penalty functions, respectively associated to the soft-NMS filtering polarimetric and color predictions, are set to $\sigma_{\text{polar}} = 0.4$ and $\sigma_{\text{color}} = 2.0$. Finally, the hyper-parameter σ_{final} of the Gaussian penalty function of the soft-NMS filter, applied to the combination of the previously filtered predictions of the two modalities, it is set to $\sigma_{\text{final}} = 1.0$.

The two other designed filters are meant to target the false negative and false positive rates. Regarding the OR filter, after getting the two sets of filtered bounding boxes from each modality, there are two possibilities. If the bounding box is detected in one modality and not in the other, it is kept. If the bounding box is detected in the two modalities, only the one with the best detection score is kept. This filter reduces the false negative rate and its details are presented in algorithm 3. As for the AND fusion filter, the final bounding boxes are the ones detected in both modalities. This filter reduces the false positive rate and its details are presented in algorithm 4.

Algorithm 3 Or filter algorithm. In this work, lThresh=0.05 and uThresh=0.89 and were found experimentally, i.e. several values of each parameter are tested while fixing the other to find the optimal ones.

Require: Detections of the two modalities : detectionsMod1, detectionsMod2
  detectionsFusion ← detectionsMod1
  for d2 ∈ detectionsMod2 **do**
    for d1 ∈ detectionsMod1 **do**
      if IOU(d1, d2) < lThresh **then**
        Add d2 to detectionsFusion
      else if IOU(d1, d2) > uThresh & score(d2) > score(d1) **then**
        Replace d1 by d2 in detectionsFusion
      end if
    end for
  end for
  return detectionsFusion

Algorithm 4 AND filter algorithm. In this work, thresh=0.55 and was found experimentally, i.e. several values of this parameter are tested to find the optimal one.

Require: Detections of the two modalities : detectionsMod1, detectionsMod2
  detectionsFusion ← ∅
  for d1 ∈ detectionsMod1 **do**
    IOUtemp ← 0
    maxDetection ← ∅
    for d2 ∈ detectionsMod2 **do**
      if IOU(d1, d2) > thresh & IOU(d1, d2) > IOUtemp **then**
        IOUtemp ← IOU(d1, d2)
        if score(d1) ≥ score(d2) **then**
          maxDetection ← d1
        else
          maxDetection ← d2
        end if
      end if
    end for
    if maxDetection ≠ ∅ **then**
      Add maxDetection to detectionsFusion
    end if
  end for
  return detectionsFusion

The late fusion scheme, unlike the early fusion process, enables to analyze separately the information provided by the two sensors. The functional sensor is thus not impacted by the other one's potential perturbations. Thereby, it still provides a reliable road scene analysis.

6.2.2 Data registration

In this section, the different registration processes for non-aligned data are presented. The registration of the predicted bounding boxes of the color modality towards the polarimetric modality is first described. Then, the other registration scheme, which consists in registering color images towards the polarimetric ones, is detailed.

Bounding box registration

Because the color and polarimetric images are not stackable pixelwise, the color bounding boxes are readjusted towards the polarimetric ones using two functions. The first function expresses the polarimetric abscissas of the bounding boxes (x_{polar}) against the color abscissas (x_{color}) of the bounding boxes and the second one expresses the polarimetric ordinates of the bounding boxes (y_{polar}) against the color ordinates (y_{color}) of the bounding boxes. These functions are found by plotting several polarimetric bounding boxes coordinates against color ones and applying a linear regression to this scatter plot. The obtained functions are the followings:

$$\begin{aligned} x_{\text{polar}} &= \max(0, \min(0.919 x_{\text{color}} - 15.4, W_{\text{polar}})) \ , \\ y_{\text{polar}} &= \max(0, \min(1.04 y_{\text{color}} - 83, H_{\text{polar}})) \ . \end{aligned}$$

where W_{polar} is the width of the polarimetric image and H_{polar} is the height of the polarimetric image.

Figure 6.3 illustrates the superposition of the bounding boxes of the two modalities before and after registration.

This registration process enables to recover information from the two modalities while keeping the original images intact. However, it might be limited by the inconsistent offset between the different pair of images in the PolarLITIS dataset. This is the reason why this experiment also explores a more adapted adapted process which is image registration. This process would enable to overcome data misalignment penalizing early fusion.

Image registration

As mentioned previously, the polarimetric and color images are not stackable pixelwise. Since the images from the two modalities are selected automatically according to their sensors' theoretical fps, which do not exactly match their sensors' actual fps, there is a non-consistent temporal offset between the two modalities. An illustration of the offset variation can be found in Figure 6.4. This implies that the spatial offset between two paired images can not be generalized to the whole dataset.

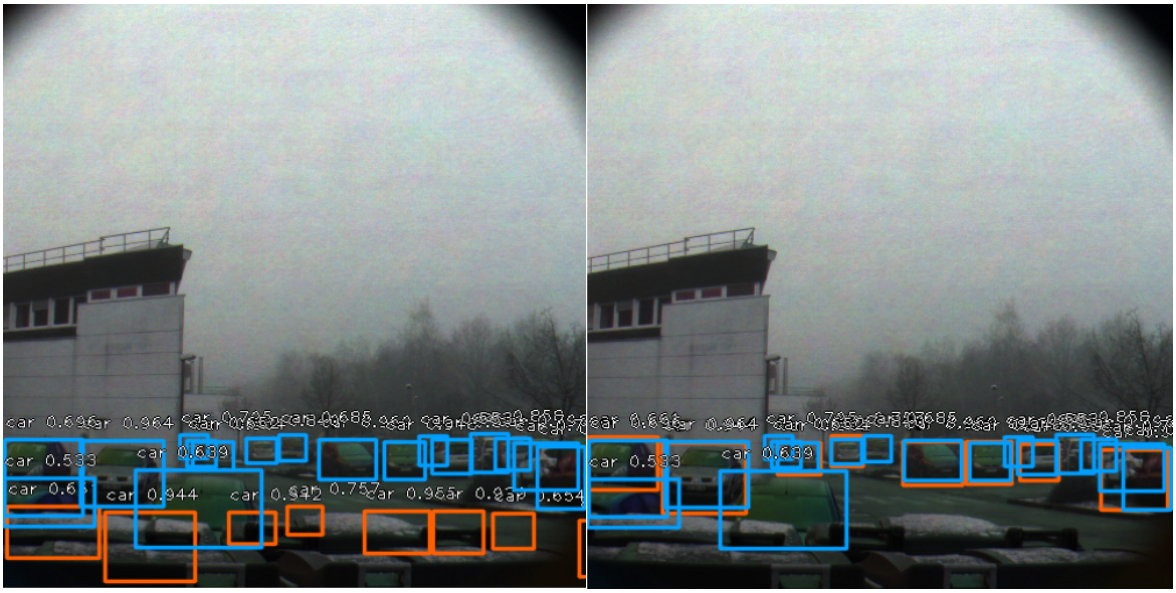

Figure 6.3: Illustration of the RGB (orange) and polarimetric (blue) predicted bounding boxes projected on polarimetric images. On the left, the RGB bounding boxes are projected towards the polarimetric images without bounding boxes registration. On the right, the RGB bounding boxes are projected towards the polarimetric images after bounding boxes registration.

To overcome this limitation, it is paramount to find a method that adapts the registration of color images towards polarimetric regardless of the offset variations. To achieve this goal, a CycleGAN [262] is trained on the paired color and polarimetric images. By training a CycleGAN in this specific configuration, the optimal projection of each color image towards its polarimetric equivalent is found. The registered RGB images are thus stackable pixelwise with the polarimetric ones, enabling each pair of images from the two modalities to contain the exact same content. It is important to note that there are three weather conditions in the PolarLITIS datasets, sunny, cloudy and foggy weather. These three weather conditions corresponding to three different domains, three CycleGANs are respectively trained on them separately. These CycleGANs are trained on 400 epochs each, with a learning rate $\kappa = 2 \times 10^{-4}$, decreased linearly to 0 from epoch 100. To restore the sharp details potentially lost during the generation process, the guided filter [263] is used as a post-processing step. An illustration of the registration setup can be found in Figure 6.5. Examples of registered images using this setup in the three weather conditions can be found in Figure 6.6.

6.2.3 Experimental setup

A RetinaNet network is used as a basis for all the fusion schemes. The motivations for this choice are detailed in section 5.2. To reinforce the obtained results, three different backbones are used in the RetinaNet network, respectively ResNet-50, ResNet-101 and ResNet-152 [166]. For the early fusion scheme, the backbones are modified to take a six-channels image in entry.

Figure 6.4: Illustration of the offset variation. The first row is the polarimetric images represented by I_0 and the second row their RGB equivalent. In the first column, the polarimetric and RGB images are almost stackable whereas in the second and third columns a slight temporal offset is noticed between the RGB and polarimetric images.

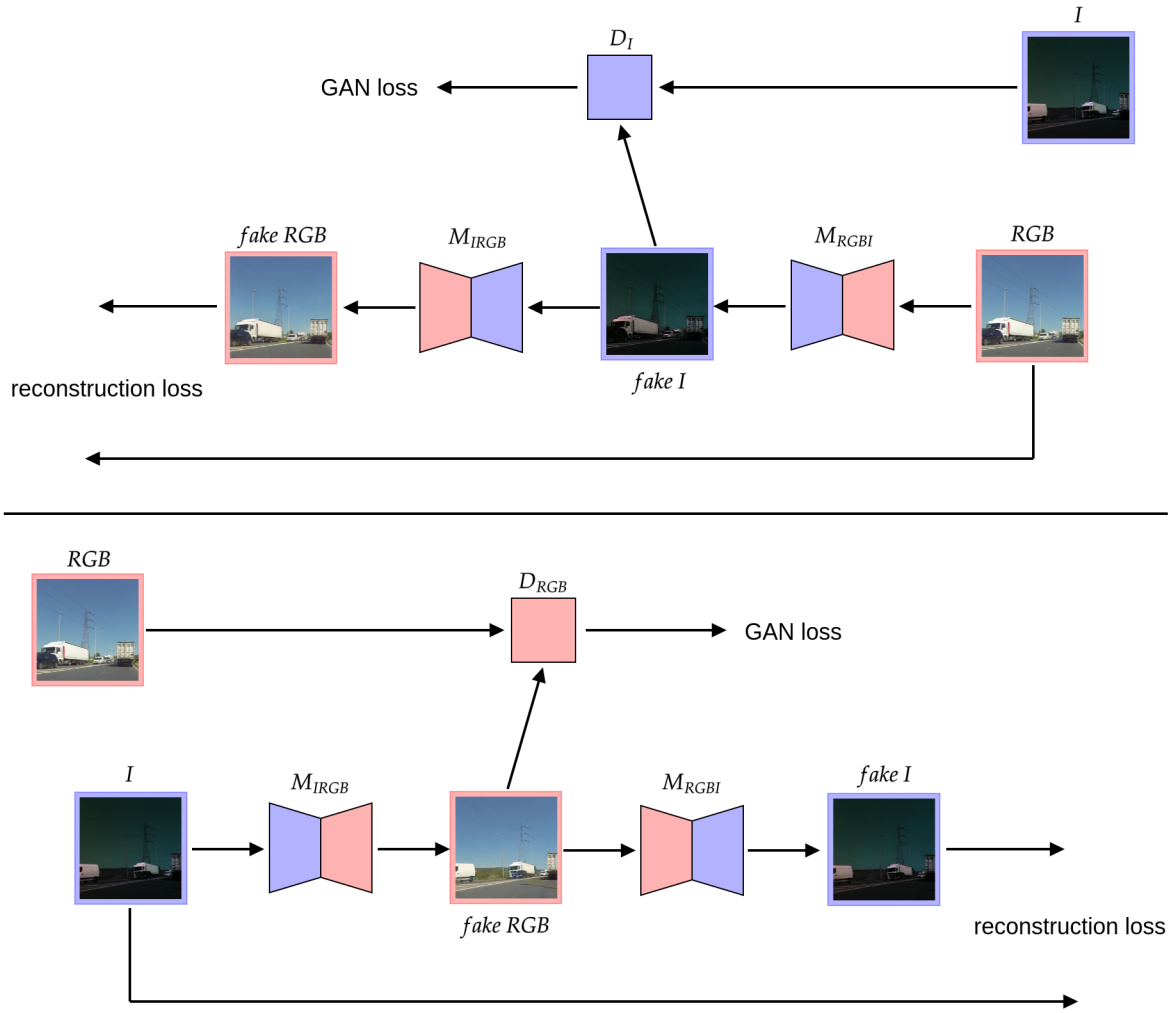

Figure 6.5: Illustration of the image registration process using a CycleGAN. Here, the CycleGAN is trained on paired polarimetric I and RGB images. D_{RGB} and D_I are the discriminators that respectively evaluate the distance between the generated RGB image and the real one and the generated polarimetric image and the real one. $M_{I\mathrm{RGB}}$ and $M_{\mathrm{RGB}I}$ are respectively the generator of RGB modality from the polarimetric modality I and the generator of the polarimetric modality I from the RGB one.

Figure 6.6: Example of registered images using a CycleGAN. The first column contains the real RGB images, the second column contains polarimetric intensities images I and the third column contains the RGB images that are registered towards I.

Because the PolarLITIS dataset only contains 1640 images in its training set and 420 in its validation set, it is paramount to pre-train the network on another dataset to avoid over-fitting [264]. Since the intended purpose is detecting objects in road scenes, the network is pre-trained on the BDD100K dataset [265] from scratch. This dataset is rather large and aims to perform object detection in road scenes. It is thus adapted to perform this task. Regarding the early fusion scheme, each image is stacked with itself to constitute a six-channels image for the pre-training purpose to get the desired dimensions in entry. The obtained networks are then fine-tuned on their respective data format or combination of formats regarding the early fusion scheme.

Based on the results obtained in section 5.4, the most relevant polarimetric data formats are selected. These data formats include the intensities images $I = (I_0, I_{45}, I_{90})$, the Pauli inspired images Pauli $= (S_1, I_{45}, S_0)$ and the Stokes images $S = (S_0, S_1, S_2)$ (see section 4.3 for more details). Four different color spaces are used, including RGB, HSV, CIE Lab and YCrCb (see section 1.3 for more details). This experiment aims to select the most adapted color space for polarimetric and color fusion.

As for the training hyper-parameters, all the architectures are trained with the Adam optimizer with a learning rate of 10^{-5}. The network is trained on BDD100K until convergence (i.e. until the validation loss reaches its lowest value) for all the architectures. The obtained networks are trained for 40 epochs on each data format or combination of data formats. The weights are selected according to the lowest value of the validation loss. All the training processes are repeated five times to provide reliable results.

6.2.4 Discussion and results

In order to prove the efficiency of the fusion schemes it is important to compute the detection scores on the different data formats separately. These results are summed up in Table 6.1. As one can see, the three polarimetric data formats overcome the color spaces when it comes to object detection in road scenes under fog. Regarding the color-based features, the RGB color space achieves the best detection scores with a mAP of 71.63%, which implies respectively scores of 78.73% and 64.53% for person and car detection. The same experiment results in a mAP of 80.82% for polarimetric intensities images I which implies respectively scores of 89.61% and 72.02% for person and car detection. The intensities images, providing the highest scores, are used as the reference to evaluate the efficiency of each fusion scheme from now on.

Regarding the different fusion schemes, their results can be found in Table 6.2 for bounding boxes registration and in Table 6.3 for image registration. To evaluate the increase in detection scores, the error rate evolution is computed as follows:

$$ER_o^{I+M} = \frac{1 - AP_o^{I+M} - (1 - AP_o^I)}{1 - AP_o^I} \times 100 \ ,$$

where ER_o^{I+M} is the error rate evolution associated to the fusion scheme between I and the modality $M \in \{S$, RGB, HSV, CIE Lab, YCrCb$\}$ for object $o \in \{'\text{person}', '\text{car}', mAP\}$, AP_o^I is the average precision for object o with I, used as the reference score, while AP_o^{I+M} denotes the average precision on the object o and the related fusion

Backbone	Class	RGB	HSV	CIE Lab	YCrCb	I	Pauli	S
ResNet-50	person	72.52 ± 1.3	10.30 ± 3.1	61.94 ± 1.6	63.98 ± 4.7	**83.73 ± 1.1**	**83.19 ± 2.7**	**80.47 ± 1.3**
	car	58.55 ± 3.4	3.74 ± 0.3	50.14 ± 4.0	47.82 ± 1.4	**71.77 ± 1.2**	**69.85 ± 2.7**	**67.88 ± 1.7**
	mAP	65.54 ± 1.9	7.02 ± 1.6	56.04 ± 2.7	55.90 ± 2.3	**77.75 ± 0.4**	**76.52 ± 2.2**	**74.18 ± 0.5**
ResNet-101	person	73.27 ± 3.8	5.78 ± 1.8	66.94 ± 2.6	67.62 ± 2.4	**86.26 ± 0.9**	**85.93 ± 1.0**	**79.49 ± 2.2**
	car	59.92 ± 2.6	5.37 ± 2.2	52.13 ± 3.6	52.35 ± 2.7	**71.80 ± 1.0**	**70.44 ± 2.1**	**68.84 ± 2.4**
	mAP	66.60 ± 2.6	5.57 ± 1.8	59.54 ± 2.9	59.99 ± 1.9	**79.03 ± 0.9**	**78.19 ± 1.1**	**74.17 ± 1.7**
ResNet-152	person	78.73 ± 1.9	17.34 ± 5.9	67.44 ± 4.1	70.66 ± 3.7	**89.61 ± 1.4**	**89.16 ± 1.1**	**84.73 ± 1.6**
	car	64.53 ± 1.4	3.94 ± 1.5	52.22 ± 4.5	52.93 ± 2.3	**72.02 ± 2.0**	**70.87 ± 1.4**	**73.68 ± 0.7**
	mAP	71.63 ± 0.4	10.64 ± 3.6	59.83 ± 4.0	61.80 ± 2.7	**80.82 ± 1.2**	**80.02 ± 1.1**	**79.21 ± 0.7**

Table 6.1: Comparison of the detection on the different data formats. The detection scores of the RGB color space are in blue, the ones that overcome it are in bold and the best detection score is in green.

scheme between modalities I and M. Note that a negative error rate is associated to an increase of AP_o^{I+M} with regards to AP_o^I and a positive error rate is associated to a decrease of AP_o^{I+M} with regards to AP_o^I.

Bounding boxes registration

Since the best results on the different data formats are obtained using RetinaNet with a ResNet-152 backbone, the influence of the fusion schemes on the detection scores is evaluated accordingly. All the results, including the ones using the other backbones, are shown in Table 6.2. As it can be seen in this table, the best results are provided by the late fusion scheme using an OR filter and a Double soft-NMS filter. Compared to classical filters (NMS and soft-NMS), the Double soft-NMS filter reaches a 10% increase of the mAP. It also enables a 13.8% increase of the mAP when fusing I an S which leads to a 18.3% increase of the car detection and a 1.9% increase for person detection. Regarding the OR filter, an 3.2% increase in the mAP can be noticed when fusing I and S which leads to a 3.9% increase for car detection and a 1.3% increase regarding the person detection. The early fusion between I and color images does not overcome the reference scores, which is verified for the three backbones. As it is expected, the offset between color and polarimetric images probably penalizes the training process. However, even the fusion between polarimetric features (I and S) does not overcome the reference scores, even though they are stackable pixelwise. Consequently, early fusion is not adapted when it comes to detect object in road scenes under fog. As for the AND filter used in the late fusion scheme, it is not adapted for this purpose. The reduction of the false negatives aimed by this filter does not overcome the reduction of the true positives. Regarding the different color spaces, we can notice that HSV, CIE Lab and YCrCb are more adapted than RGB to be fused with polarimetric features, especially HSV. This might be due to the fact that images in these color spaces provide less false positive detections than images in the RGB color space.

Fusion	Backbone	Class	I	$I+S$	ER^{I+S}	I+ RGB	ER^{I+RGB}	I+ HSV	ER^{I+HSV}	I+ CIE Lab	ER^{I+LAB}	I+ YCrCb	$ER^{I+YCrCb}$
1	ResNet-50	person	**83.73 ± 1.1**	81.03 ± 1.3	16.6	73.86 ± 2.2	60.6	70.35 ± 1.9	82.2	78.44 ± 1.8	32.5	80.38 ± 1.5	2.6
		car	**71.77 ± 1.2**	67.63 ± 1.2	14.7	57.85 ± 1.3	49.3	48.63 ± 1.6	82.0	61.65 ± 1.7	35.8	61.24 ± 1.1	37.3
		mAP	**77.75 ± 0.4**	74.33 ± 0.7	15.4	65.85 ± 1.1	53.5	59.49 ± 1.5	82.1	70.04 ± 0.8	34.7	70.81 ± 1.3	31.2
	ResNet-101	person	**86.26 ± 0.9**	81.83 ± 2.8	32.2	73.53 ± 1.9	92.6	49.89 ± 8.2	264.7	79.87 ± 1.4	46.5	81.94 ± 0.6	31.4
		car	**71.80 ± 1.0**	70.83 ± 2.3	3.4	58.41 ± 0.7	47.5	38.21 ± 4.2	119.1	65.15 ± 1.7	23.6	61.79 ± 1.5	35.5
		mAP	**79.03 ± 0.9**	76.33 ± 0.9	12.9	65.97 ± 0.6	62.3	44.05 ± 5.7	166.8	72.52 ± 0.8	31.0	71.87 ± 0.6	34.1
	ResNet-152	person	**89.61 ± 1.4**	83.49 ± 1.3	58.9	72.77 ± 1.6	162.1	70.62 ± 2.0	182.8	81.51 ± 1.5	80.0	80.93 ± 1.8	83.5
		car	**72.02 ± 2.0**	63.79 ± 2.3	29.4	57.16 ± 3.4	53.1	49.70 ± 2.7	79.8	58.22 ± 0.4	49.3	59.04 ± 2.3	46.4
		mAP	**80.82 ± 1.2**	73.64 ± 1.3	37.4	64.96 ± 2.3	82.7	60.16 ± 2.1	107.7	69.86 ± 0.8	57.1	69.98 ± 1.4	56.5
2	ResNet-50	person	**83.73 ± 1.1**	83.46 ± 1.7	1.7	77.50 ± 3.2	38.3	83.68 ± 1.1	0.3	81.43 ± 2.8	14.1	80.46 ± 3.2	20.1
		car	**71.77 ± 1.2**	71.75 ± 0.6	0.1	70.13 ± 1.0	5.8	71.19 ± 1.2	2.1	69.85 ± 1.2	6.8	69.89 ± 1.4	6.7
		mAP	**77.75 ± 0.4**	77.61 ± 1.0	0.6	73.81 ± 1.4	17.7	77.43 ± 0.3	1.4	75.64 ± 1.3	9.5	75.18 ± 1.3	11.6
	ResNet-101	person	**86.26 ± 0.9**	84.23 ± 1.3	14.8	75.59 ± 4.0	77.7	86.24 ± 1.0	0.1	78.01 ± 3.4	60.0	77.04 ± 3.8	67.1
		car	**71.80 ± 1.0**	71.59 ± 1.2	0.7	66.84 ± 2.3	17.6	71.44 ± 1.1	1.3	68.90 ± 1.9	10.3	68.92 ± 1.9	10.2
		mAP	**79.03 ± 0.9**	77.91 ± 0.9	5.3	71.21 ± 3.0	37.3	78.84 ± 0.9	0.9	73.46 ± 2.3	26.6	72.98 ± 2.3	28.3
	ResNet-152	person	**89.61 ± 1.4**	89.00 ± 1.7	5.9	80.61 ± 1.6	86.6	89.30 ± 1.4	3.0	79.29 ± 4.0	99.3	81.95 ± 2.1	73.7
		car	72.02 ± 2.0	**74.02 ± 1.2**	-7.1	67.65 ± 1.4	15.6	71.43 ± 2.0	2.1	66.51 ± 2.0	19.7	67.22 ± 2.7	17.2
		mAP	80.82 ± 1.2	**81.51 ± 0.9**	-3.6	74.13 ± 1.2	34.9	80.37 ± 1.2	2.3	72.90 ± 2.1	41.3	74.59 ± 2.3	32.3
3	ResNet-50	person	**83.73 ± 1.1**	83.53 ± 1.7	1.2	77.53 ± 3.2	38.1	83.71 ± 1.0	0.1	81.50 ± 2.8	13.7	80.51 ± 3.2	19.8
		car	71.77 ± 1.2	**72.30 ± 0.5**	-1.9	70.27 ± 0.9	5.3	71.39 ± 1.2	1.3	70.02 ± 1.2	6.2	70.05 ± 1.4	6.1
		mAP	77.75 ± 0.4	**77.92 ± 1.0**	-0.8	73.90 ± 1.4	17.3	77.55 ± 0.3	0.9	75.76 ± 1.3	8.9	75.28 ± 1.2	11.1
	ResNet-101	person	86.26 ± 0.9	83.91 ± 1.3	17.1	75.72 ± 4.0	76.7	**86.29 ± 0.9**	-0.2	78.02 ± 3.4	60.0	77.10 ± 3.8	66.7
		car	71.80 ± 1.0	**72.09 ± 0.9**	-1.0	67.03 ± 2.2	16.9	71.66 ± 1.1	0.5	69.03 ± 1.9	9.8	69.09 ± 1.9	9.6
		mAP	**79.03 ± 0.9**	78.00 ± 0.8	4.9	71.38 ± 2.9	36.5	78.98 ± 0.9	0.2	73.53 ± 2.3	26.2	73.09 ± 2.3	28.3
	ResNet-152	person	**89.61 ± 1.4**	89.01 ± 1.7	5.8	81.07 ± 1.6	82.2	89.33 ± 1.4	2.7	79.31 ± 4.1	99.1	81.97 ± 2.1	73.5
		car	72.02 ± 2.0	**74.17 ± 1.2**	-7.7	68.52 ± 1.7	12.5	71.53 ± 2.0	1.8	66.64 ± 2.0	19.2	67.28 ± 2.8	16.9
		mAP	80.82 ± 1.2	**81.59 ± 0.9**	-4.0	74.80 ± 1.2	31.4	80.43 ± 1.2	2.0	72.98 ± 2.1	40.9	74.63 ± 2.3	32.3
4	ResNet-50	person	83.73 ± 1.1	**84.42 ± 1.7**	-4.2	77.29 ± 3.8	39.5	84.23 ± 1.0	-3.1	81.99 ± 3.0	10.7	80.81 ± 3.9	17.9
		car	71.77 ± 1.2	**74.83 ± 0.9**	-10.8	73.11 ± 0.9	-4.7	73.63 ± 1.0	-6.6	72.94 ± 1.0	-4.1	72.79 ± 0.9	-3.6
		mAP	77.75 ± 0.4	**79.63 ± 1.2**	-8.4	75.20 ± 1.8	11.5	78.93 ± 0.4	-5.3	77.47 ± 1.6	1.3	76.80 ± 1.7	4.3
	ResNet-101	person	86.26 ± 0.9	84.45 ± 1.5	13.2	74.84 ± 4.2	83.1	**86.29 ± 1.0**	-0.2	76.14 ± 3.6	73.7	74.38 ± 5.4	86.5
		car	71.80 ± 1.0	**75.13 ± 1.2**	-11.8	70.43 ± 2.1	4.9	73.63 ± 1.0	-6.5	72.00 ± 1.6	-0.7	73.15 ± 0.7	-4.8
		mAP	79.03 ± 0.9	79.79 ± 1.1	-3.6	72.63 ± 2.7	30.5	**79.96 ± 0.7**	-4.4	74.07 ± 2.3	23.7	73.77 ± 2.6	25.1
	ResNet-152	person	89.61 ± 1.4	**89.81 ± 1.7**	-1.9	79.48 ± 1.8	97.5	86.46 ± 6.4	30.3	76.69 ± 5.6	124.4	80.77 ± 2.7	85.1
		car	72.02 ± 2.0	**77.13 ± 1.1**	-18.3	69.53 ± 1.7	8.9	76.33 ± 8.3	-15.4	69.20 ± 1.7	10.1	70.40 ± 2.3	5.8
		mAP	80.82 ± 1.2	**83.47 ± 0.9**	-13.8	74.50 ± 1.6	33.0	81.39 ± 1.1	-3.0	72.94 ± 3.0	41.1	75.58 ± 2.4	27.3
5	ResNet-50	person	83.73 ± 1.1	**84.16 ± 0.8**	-2.6	84.11 ± 0.9	-2.3	84.03 ± 0.9	-1.8	84.05 ± 0.9	-2.0	84.02 ± 0.9	-1.8
		car	71.77 ± 1.2	**72.88 ± 1.0**	-3.9	72.74 ± 1.2	-3.4	72.71 ± 1.2	-3.3	72.72 ± 1.2	-3.4	72.72 ± 1.2	-3.4
		mAP	77.75 ± 0.4	**78.52 ± 0.3**	-3.5	78.43 ± 0.5	-3.1	78.37 ± 0.5	-2.8	78.38 ± 0.4	-2.8	78.37 ± 0.4	2.8
	ResNet-101	person	86.26 ± 0.9	86.36 ± 1.2	-0.7	86.25 ± 0.8	0.1	**86.49 ± 0.9**	-1.7	86.47 ± 0.8	-1.5	86.19 ± 0.8	0.5
		car	71.80 ± 1.0	**72.86 ± 1.1**	-3.8	72.73 ± 1.0	-3.3	72.70 ± 1.0	-3.2	72.76 ± 1.0	-3.4	72.77 ± 1.0	-3.4
		mAP	79.03 ± 0.9	**79.61 ± 0.9**	-2.8	79.49 ± 0.8	-2.2	79.60 ± 0.8	-2.7	79.62 ± 0.8	-2.8	79.48 ± 0.8	-2.1
	ResNet-152	person	89.61 ± 1.4	**89.75 ± 1.4**	-1.3	89.72 ± 1.4	-1.1	89.72 ± 1.4	-1.1	89.64 ± 1.3	-0.3	89.73 ± 1.4	-1.2
		car	72.02 ± 2.0	**73.10 ± 2.2**	-3.9	72.75 ± 1.9	-2.6	72.75 ± 1.9	-2.6	72.71 ± 2.0	-2.5	72.74 ± 2.0	-2.6
		mAP	80.82 ± 1.2	**81.43 ± 1.1**	-3.2	81.24 ± 1.1	-2.2	81.27 ± 1.1	-2.3	81.17 ± 1.1	-1.8	81.23 ± 1.1	-2.1
6	ResNet-50	person	**83.73 ± 1.1**	82.91 ± 1.3	5.0	70.29 ± 3.9	82.6	20.74 ± 4.9	387.1	72.59 ± 2.0	68.4	72.05 ± 2.5	71.8
		car	**71.77 ± 1.2**	71.60 ± 1.2	0.6	56.43 ± 1.6	54.3	10.52 ± 1.4	216.9	51.72 ± 1.4	71.0	50.00 ± 1.5	77.1
		mAP	**77.75 ± 0.4**	77.26 ± 1.1	2.2	63.36 ± 2.6	64.7	15.63 ± 3.1	279.2	62.16 ± 1.1	70.1	61.03 ± 1.8	75.1
	ResNet-101	person	**86.26 ± 0.9**	80.60 ± 2.0	41.2	67.51 ± 4.0	136.5	9.34 ± 3.3	559.8	67.38 ± 3.6	137.4	65.67 ± 6.2	149.9
		car	**71.80 ± 1.0**	70.24 ± 1.2	5.5	55.91 ± 2.4	56.3	12.00 ± 4.4	212.1	51.93 ± 3.8	70.5	53.56 ± 1.9	64.7
		mAP	**79.03 ± 0.9**	75.42 ± 1.2	17.2	61.71 ± 2.7	83.0	10.67 ± 3.5	326.0	59.66 ± 2.5	92.4	59.62 ± 3.8	92.6
	ResNet-152	person	**89.61 ± 1.4**	86.10 ± 1.2	33.8	71.08 ± 2.9	178.3	25.83 ± 8.8	613.9	66.95 ± 6.6	218.1	70.18 ± 3.9	187.0
		car	72.02 ± 2.0	**72.35 ± 1.0**	-1.2	57.02 ± 1.2	53.6	8.18 ± 3.2	228.2	51.13 ± 2.6	74.7	51.09 ± 2.5	74.8
		mAP	**80.82 ± 1.2**	79.23 ± 0.8	8.3	64.05 ± 1.8	87.4	17.00 ± 5.9	333.7	59.04 ± 3.2	113.6	60.64 ± 2.8	105.2

Table 6.2: Comparison of the detection scores with the different fusion schemes with bounding boxes registration. I reaches the best detection scores on the individual formats and is used as a reference (blue). From top to bottom: Early Fusion (1), Naive NMS (2), Naive soft-NMS (3), Double soft-NMS (4), OR filter (5), AND filter (6). The scores that overcome the reference scores are in green and the best detection score is in bold.

Image registration

Registering the color images towards the polarimetric ones lead to improvements on the detection performances. The evaluation of the different fusion schemes using registered images are shown in Table 6.3. Providing multimodal images, stackable pixelwise, indeed improves the performances regarding the early fusion schemes. Again, these improvement still do not overcome the detection on the intensities images without fusion. The early fusion scheme is therefore not adapted for road object detection in foggy scenes. The Double soft-NMS filter also shows great improvements regarding color and polarimetric fusion. The mAP reached by the intensities and RGB images fusion reaches 74.50% when registering the bounding boxes whereas it reaches 80.22% when registering images. The mAP achieved when fusing the intensities and HSV images is of 81.39% when registering the bounding boxes whereas it is of 81.47% when registering images. As for the intensities and CIE Lab images fusion, the mAP is of 72.94% when registering the bounding boxes whereas it reaches 77.53% when registering images. Finally, the YCrCb and intensities images fusion reaches 75.58% when registering bounding boxes and it is of 79.27% when registering images.

Overall discussion

To summarize the obtained results, the fusion schemes enabling the best results are the late fusion scheme with the Double soft-NMS filter and the OR filter. The Double soft-NMS enables to provide the best results when combining the most relevant data format for road object detection under fog which are the intensities and the Stokes images. The OR filter, however, reaches slightly lower results than the Double soft-NMS filter but it is more robust to the data combination, since it reaches at least the performances of the control data format. The modalities combinations that provide the best results are the intensities images combined with the Stokes images and the intensities images combined with the HSV images. Regarding the registration processes, the image registration provides better results than the bounding boxes registration.

To conclude, even if these fusion schemes enable to both generalize the features learnt in good weather conditions to adverse scenes and improve the detections results under fog, some improvements could be made to fuse more efficiently color and polarimetric features. One can imagine an optimized fusion between color and polarimetric features as a deep fusion scheme. In this way, these complementary features could be associated in an optimal way during the training process and thus improve the detection results under fog. Nevertheless, the results achieved so far are encouraging especially to improve road object detection under fog. They also enable to find color spaces which are more adapted than RGB for multimodal fusion with polarimetric images.

Fusion	Backbone	Class	I	$I+S$	ER^{I+S}	I+ RGB	ER^{I+RGB}	I+ HSV	ER^{I+HSV}	I+ CIE Lab	ER^{I+LAB}	I+ YCrCb	$ER^{I+YCrCb}$
1	ResNet-50	person	**83.73 ± 1.1**	81.03 ± 1.3	16.6	77.80 ± 2.5	36.4	52.46 ± 3.4	192.2	79.50 ± 1.5	26.0	78.10 ± 1.6	34.6
		car	**71.77 ± 1.2**	67.63 ± 1.2	14.7	65.00 ± 2.3	24.0	44.99 ± 1.6	94.8	66.32 ± 1.8	19.3	65.75 ± 1.0	21.3
		mAP	**77.75 ± 0.4**	74.33 ± 0.7	15.3	71.40 ± 1.9	28.5	48.72 ± 2.3	130.5	72.91 ± 1.6	21.8	71.93 ± 1.1	26.2
	ResNet-101	person	**86.26 ± 0.9**	81.83 ± 2.8	32.24	80.22 ± 5.0	44.0	30.21 ± 2.7	407.9	82.26 ± 1.7	29.1	81.55 ± 1.8	34.3
		car	**71.80 ± 1.0**	70.83 ± 2.3	3.4	63.95 ± 2.1	27.8	39.57 ± 4.0	114.3	62.29 ± 1.9	33.7	63.43 ± 1.6	29.7
		mAP	**79.03 ± 0.9**	76.33 ± 0.9	12.9	72.09 ± 2.9	33.1	34.89 ± 3.2	210.5	72.28 ± 1.7	32.2	72.49 ± 1.6	31.2
	ResNet-152	person	**89.61 ± 1.4**	83.49 ± 1.3	58.9	84.04 ± 1.2	53.6	66.72 ± 4.6	220.3	83.54 ± 0.7	58.4	79.01 ± 3.0	102.0
		car	**72.02 ± 2.0**	63.79 ± 2.3	29.4	60.92 ± 1.5	39.7	51.31 ± 2.4	74.0	60.93 ± 3.5	39.6	60.44 ± 1.6	41.4
		mAP	**80.82 ± 1.2**	73.64 ± 1.3	37.4	72.48 ± 1.1	43.5	59.02 ± 3.4	113.7	72.23 ± 2.0	44.8	69.72 ± 1.5	57.9
2	ResNet-50	person	**83.73 ± 1.1**	83.46 ± 1.7	1.7	81.50 ± 2.2	13.7	83.72 ± 1.0	0.1	82.58 ± 1.9	7.1	83.57 ± 1.0	1.0
		car	**71.77 ± 1.2**	71.75 ± 0.6	0.1	71.16 ± 1.5	2.2	70.94 ± 1.3	2.9	70.35 ± 1.2	5.0	70.97 ± 1.4	2.8
		mAP	**77.75 ± 0.4**	77.61 ± 1.0	0.6	76.33 ± 0.5	6.4	77.33 ± 0.3	1.9	76.47 ± 0.8	5.8	77.27 ± 0.4	2.2
	ResNet-101	person	**86.26 ± 0.9**	84.23 ± 1.3	14.8	82.12 ± 3.0	30.1	86.25 ± 1.0	0.1	83.22 ± 1.9	22.1	84.38 ± 2.5	13.7
		car	**71.80 ± 1.0**	71.59 ± 1.2	0.7	69.53 ± 1.5	8.0	71.29 ± 1.2	1.8	68.99 ± 2.1	10.0	70.12 ± 1.5	6.0
		mAP	**79.03 ± 0.9**	77.91 ± 0.9	5.3	75.83 ± 2.2	15.3	78.77 ± 1.0	1.2	76.11 ± 1.7	13.9	77.25 ± 1.2	8.5
	ResNet-152	person	**89.61 ± 1.4**	89.00 ± 1.7	5.9	87.55 ± 0.4	198.3	89.57 ± 1.5	0.4	83.18 ± 3.6	61.9	85.78 ± 2.6	36.9
		car	72.02 ± 2.0	**74.02 ± 1.2**	-7.1	69.56 ± 1.6	8.8	71.17 ± 2.1	3.3	69.41 ± 2.4	9.3	69.66 ± 2.4	8.4
		mAP	80.82 ± 1.2	**81.51 ± 0.9**	-3.6	78.56 ± 0.9	11.8	80.37 ± 1.2	2.3	76.30 ± 2.1	23.6	77.72 ± 2.1	16.2
3	ResNet-50	person	83.73 ± 1.1	83.53 ± 1.7	1.2	81.52 ± 2.2	13.6	**83.74 ± 1.0**	-0.1	82.60 ± 1.9	6.9	83.60 ± 1.0	0.8
		car	71.77 ± 1.2	**72.30 ± 0.5**	-1.9	71.13 ± 1.7	2.3	71.13 ± 1.2	2.3	70.56 ± 1.2	4.3	71.17 ± 1.4	2.1
		mAP	77.75 ± 0.4	**77.92 ± 1.0**	-0.7	76.33 ± 0.6	6.4	77.44 ± 0.3	1.4	76.58 ± 0.8	5.3	77.39 ± 0.4	1.6
	ResNet-101	person	86.26 ± 0.9	83.91 ± 1.3	17.1	81.65 ± 3.2	33.6	**86.31 ± 0.9**	-0.4	83.28 ± 1.9	21.7	84.39 ± 2.5	13.6
		car	71.80 ± 1.0	72.09 ± 0.9	-1.0	**72.64 ± 1.1**	-3.0	71.52 ± 1.2	1.0	69.20 ± 2.1	9.2	70.28 ± 1.5	5.4
		mAP	**79.03 ± 0.9**	78.00 ± 0.8	4.9	77.15 ± 2.1	9.0	78.92 ± 1.0	0.5	76.24 ± 1.7	13.3	77.34 ± 1.2	6.2
	ResNet-152	person	**89.61 ± 1.4**	89.01 ± 1.7	5.8	87.62 ± 0.5	19.2	89.59 ± 1.5	0.2	83.19 ± 3.6	61.8	85.77 ± 2.6	37.0
		car	72.02 ± 2.0	**74.17 ± 1.2**	-7.7	69.71 ± 1.7	8.3	71.28 ± 2.2	2.6	69.5 ± 2.5	9.0	69.79 ± 2.5	8.0
		mAP	80.82 ± 1.2	**81.59 ± 0.9**	-4.0	78.66 ± 0.9	11.3	80.43 ± 1.2	2.0	76.35 ± 2.1	23.3	77.78 ± 2.2	15.8
4	ResNet-50	person	83.73 ± 1.1	**84.42 ± 1.7**	-4.2	82.00 ± 2.3	10.6	84.37 ± 0.9	-3.9	83.22 ± 1.8	3.1	84.23 ± 1.0	-3.1
		car	71.77 ± 1.2	**74.83 ± 0.9**	-10.8	73.81 ± 1.0	-7.2	73.40 ± 1.1	-5.8	73.39 ± 0.9	-5.7	73.52 ± 1.1	-6.2
		mAP	77.75 ± 0.4	**79.63 ± 1.2**	-8.4	77.91 ± 0.9	-0.7	78.89 ± 0.3	-5.1	78.31 ± 0.7	-2.5	78.87 ± 0.4	-5.0
	ResNet-101	person	86.26 ± 0.9	84.45 ± 1.5	13.1	81.65 ± 3.2	33.6	**86.33 ± 1.0**	-0.5	83.10 ± 1.8	23.0	84.15 ± 2.8	15.4
		car	71.80 ± 1.0	**75.13 ± 1.2**	-11.8	72.64 ± 1.1	-3.0	71.52 ± 1.2	1.0	72.70 ± 1.4	-3.2	73.30 ± 1.1	-5.3
		mAP	79.03 ± 0.9	**79.79 ± 1.1**	-3.6	77.15 ± 2.1	9.0	78.92 ± 1.0	0.5	77.90 ± 1.3	5.4	78.73 ± 1.3	1.4
	ResNet-152	person	89.61 ± 1.4	**89.81 ± 1.7**	-1.9	87.78 ± 0.5	17.6	89.77 ± 1.6	-1.5	82.12 ± 4.3	72.1	85.34 ± 3.2	41.1
		car	72.02 ± 2.0	**77.13 ± 1.1**	-18.3	72.66 ± 1.2	-2.3	73.17 ± 1.6	-4.1	72.95 ± 1.6	-3.3	73.20 ± 1.6	-4.2
		mAP	80.82 ± 1.2	**83.47 ± 0.9**	-13.8	80.22 ± 0.7	3.1	81.47 ± 1.1	-3.4	77.53 ± 2.3	17.2	79.27 ± 2.2	8.1
5	ResNet-50	person	83.73 ± 1.1	**84.16 ± 0.8**	-2.6	84.07 ± 0.9	-2.1	84.03 ± 0.9	-1.8	84.03 ± 0.9	-1.8	84.03 ± 0.9	-1.8
		car	71.77 ± 1.2	**72.88 ± 1.0**	-3.9	72.72 ± 1.2	-3.4	72.71 ± 1.2	-3.3	72.71 ± 1.2	-3.3	72.70 ± 1.2	-3.3
		mAP	77.75 ± 0.4	**78.52 ± 0.3**	-3.5	78.40 ± 0.5	-2.9	78.37 ± 0.5	-2.8	78.37 ± 0.5	-2.8	78.37 ± 0.5	-2.8
	ResNet-101	person	86.26 ± 0.9	86.36 ± 1.2	-0.7	85.87 ± 0.6	2.8	**86.49 ± 0.9**	-1.7	86.17 ± 0.6	0.7	86.28 ± 1.1	-0.1
		car	71.80 ± 1.0	**72.86 ± 1.1**	-3.8	72.73 ± 1.0	-3.3	72.70 ± 1.0	-3.2	72.71 ± 1.0	-3.2	72.70 ± 1.0	-3.2
		mAP	79.03 ± 0.9	**79.61 ± 0.9**	-2.8	79.30 ± 0.6	-1.3	79.60 ± 0.8	-2.7	79.44 ± 0.8	-2.0	79.49 ± 0.8	-2.2
	ResNet-152	person	89.61 ± 1.4	**89.75 ± 1.4**	-1.3	89.58 ± 1.3	0.3	89.72 ± 1.4	-1.1	89.65 ± 1.3	-0.3	89.58 ± 1.4	0.3
		car	72.02 ± 2.0	**73.10 ± 2.2**	-3.9	72.87 ± 1.9	-3.0	72.81 ± 2.0	-2.8	72.81 ± 2.0	-2.8	72.81 ± 2.0	-2.8
		mAP	80.82 ± 1.2	**81.43 ± 1.1**	-3.2	81.23 ± 1.1	-2.1	81.26 ± 1.1	-2.3	81.23 ± 1.2	-2.1	81.20 ± 1.2	-2.0
6	ResNet-50	person	**83.73 ± 1.1**	82.91 ± 1.3	5.0	77.39 ± 1.7	39.0	8.87 ± 2.0	460.1	70.14 ± 4.6	83.5	15.86 ± 8.33	417.1
		car	**71.77 ± 1.2**	71.60 ± 1.2	0.6	55.85 ± 5.0	56.4	13.67 ± 1.6	205.8	48.47 ± 2.5	82.5	24.06 ± 6.8	169.0
		mAP	**77.75 ± 0.4**	77.26 ± 1.1	2.2	66.62 ± 3.1	50.0	11.27 ± 1.5	298.8	59.31 ± 3.4	82.9	19.96 ± 7.3	259.7
	ResNet-101	person	**86.26 ± 0.9**	80.60 ± 2.0	41.2	73.45 ± 5.2	93.2	2.3 ± 0.9	611.1	69.95 ± 3.5	118.7	64.36 ± 11.9	159.4
		car	**71.80 ± 1.0**	70.24 ± 1.2	5.5	54.23 ± 2.6	62.3	6.74 ± 1.1	230.7	39.73 ± 5.2	113.7	40.62 ± 8.2	110.5
		mAP	**79.03 ± 0.9**	75.42 ± 1.2	17.2	63.84 ± 3.1	72.4	4.5 ± 0.7	355.4	54.84 ± 3.9	115.4	52.49 ± 9.6	126.6
	ResNet-152	person	**89.61 ± 1.4**	86.10 ± 1.2	33.8	82.94 ± 1.7	64.2	4.29 ± 2.4	821.2	75.02 ± 5.8	140.4	75.73 ± 5.9	133.6
		car	72.02 ± 2.0	**72.35 ± 1.0**	-1.2	56.61 ± 2.1	55.1	9.19 ± 2.0	224.6	48.78 ± 5.7	83.1	47.41 ± 4.5	88.0
		mAP	**80.82 ± 1.2**	79.23 ± 0.8	8.3	69.78 ± 1.5	57.6	6.74 ± 2.0	386.2	61.90 ± 5.2	98.6	61.57 ± 4.6	100.4

Table 6.3: Comparison of the detection scores with the different fusion schemes with image registration. I reaches the best detection scores on the individual formats and is used as a reference (blue). From top to bottom: Early Fusion (1), Naive NMS (2), Naive soft-NMS (3), Double soft-NMS (4), OR filter (5), AND filter (6). The scores that overcome the reference scores are in green and the best detection score is in bold.

6.3 Validation on more diverse adverse situations

In section 6.2, different fusion schemes are explored to enhance road object detection in foggy scenes. It is shown that using a late fusion scheme with a Double soft-NMS or and OR filter improves detection scores, especially when fusing intensities and Stokes images. However, even if this pipeline can be applied to other adverse weather conditions, its behavior in these situations is unknown.

This section focuses on the study of polarimetric and color-based features in a wide range of adverse weather situations. The performances of intensities images, Stokes images, RGB images and their fusion in eleven different weather situations, including different densities of fog and tropical rain, are evaluated. The experimental results enable to extend the conclusions drawn previously to other weather conditions. They also show the limits of this pipeline in adverse weather conditions with very low visibility.

6.3.1 Experimental setup

In order to study the behavior of the polarimetric features in several weather conditions, the following experiments are carried out. The first goal of this experiment is to evaluate how invariant are the polarimetric features characterizing road objects to the visibility conditions. The second goal of the experiment is to evaluate the relevance of multimodal fusion to enhance road object detection in several adverse weather situations. The polarimetric features selected for this experiment are the intensities images $I = (I_0, I_{45}, I_{90})$ and the Stokes images $S = (S_0, S_1, S_2)$ described in [266]. The color-based features are RGB information.

The training and validation sets used for this experiments are the ones of the Polar-LITIS dataset (see third row of Table 4.3) and contain sunny and cloudy road scenes. The testing set, however is constituted exclusively of eleven adverse weather conditions, including several densities of fog, including 15m, 20m, 25m, 30m, 35m, 40m, 45m, 50m, 60m and 70m of visibility distance, and tropical rain (see sixth row of Table 4.3). Because the adverse scenes are acquired in a tunnel, their variability is limited. Including them into the training process is likely to cause over-fitting towards the testing set and bias the detection results. On top of that, this design enables to evaluate if the road objects features learnt in good weather conditions are still valid to detect objects when the visibility is altered. The RetinaNet network [3], using a ResNet-152 backbone [166] is used for this task. As a matter of fact, this object detector, thanks to the Focal loss, focuses on hard misclassified examples during its training process. This property is useful to process datasets with unbalanced classes, as it is the case in this experiment. It is also able to process images in real time with a high accuracy, which is paramount to perform object detection in road scenes. All the experimental setups are sketched in Figure 6.7.

A late fusion scheme is used to fuse the different modalities used in this experiment. This fusion architecture is based on the results of section 6.2. This work proves that an early fusion scheme is not adapted to analyze road scenes in adverse weather conditions because polarimetric and color-based images are not stackable pixelwise. Based on this work, the Double soft-NMS and the OR filters are used to fuse these modalities since

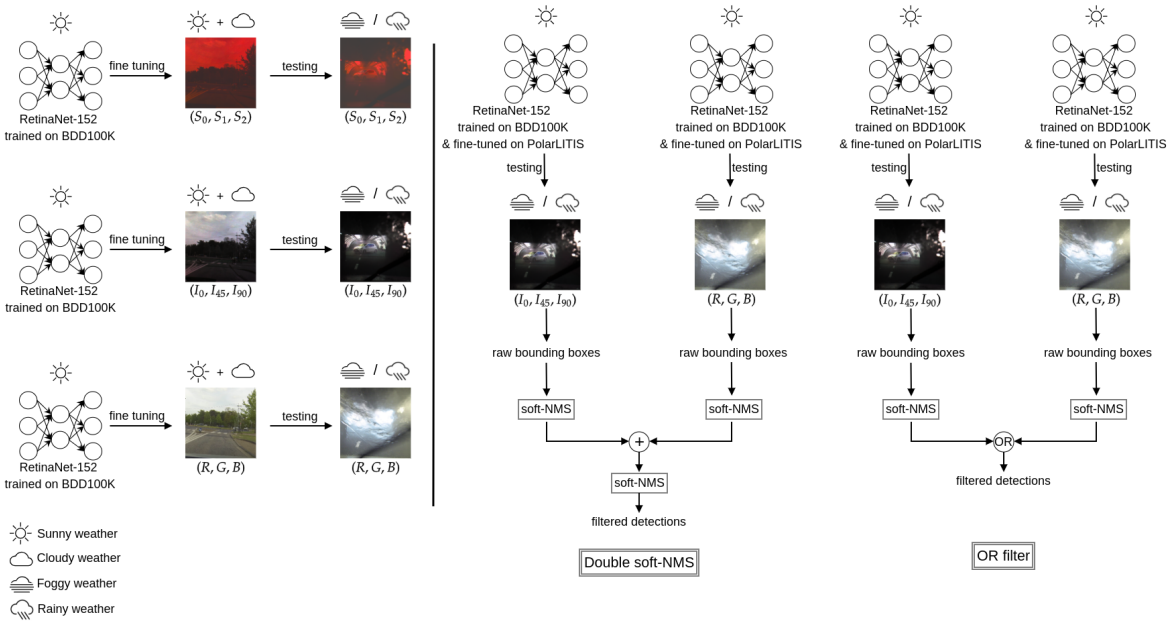

Figure 6.7: Experimental setup. On the left, the training processes on each modality, respectively I, S and RGB is illustrated. On the right, the two fusion schemes (Double soft-NMS filter and OR filter) are illustrated with I and RGB fusion and can be extended to I and S fusion.

they provide the best results. To fuse the polarimetric and color-based images, the offset between these two modalities is computed.

Since the training set is composed of 1640 images and the validation set of 420 images, it is paramount to pre-train the network on a larger dataset. The BDD100K dataset [244] is selected for this task since it is rather large and aims to detect objects in road scenes in good weather conditions. On top of that, it contains all the classes of the PolarLITIS dataset, making fine-tuning towards this dataset easier. Once the architecture pre-trained on BDD100K, it is fine-tuned on PolarLITIS, on each modality separately (I, S and RGB).

Regarding the training hyperparameters, the ones provided by the RetinaNet's article are kept, i.e. a learning rate of 10^{-5} and the Adam optimizer [257]. Each training process is repeated five times to provide reliable results. Note that the different architectures are trained for 50 epochs on BDD100K and for 20 epochs on PolarLITIS. The optimal weights are selected according to the lowest value of the validation loss.

6.3.2 Discussion and results

To evaluate the increase in detection scores, the error rate evolution is computed as follows:

$$ER_o^M = \frac{1 - AP_o^M - (1 - AP_o^I)}{1 - AP_o^I} \times 100 \ , \tag{6.1}$$

where ER_o^M is the error rate evolution between the intensities polarimetric data format

Modality	Class	15m	20m	25m	30m	35m	40m	45m	50m	60m	70m	rain
I	person	49.46 ± 2.6	33.96 ± 5.1	35.63 ± 2.8	48.73 ± 4.6	68.76 ± 3.2	60.61 ± 3.1	71.89 ± 2.3	74.72 ± 1.3	76.49 ± 1.5	71.97 ± 2.1	73.68 ± 2.8
	car	0 ± 0	6.99 ± 1.2	41.65 ± 3.7	69.21 ± 4.3	85.21 ± 5.4	58.85 ± 7.4	75.40 ± 2.7	79.83 ± 8.5	90.97 ± 3.2	86.56 ± 4.6	85.69 ± 7.3
	mAP	24.73 ± 1.3	20.48 ± 2.4	38.64 ± 3.2	58.97 ± 4.1	76.98 ± 2.1	59.73 ± 4.2	73.65 ± 2.2	77.28 ± 4.7	83.73 ± 1.7	79.27 ± 2.6	79.69 ± 4.1
S	person	50.64 ± 5.1	51.58 ± 8.1	44.47 ± 3.4	66.69 ± 2.5	68.88 ± 4.1	68.61 ± 4.2	70.24 ± 2.2	77.29 ± 3.3	68.40 ± 2.2	68.43 ± 2.9	78.26 ± 6.1
	car	0.48 ± 0.6	13.73 ± 1.9	37.59 ± 8.5	47.16 ± 10.2	77.06 ± 7.2	53.51 ± 17.7	63.76 ± 14.4	64.61 ± 15.6	70.35 ± 7.4	70.78 ± 9.7	93.10 ± 4.4
	mAP	25.56 ± 2.7	32.66 ± 3.6	41.03 ± 4.6	56.92 ± 5.6	72.97 ± 4.9	61.06 ± 10.2	67.00 ± 8.2	70.95 ± 9.4	69.37 ± 4.1	69.61 ± 6.0	85.68 ± 4.7
ER^S	person	-2.3	-26.7	-13.7	-35.0	-0.4	-20.3	5.9	-10.2	34.4	12.6	-17.4
	car	-0.5	-7.3	7.0	71.6	55.1	13.0	47.3	75.5	228.3	117.4	-51.8
	mAP	-1.1	-15.3	-3.9	5.0	17.4	-3.3	25.2	27.9	88.3	46.6	-29.5
RGB	person	14.56 ± 4.9	16.27 ± 3.2	✘	✘	18.19 ± 2.5	16.68 ± 1.8	18.80 ± 3.1	12.00 ± 4.3	24.26 ± 4.3	18.22 ± 1.9	21.08 ± 4.3
	car	0.00 ± 0.0	0.00 ± 0.0	✘	✘	0.00 ± 0.0	6.82 ± 0.0	8.89 ± 0.4	1.84 ± 4.1	28.48 ± 0.7	20.90 ± 0.9	14.52 ± 3.5
	mAP	7.28 ± 2.5	8.14 ± 1.6	✘	✘	9.10 ± 1.2	11.75 ± 0.9	13.85 ± 1.7	6.90 ± 3.4	26.37 ± 2.1	19.56 ± 1.2	17.80 ± 4.3
ER^{RGB}	person	69.1	26.8	✘	✘	161.9	11.5	188.9	248.1	222.2	191.8	199.8
	car	0.0	7.5	✘	✘	576.1	126.4	270.4	386.7	692.0	488.5	497.3
	mAP	23.2	15.5	✘	✘	294.9	191.5	226.9	309.8	352.6	288.0	304.7
$I + S$ Double soft-NMS	person	54.14 ± 3.1	53.03 ± 7.4	45.07 ± 3.2	69.23 ± 3.1	72.62 ± 2.7	69.84 ± 4.0	76.93 ± 2.1	78.99 ± 1.9	77.00 ± 1.4	74.51 ± 2.5	83.24 ± 2.8
	car	0.45 ± 0.6	13.60 ± 2.1	46.85 ± 5.3	71.19 ± 4.4	88.85 ± 2.4	61.70 ± 10.5	74.58 ± 4.3	75.40 ± 11.5	90.28 ± 3.8	84.59 ± 5.3	93.35 ± 3.7
	mAP	27.30 ± 1.7	33.31 ± 3.4	45.96 ± 3.3	70.21 ± 3.4	80.74 ± 2.5	65.77 ± 5.5	75.76 ± 3.1	77.19 ± 6.3	83.64 ± 2.3	79.55 ± 3.6	88.29 ± 2.3
ER^{I+S} Double soft-NMS	person	-10.3	-28.9	-14.7	-40.0	-12.4	-23.4	-17.9	-16.9	-2.2	-9.1	-36.3
	car	-0.5	-7.1	-8.9	-6.4	-24.6	-6.9	3.33	22.0	7.6	14.6	-53.5
	mAP	-3.4	-16.1	-11.9	-27.4	-16.3	-15.0	-8.0	0.4	0.6	-1.4	-42.3
$I + S$ OR filter	person	51.05 ± 2.6	41.25 ± 5.5	37.49 ± 3.3	50.25 ± 4.9	69.10 ± 3.0	63.97 ± 1.7	72.30 ± 2.4	75.48 ± 1.8	76.81 ± 1.4	72.72 ± 2.2	73.61 ± 2.9
	car	0.08 ± 0.2	8.34 ± 2.7	40.30 ± 5.6	70.08 ± 3.3	83.42 ± 4.4	59.04 ± 7.1	74.66 ± 3.7	78.84 ± 7.9	90.82 ± 3.3	86.40 ± 4.0	86.64 ± 6.6
	mAP	25.56 ± 1.3	24.79 ± 2.4	38.90 ± 3.5	60.17 ± 3.8	76.26 ± 1.1	61.51 ± 4.0	73.48 ± 2.5	77.16 ± 4.3	83.82 ± 1.6	79.56 ± 2.4	80.13 ± 3.7
ER^{I+S} OR filter	person	-3.1	-11.0	-2.9	-3.0	-1.1	-8.5	-1.5	-3.0	-1.4	-2.7	0.3
	car	-0.1	-8.2	2.3	-2.8	12.1	-0.5	3.0	4.9	1.7	1.2	-6.6
	mAP	-1.1	-5.4	-0.4	-2.9	3.1	-4.4	0.6	0.5	-0.6	-1.4	-2.2
I+RGB Double soft-NMS	person	49.65 ± 2.7	38.06 ± 3.8	✘	✘	69.07 ± 4.5	61.08 ± 3.1	72.94 ± 2.6	75.10 ± 1.5	76.08 ± 3.1	72.83 ± 2.5	72.95 ± 2.7
	car	0.00 ± 0.0	5.49 ± 2.0	✘	✘	83.64 ± 5.2	59.26 ± 6.26	76.80 ± 2.4	78.39 ± 9.7	87.59 ± 3.6	89.95 ± 3.2	77.23 ± 7.5
	mAP	24.83 ± 1.4	21.77 ± 1.5	✘	✘	76.35 ± 2.5	60.17 ± 3.5	74.87 ± 2.2	76.74 ± 5.4	81.84 ± 3.2	81.39 ± 2.6	75.09 ± 3.6
$ER^{I+\text{RGB}}$ Double soft-NMS	person	-0.4	-6.2	✘	✘	-1.0	-1.2	-3.7	-1.5	1.7	-3.1	2.8
	car	0.0	1.6	✘	✘	10.6	-1.0	-5.7	7.1	37.4	-25.2	59.1
	mAP	-0.1	-1.6	✘	✘	2.7	-1.1	-4.6	2.4	11.6	-10.2	22.6
I+RGB OR filter	person	49.58 ± 2.6	31.34 ± 5.8	✘	✘	68.74 ± 3.3	60.26 ± 3.1	71.35 ± 1.9	74.19 ± 2.0	73.89 ± 1.5	72.43 ± 2.2	73.72 ± 2.8
	car	0.00 ± 0.0	6.29 ± 1.9	✘	✘	84.94 ± 5.5	59.10 ± 7.4	74.88 ± 2.8	79.12 ± 9.1	89.89 ± 3.0	86.54 ± 4.6	81.72 ± 7.2
	mAP	24.79 ± 1.3	18.82 ± 3.0	✘	✘	76.84 ± 2.0	59.68 ± 4.2	73.12 ± 1.8	76.65 ± 5.5	81.89 ± 1.8	79.49 ± 2.6	77.72 ± 3.9
$ER^{I+\text{RGB}}$ OR filter	person	-0.2	-4.0	✘	✘	0.1	0.9	1.9	2.1	11.1	-1.6	-0.2
	car	0.0	0.8	✘	✘	18.3	-0.6	2.1	3.5	12.0	0.1	27.7
	mAP	-0.1	2.1	✘	✘	0.6	0.1	2.0	2.8	11.3	-1.1	9.7

Table 6.4: Comparison of the detection scores. The best detection scores for each adverse weather condition are in blue. The crosses (✘) remind that the RGB images of foggy scenes with 25m and 30m visibility are not available for this experiment.

I and the modality $M \in \{S, \text{RGB}, I + S, I\text{+RGB}\}$ for object $o \in \{'\text{person}', '\text{car}', mAP\}$, AP_o^I is the average precision for object o with data format I, while AP_o^M denotes the average precision on the object o with modality M. A negative error rate is associated to an increase of AP_o^M with regards to AP_o^I and a positive error rate is associated to a decrease of AP_o^M with regards to AP_o^I.

The intensities images I are used as a reference to compute the error rate evolution since they provide the best results in sections 6.2 and 5.4.

As PolarLITIS does not contain enough instances of the class bike, it is not taken into account during the evaluation process. Note that there are no instances of the class motorbike in the testing set. It is important to remind that the architectures used for this experiment are exclusively trained on good weather conditions (sunny and cloudy) and tested exclusively on adverse scenes (foggy scenes with different visibilities and rain). This pipeline enables to evaluate how polarimetric and color-based features vary with the visibility conditions. On top of that, since the acquisitions are made into a tunnel, the glare is an additional visibility alteration. The results of the experiments can be found in Table 6.4.

As can be seen in this table, regarding the three data formats, the polarimetric

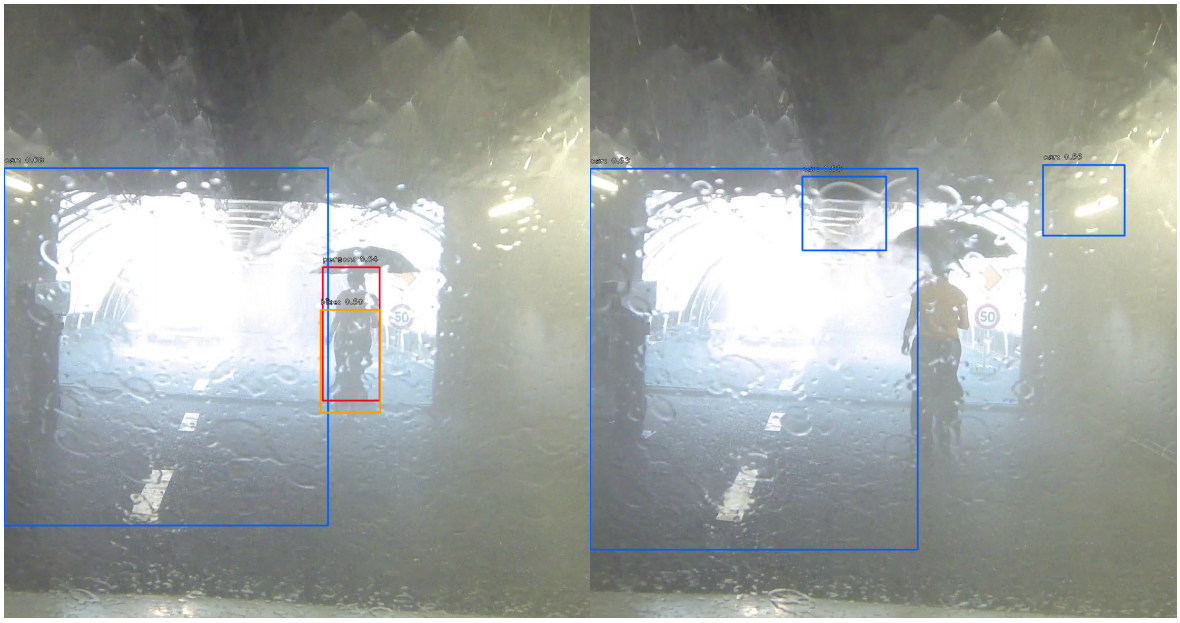

Figure 6.8: Examples of false positives detection in adverse weather conditions. Blue red and orange bounding boxes respectively denote car, person and bike detection.

detection scores overcome the RGB detection scores in every adverse situation. We can also notice that the Stokes images S are more adapted to detect road objects in foggy scenes when the fog visibility is lower than 30m with up to 15% amelioration in the mAP. S are also more adapted to detect objects under tropical rain with a 30% increase in the mAP. When processing scenes under fog with greater visibility distances, the intensities images I are more adapted. These results are also summarized in Figure 6.9 regarding fog detection. Note that even if there is a gap between the detection scores corresponding to 35m and 40m visibility. This gap can be due to a higher number of non-ideal images contained in the class 40m visibility or ideal images contained in the 35m visibility class. Nevertheless, the increasing curve tendency shows enhanced detection scores with a greater visibility distance. These results give a first intuition on the use of fusing I and S in order to improve road object detection in every situation.

Regarding the fusion schemes, when fusing I and S using a late fusion scheme with the Double soft-NMS filter, it leads up to a 27% increase of the mAP for road object detection under fog and to a 42% increase of the mAP for road object detection under tropical rain. The same fusion scheme with an OR filter is less adapted since it takes into account the false positives, which are more numerous in adverse weather conditions as seen in Figure 6.8. It enables a slight amelioration for I and S fusion with up to a 5% amelioration of the mAP for road object detection under fog and a 2% increase of the mAP under tropical rain.

As for the polarimetric and color-based fusion, as mentioned previously, the end-fusion scheme using an OR filter is not adapted to fuse RGB images and I since it takes the false positives into account. The same pipeline using soft-NMS filter applied to these two modalities, however, is not adapted to every situation. They overcome I and S fusion in foggy scenes when the visibility is the greatest, i.e. of 70m. As a

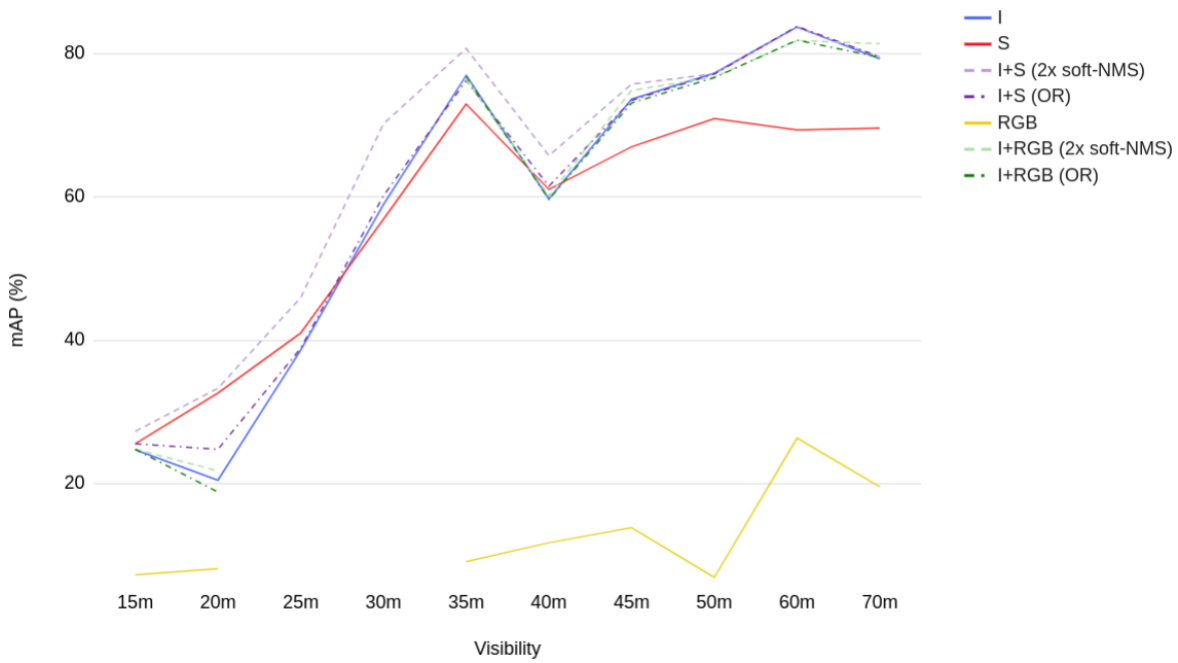

Figure 6.9: Evolution of the mAP in foggy scenes while varying the visibility distance. I, S and the RGB scores are respectively in blue, red and yellow (full lines). The fusion scores of I and S are respectively in pale and dark purple for the Double soft-NMS and the OR filters (dashed lines). The fusion scores of I and RGB are respectively in pale and dark green for the Double soft-NMS and the OR filters (dashed lines).

matter of fact, the mAP in this situation is increased by 10%.

From all these results, we can conclude that when the visibility is very low, as it is the case in very dense fog and tropical rain, S and I fusion provides the best results. The color-based and polarimetric fusion is beneficial in adverse road scenes with a better visibility, such as light fog. However, as can be seen in Figure 6.9, when the visibility is lower than 30m, the polarimetric features learnt in good weather conditions are not able to detect efficiently all road objects in adverse scenes. This limitation could be overcome by including adverse situations in the training process. Despite this limitation, the experimental results show that under tropical rain and under fog from 30m visibility, polarimetric features are a real added value to enhance road objects detection. On top of that, as it can be seen in Figure 4.5, polarimetric features are more robust to the glare and to drops or veils of water on the windshield, causing deformation and loss of information in color-based images. Overall, polarimetric features are more adapted than color-based features to characterize objects in unexpected visibility alterations, as it is illustrated in Figure 6.10.

Figure 6.10: Detection results in several adverse weather conditions. From top to bottom: tropical rain and fog with respectively 35m and 60m visibility. From left to right: I, S, RGB, $I + S$ (Double soft-NMS), $I + S$ (OR filter), I+RGB (Double soft-NMS) and I+RGB (OR filter). Bounding boxes in green, blue, red and orange denote respectively the ground truth, car, person and bike detection.

6.4 Summary

In this chapter, six fusion schemes are studied to enhance road object detection under fog as a first step. Several color-spaces are explored for multimodal polarimetric and color-based fusion. It is demonstrated that fusing polarimetric intensities and Stokes images, using a late fusion scheme with a Double soft-NMS filter, provides the best results for road scene analysis in adverse weather. Fusing HSV with polarimetric intensities images is the most adapted combination for multimodal polarimetric and color-based fusion. The OR filter is also an added value since it increases the true positive rate. Finally, a CycleGAN is used for image registration with an inconsistent offset. The registered images increase the detection results when fusing the polarimetric and color-based features, since the fused images are stackable pixelwise.

A final experiment is carried out to study the behavior of polarimetric features in a wide range of adverse weather conditions. It is showed that once again, fusing polarimetric intensities and Stokes images, using a deep architecture with a Double soft-NMS filter, provides the best results in several densities of fog and under tropical rain. Polarimetric and color-based fusion on the other hand are an added value when the fog's visibility is greater. However, this pipeline shows limits when the visibility is very low, such as fog with a visibility distance under 30m. Even though polarimetric features are more invariant than color-based ones to the visibility alterations, some adverse situations should be included in the training process to deploy this pipeline at a large scale.

Conclusion and perspectives

Conclusion

This thesis aimed at enhancing road scene analysis in adverse weather conditions, fusing polarimetric features and color-based images. In Chapter 1, we introduced the concept of multimodality. The polarization formalism is first detailed, followed by color models and other non-conventional modalities such as infrared imaging or LiDAR point clouds. Chapter 2 gives the background knowledge on deep learning, especially on the object detection paradigm but also an overview on CycleGAN. The literature of non-conventional modalities, as well as object detectors is reviewed in Chapter 3, showing how they fall within autonomous systems' pipelines to enhance road scene analysis in complex situations. Chapter 4 presents the datasets constituted to perform road scene analysis in adverse weather conditions using polarimetric features, as well as the polarimetric data formats encoded for machine learning. It also describes the adapted CycleGAN designed to generate polarimetric images under physical constraints from RGB ones, enabling to have a polarimetric equivalent of the road object detection benchmarks. A demonstration of the invariance of polarimetric features to the visibility conditions is made in Chapter 5, by comparing their detection performances with the color-based ones in foggy road scenes. Finally, these results are enhanced and extended to other adverse weather conditions in Chapter 6, by designing an adapted color-based and polarimetric fusion scheme.

In more details, in this thesis we came up with the first large polarimetric and color-based dataset, labelled for road object detection in several weather conditions. It contains both road scenes acquired in real conditions and road scenes acquired in a tunnel simulating fog and rain. To enlarge this dataset, we designed an adapted CycleGAN, generating polarimetric images from RGB ones, with respect to the physical constraints. Empirical results demonstrated that, the polarimetric equivalent of road object detection benchmarks, are an added value to enhance road scene analysis.

From this labelled dataset, we were able to conduct experiments to evaluate the invariance of polarimetric features to weather changes. To this aim, we compared the ability of color-based and polarimetric features learnt exclusively in good weather conditions to detect road objects under fog. The experiment demonstrated that polarimetric features are more robust to weather changes than color-based ones. Further experiments demonstrated that the polarimetric intensities and the Stokes parameters are the most adapted polarimetric features to enhance road scene analysis under fog.

On top of that, we tested several fusion schemes, including an Early fusion pipeline and five Late fusion pipelines, in order to improve road scene analysis under fog. It

was shown that the Double soft-NMS filter and the OR filter applied to the Late fusion scheme were able to enhance the detection scores. Finally, these two pipelines are used to extend the results obtained in foggy conditions to a wide range of adverse weather conditions. Scenes under ten different fog densities and under rain are evaluated. The results demonstrated once again the ability of polarimetric features to detect road objects in adverse weather conditions.

To summarize up, we demonstrated in this thesis that polarimetric imaging is invariant to visibility changes, particularly induced by fog, rain, or the glare, unlike conventional sensors. It would be a real asset to enhance road scene analysis in adverse weather conditions, which is paramount in autonomous navigation.

Perspectives

Based on the work presented in this thesis, we now discuss some interesting perspectives for future research.

In Chapter 4, we presented the different datasets constituted to perform the experiments in our work and the designed pipeline for polarimetric images generation under constraints. Although the constituted datasets are the first large, publicly available datasets for road scene analysis in adverse weather, some improvements could be made to carry further experiments. First, the constituted datasets are not self-sufficient to perform an efficient training from scratch. As a matter of fact, it was paramount to use pre-trained deep architectures to avoid over-fitting on this dataset. One solution would be to augment the dataset with further road scenes, preferably from other cities than Rouen to increase its variability. Moreover, because the dataset does not contain enough instances of the classes "bike" and "motorbike", the evaluation could not be conducted on these objects. It would be beneficial to collect more instances of these two classes to generalize the obtained results to a wide range of road uses. Integrating the generated polarimetric images to the training process would also reinforce the training process. Moreover, the frames of the dataset are collected according to their sensor's fps rate which does not match their real fps rate, resulting in an inconsistent offset between each pair of images. It would be interesting to sort again the recordings of the acquisition campaigns to come up with strictly multimodal paired images. Finally, regarding the designed CycleGAN, another metric to evaluate the generated images could be designed. As a matter of fact, the current pipeline performs a pixel-wise evaluation of the physical constraints. By taking the whole object into account, it would reinforce the physical admissibility of the objects, resulting in more realistic polarimetric images.

Chapters 5 and 6 respectively presented the experiments demonstrating the invariance of polarimetric features to weather changes and the polarimetric and color-based fusion pipelines improving road scene analysis. Even if these experiments opened the path to further research on polarimetric images, to overcome unexpected weather changes in autonomous navigation, they have some drawbacks. The experiments conducted on adverse scenes are limited to a restricted area. Indeed, adverse conditions are not the most common and acquisition campaigns in such conditions are difficult to plan, since the weather is often unpredictable. One solution would be to conduct

extensive acquisition campaigns to maximize the chances to collect several weather conditions. Regarding the fusion scheme itself, it would be interesting to explore a Middle fusion scheme to explore intra-modalities representation during the training process. Finally, the experiments demonstrated the limits of polarimetric features in very dense fog. To overcome this limitation, a pre-processing step should be considered to restore the visibility of images. As it is reviewed in the literature, several dehazing algorithms have shown outstanding performances in retrieving polarimetric scenes' details.

From all the work accomplished in this thesis and the listed perspectives, we hope that polarimetric sensors will be popularized in autonomous navigation. Combined with the usual non-conventional sensors, they would provide further information on the scene, more robust to unexpected weather changes, which would be a great asset to autonomous navigation.

Conclusion et perspectives

Conclusion

Cette thèse vise à améliorer l'analyse des scènes routières en conditions météorologiques dégradées, en fusionnant des caractéristiques polarimétriques et des images couleur. Dans le chapitre 1, nous avons introduit le concept de multimodalité. Le formalisme de polarisation est d'abord détaillé, suivi des modèles de couleur et d'autres modalités non conventionnelles telles que l'imagerie infrarouge ou les nuages de points LiDAR. Le chapitre 2 présente les connaissances de base sur l'apprentissage profond, en particulier le paradigme de la détection d'objets, mais aussi un aperçu du CycleGAN. Le chapitre 3 passe en revue la littérature sur les modalités non conventionnelles, ainsi que les détecteurs d'objets, en montrant comment ils s'intègrent dans les pipelines des systèmes autonomes pour améliorer l'analyse des scènes routières en situations complexes. Le chapitre 4 présente les jeux de données constitués pour effectuer l'analyse de scènes routières dans des conditions météorologiques défavorables à l'aide de caractéristiques polarimétriques, ainsi que les formats de données polarimétriques encodés pour l'apprentissage automatique. Il décrit également le CycleGAN adapté conçu pour générer des images polarimétriques sous contraintes physiques à partir d'images RGB, ce qui permet d'avoir un équivalent polarimétrique des bases de données repères pour la détection d'obstacles routiers. Une démonstration de l'invariance des caractéristiques polarimétriques aux conditions de visibilité est faite dans le chapitre 5, en comparant leurs performances de détection avec celles des images couleur pour des scènes routières dans le brouillard. Enfin, ces résultats sont améliorés et étendus à d'autres conditions météorologiques défavorables dans le chapitre 6, en concevant un schéma de fusion adapté, basé sur l'imagerie couleur et la polarimétrie.

Plus particulièrement, nous avons constitué dans cette thèse le premier grand jeu de données multimodal, contenant des images polarimétriques et leur équivalent en imagerie couleur, étiqueté pour la détection d'obstacles routiers dans diverses conditions météorologiques. Il contient à la fois des scènes routières acquises en conditions réelles et des scènes routières acquises dans un tunnel simulant du brouillard et de la pluie. Pour élargir ce jeu de données, nous avons conçu un CycleGAN adapté, générant des images polarimétriques à partir d'images RGB, respectant les contraintes physiques de la polarimétrie. Les résultats empiriques ont démontré que l'équivalent polarimétrique des bases de données repères, utilisées pour la détection d'obstacles routiers, constitue une valeur ajoutée pour améliorer l'analyse de scènes routières.

À partir de cette base de données étiquetées, nous avons pu mener des expériences pour évaluer l'invariance des caractéristiques polarimétriques aux changements

météorologiques. De ce fait, nous avons comparé la capacité des caractéristiques polarimétriques et des caractéristiques basées sur la couleur, apprises exclusivement dans de bonnes conditions météorologiques, à détecter des obstacles routiers dans le brouillard. L'expérience a démontré que les caractéristiques polarimétriques sont plus robustes aux changements météorologiques que celles basées sur la couleur. D'autres expériences ont démontré que les intensités polarimétriques et les paramètres de Stokes sont les caractéristiques polarimétriques les plus adaptées pour améliorer l'analyse des scènes routières sous le brouillard.

En outre, nous avons testé plusieurs schémas de fusion, dont un schéma de fusion précoce et cinq schémas de fusion tardive, afin d'améliorer l'analyse des scènes routières dans le brouillard. Nous avons démontré que le filtre Double soft-NMS et le filtre OR appliqués au schéma de fusion tardive étaient capables d'améliorer les scores de détection. Enfin, ces deux pipelines sont utilisés pour étendre les résultats obtenus dans le brouillard à un large éventail de conditions météorologiques dégradées. Des scènes sous dix densités de brouillard différentes et sous la pluie intense sont évaluées. Les résultats ont démontré une fois de plus la capacité des caractéristiques polarimétriques à détecter des obstacles routiers en conditions météorologiques dégradées.

Pour résumer, nous avons démontré dans cette thèse que l'imagerie polarimétrique est invariante aux changements de visibilité, notamment induits par le brouillard, la pluie, ou l'éblouissement, contrairement aux capteurs conventionnels. Cette propriété constitue un réel atout pour améliorer l'analyse des scènes routières dans des conditions météorologiques dégradées, ce qui est primordial pour permettre la navigation autonome.

Perspectives

En nous basant sur les travaux présentés dans ce manuscrit, nous abordons maintenant quelques perspectives intéressantes pour des recherches futures.

Dans le chapitre 4, nous avons présenté les différents jeux de données constitués pour réaliser les expériences de notre travail et l'algorithme conçu pour la génération d'images polarimétriques sous contraintes. Bien que les jeux de données constitués soient les premiers grands jeux de données accessibles au public pour l'analyse de scènes routières lorsque les conditions météorologiques sont dégradées, certaines améliorations pourraient être faites pour réaliser d'autres expériences. Tout d'abord, les ensembles de données constitués ne sont pas suffisants pour effectuer un entraînement efficace à partir de zéro. Il était primordial d'utiliser des architectures profondes pré-entraînées pour éviter un sur-apprentissage sur ce jeu de données. Une solution serait d'enrichir le jeu de données avec d'autres scènes routières, de préférence provenant d'autres villes que Rouen pour augmenter sa variabilité. Par ailleurs, le jeu de données ne contenant pas suffisamment d'instances des classes "vélo" et "moto", l'évaluation n'a pas pu être menée sur ces objets. Il serait bénéfique de collecter plus d'instances de ces deux classes pour généraliser les résultats obtenus à d'autres d'usagers de la route. L'intégration des images polarimétriques générées au processus d'apprentissage renforcerait également ce dernier. De plus, les images du jeu de données sont collectées selon le taux de capture d'images par seconde théorique de leur capteur qui ne correspond pas à leur taux

de capture d'images par seconde réel, ce qui entraîne un décalage inconstant entre chaque paire d'images. Il serait intéressant de trier à nouveau les enregistrements des campagnes d'acquisition pour aboutir à des images multimodales strictement appariées. Enfin, en ce qui concerne le CycleGAN conçu, une autre métrique pour évaluer les images générées pourrait être mise au point. En effet, le pipeline actuel effectue une évaluation des contraintes physiques au pixel près. En prenant en compte l'ensemble de l'objet, on renforcerait l'admissibilité physique de ceux-ci, ce qui permettrait d'obtenir des images polarimétriques plus réalistes.

Les chapitres 5 et 6 ont respectivement présenté les expériences démontrant l'invariance des caractéristiques polarimétriques aux changements météorologiques et les schémas de fusion d'images polarimétriques et couleur améliorant l'analyse des scènes routières. Même si ces expériences ont ouvert la voie à la recherche sur les images polarimétriques pour surmonter les changements météorologiques inattendus dans la navigation autonome, elles présentent certains inconvénients. En effet, les expériences menées conditions dégradées sont effectuée dans une zone restreinte. En effet, les conditions dégradées ne sont pas les plus courantes et les campagnes d'acquisition dans de telles conditions sont difficiles à planifier, car les conditions météorologiques sont souvent imprévisibles. Une solution serait de mener des campagnes d'acquisition intensives, afin de maximiser les chances de collecter plusieurs conditions météorologiques. En ce qui concerne le schéma de fusion lui-même, il serait intéressant d'explorer un schéma de fusion intermédiaire pour explorer une représentation intra-modale des données pendant le processus d'enraînement. Enfin, les expériences ont démontré les limites des caractéristiques polarimétriques dans un brouillard très dense. Pour surmonter cette limitation, une étape de prétraitement devrait être envisagée pour restaurer la visibilité des images. En effet, comme on peut le voir dans l'état de l'art, plusieurs algorithmes de désembrumage ont montré des performances remarquables dans la reconstruction détaillée des scènes polarimétriques.

En nous basant sur l'ensemble du travail accompli dans cette thèse et des perspectives énumérées, nous espérons que l'usage des capteurs polarimétriques sera popularisé dans la navigation autonome. Combinés avec les capteurs non conventionnels habituels, ils fourniraient des informations supplémentaires sur la scène, plus robustes aux changements météorologiques inattendus, ce qui serait un grand atout pour la voiture autonome.

Bibliography

[1] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[2] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[3] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2999–3007, 2017.

[4] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[5] Sadayuki Tsugawa, Teruo Yatabe, Takeshi Hirose, and Shuntetsu Matsumoto. An automobile with artificial intelligence. In *Proceedings of the 6th international joint conference on Artificial intelligence-Volume 2*, pages 893–895, 1979.

[6] Charles Thorpe, Martial H Hebert, Takeo Kanade, and Steven A Shafer. Vision and navigation for the carnegie-mellon navlab. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(3):362–373, 1988.

[7] Reinhold Behringer, Sundar Sundareswaran, Brian Gregory, Richard Elsley, Bob Addison, Wayne Guthmiller, Robert Daily, and David Bevly. The darpa grand challenge-development of an autonomous vehicle. In *IEEE Intelligent Vehicles Symposium, 2004*, pages 226–231. IEEE, 2004.

[8] SAE J3016 L evels Of Driving. `https://www.sae.org/news/2019/01/sae-updates-j3016-automated-driving-graphic`. Accessed: 2021-02-07.

[9] Mario Bijelic, Tobias Gruber, and Werner Ritter. Benchmarking image sensors under adverse weather conditions for autonomous driving. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1773–1779. IEEE, 2018.

[10] Shizhe Zang, Ming Ding, David Smith, Paul Tyler, Thierry Rakotoarivelo, and Mohamed Ali Kaafar. The impact of adverse weather conditions on autonomous

vehicles: how rain, snow, fog, and hail affect the performance of a self-driving car. *IEEE vehicular technology magazine*, 14(2):103–111, 2019.

[11] P Govardhan and Umesh C Pati. Nir image based pedestrian detection in night vision with cascade classification and validation. In *2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies*, pages 1435–1438. IEEE, 2014.

[12] Nicolas Pinchon, Olivier Cassignol, Adrien Nicolas, Frédéric Bernardin, Patrick Leduc, Jean-Philippe Tarel, Roland Brémond, Emmanuel Bercier, and Johann Brunet. All-weather vision for automotive safety: which spectral band? In *International Forum on Advanced Microsystems for Automotive Applications*, pages 3–15. Springer, 2018.

[13] Massimo Bertozzi, Alberto Broggi, Alessandra Fascioli, Thorsten Graf, and M-M Meinecke. Pedestrian detection for driver assistance using multiresolution infrared vision. *IEEE transactions on vehicular technology*, 53(6):1666–1678, 2004.

[14] Marcel Sheeny, Emanuele De Pellegrin, Saptarshi Mukherjee, Alireza Ahrabian, Sen Wang, and Andrew Wallace. Radiate: A radar dataset for automotive perception. *arXiv preprint arXiv:2010.09076*, 3(4):7, 2020.

[15] Shiping Song, Jian Wu, Sumin Zhang, Yunhang Liu, and Shun Yang. Research on target tracking algorithm using millimeter-wave radar on curved road. *Mathematical Problems in Engineering*, 2020, 2020.

[16] Taewan Kim and Joydeep Ghosh. Robust detection of non-motorized road users using deep learning on optical and lidar data. In *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on*, pages 271–276. IEEE, 2016.

[17] Robin Heinzler, Florian Piewak, Philipp Schindler, and Wilhelm Stork. Cnn-based lidar point cloud de-noising in adverse weather. *IEEE Robotics and Automation Letters*, 5(2):2514–2521, 2020.

[18] Michael Bass, Eric W Van Stryland, David R Williams, and William L Wolfe. *Handbook of optics*, volume 2. McGraw-Hill New York, 1995.

[19] Ji Qi and Daniel S Elson. Mueller polarimetric imaging for surgical and diagnostic applications: a review. *Journal of biophotonics*, 10(8):950–982, 2017.

[20] Jian Liang, Liyong Ren, Enshi Qu, Bingliang Hu, and Yingli Wang. Method for enhancing visibility of hazy images based on polarimetric imaging. *Photonics Research*, 2(1):38–44, 2014.

[21] Zhaopeng Cui, Jinwei Gu, Boxin Shi, Ping Tan, and Jan Kautz. Polarimetric multi-view stereo. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1558–1567, 2017.

[22] Wang Fan, Samia Ainouz, Fabrice Meriaudeau, and Abdelaziz Bensrhair. Polarization-based car detection. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 3069–3073. IEEE, 2018.

[23] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[24] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018.

[25] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020.

[26] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European conference on computer vision (ECCV)*, pages 734–750, 2018.

[27] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6569–6578, 2019.

[28] Xingyi Zhou, Jiacheng Zhuo, and Philipp Krahenbuhl. Bottom-up object detection by grouping extreme and center points. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 850–859, 2019.

[29] Ze Yang, Shaohui Liu, Han Hu, Liwei Wang, and Stephen Lin. Reppoints: Point set representation for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9657–9666, 2019.

[30] Shu Wang Jingjing Liu, Shaoting Zhang and Dimitris Metaxas. Multispectral deep neural networks for pedestrian detection. In *Proc. British Machine Vision Conf.*, pages 73.1–73.13, September 2016.

[31] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 6526–6534. IEEE, 2017.

[32] Ramin Nabati and Hairong Qi. Rrpn: Radar region proposal network for object detection in autonomous vehicles. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 3093–3097. IEEE, 2019.

[33] Sergio Casas, Wenjie Luo, and Raquel Urtasun. Intentnet: Learning to predict intention from raw sensor data. In *Conference on Robot Learning*, pages 947–956, 2018.

[34] Di Feng, Christian Haase-Schütz, Lars Rosenbaum, Heinz Hertlein, Claudius Glaeser, Fabian Timm, Werner Wiesbeck, and Klaus Dietmayer. Deep multimodal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[35] Soonmin Hwang, Jaesik Park, Namil Kim, Yukyung Choi, and In So Kweon. Multispectral pedestrian detection: Benchmark dataset and baseline. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1037–1045, 2015.

[36] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361. IEEE, 2012.

[37] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2446–2454, 2020.

[38] Dana Lahat, Tülay Adali, and Christian Jutten. Multimodal data fusion: an overview of methods, challenges, and prospects. *Proceedings of the IEEE*, 103(9):1449–1477, 2015.

[39] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen Awm Van Der Laak, Bram Van Ginneken, and Clara I Sánchez. A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88, 2017.

[40] Sang-Hack Jung, Jayan Eledath, Stefan Johansson, and Vincent Mathevon. Egomotion estimation in monocular infra-red image sequence for night vision applications. In *2007 IEEE Workshop on Applications of Computer Vision (WACV'07)*, pages 8–8. IEEE, 2007.

[41] Massimo Bertozzi, Alberto Broggi, Mirko Felisa, Guido Vezzoni, and Michael Del Rose. Low-level pedestrian detection by means of visible and far infra-red tetra-vision. In *2006 IEEE Intelligent Vehicles Symposium*, pages 231–236. IEEE, 2006.

[42] Mohammad Aldibaja, Noaki Suganuma, and Keisuke Yoneda. Improving localization accuracy for autonomous driving in snow-rain environments. In *2016 IEEE/SICE International Symposium on System Integration (SII)*, pages 212–217. IEEE, 2016.

[43] Luca Caltagirone, Mauro Bellone, Lennart Svensson, and Mattias Wahde. Lidar–camera fusion for road detection using fully convolutional neural networks. *Robotics and Autonomous Systems*, 111:125–131, 2019.

[44] Xiangyun Hu, Yijing Li, Jie Shan, Jianqing Zhang, and Yongjun Zhang. Road centerline extraction in complex urban scenes from lidar data based on multiple features. *IEEE Transactions on Geoscience and Remote Sensing*, 52(11):7448–7456, 2014.

[45] James Clerk Maxwell. Viii. a dynamical theory of the electromagnetic field. *Philosophical transactions of the Royal Society of London*, (155):459–512, 1865.

[46] Max Born and Emil Wolf. *Principles of optics: electromagnetic theory of propagation, interference and diffraction of light*. Elsevier, 2013.

[47] Samia Ainouz, Olivier Morel, David Fofi, Saleh Mosaddegh, and Abdelaziz Bensrhair. Adaptive processing of catadioptric images using polarization imaging: towards a pola-catadioptric model. *Optical engineering*, 52(3):037001, 2013.

[48] H von Helmholtz. Lxxxi. on the theory of compound colours. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 4(28):519–534, 1852.

[49] James Clerk Maxwell. Iv. on the theory of compound colours, and the relations of the colours of the spectrum. *Philosophical Transactions of the Royal Society of London*, (150):57–84, 1860.

[50] Alvy Ray Smith. Color gamut transform pairs. *ACM Siggraph Computer Graphics*, 12(3):12–19, 1978.

[51] Thomas Smith and John Guild. The cie colorimetric standards and their use. *Transactions of the optical society*, 33(3):73, 1931.

[52] Alan R Robertson. The cie 1976 color-difference formulae. *Color Research & Application*, 2(1):7–11, 1977.

[53] William B. Pennebaker and Joan L. Mitchell. *JPEG Still Image Data Compression Standard*. Van Nostrand Reinhold, New York, 1992.

[54] William Herschel. Xiv. experiments on the refrangibility of the invisible rays of the sun. *Philosophical Transactions of the Royal Society of London*, (90):284–292, 1800.

[55] Christian Herrmann, Miriam Ruf, and Jürgen Beyerer. Cnn-based thermal infrared person detection by domain adaptation. In *Autonomous Systems: Sensors, Vehicles, Security, and the Internet of Everything*, volume 10643, page 1064308. International Society for Optics and Photonics, 2018.

[56] Michael Teutsch, Thomas Muller, Marco Huber, and Jurgen Beyerer. Low resolution person detection with a moving thermal infrared camera by hot spot classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 209–216, 2014.

[57] Jisoo Park, Jingdao Chen, Yong K Cho, Dae Y Kang, and Byung J Son. Cnn-based person detection using infrared images for night-time intrusion warning systems. *Sensors*, 20(1):34, 2020.

[58] Theodore H Maiman et al. Stimulated optical radiation in ruby. 1960.

[59] Stephen E Reutebuch, Robert J McGaughey, Hans-Erik Andersen, and Ward W Carson. Accuracy of a high-resolution lidar terrain model under a conifer forest canopy. *Canadian journal of remote sensing*, 29(5):527–535, 2003.

[60] Bisheng Yang, Lina Fang, Qingquan Li, and Jonathan Li. Automated extraction of road markings from mobile lidar point clouds. *Photogrammetric Engineering & Remote Sensing*, 78(4):331–338, 2012.

[61] Zhuang Jie Chong, Baoxing Qin, Tirthankar Bandyopadhyay, Marcelo H Ang, Emilio Frazzoli, and Daniela Rus. Synthetic 2d lidar for precise vehicle localization in 3d urban environment. In *2013 IEEE International Conference on Robotics and Automation*, pages 1554–1559. IEEE, 2013.

[62] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[63] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[64] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[65] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[66] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[67] Richard Socher, Yoshua Bengio, and Christopher D Manning. Deep learning for nlp (without magic). In *Tutorial Abstracts of ACL 2012*, pages 5–5. 2012.

[68] Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. Multi-task learning for multiple language translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1723–1732, 2015.

[69] Sorin Grigorescu, Bogdan Trasnea, Tiberiu Cocias, and Gigel Macesanu. A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, 37(3):362–386, 2020.

[70] S McCulloch Warren and Pitts Walter. A logical calculus of the ideas immanent in nervous activity. *Bulletin of mathematical Biophysics*, 5(4):115–133, 1943.

[71] Frank Rosenblatt. *The perceptron, a perceiving and recognizing automaton Project Para*. Cornell Aeronautical Laboratory, 1957.

[72] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[73] Daniel Wallach and Bruno Goffinet. Mean squared error of prediction as a criterion for evaluating and comparing system models. *Ecological modelling*, 44(3-4):299–306, 1989.

[74] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[75] Alexander Neubeck and Luc Van Gool. Efficient non-maximum suppression. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 3, pages 850–855. IEEE, 2006.

[76] N Bodla, B Singh, R Chellappa, and LS Davis. Improving object detection with one line of code. 2017. *Source:< https://arxiv. org/pdf/1704.04503. pdf*.

[77] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[78] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

[79] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2794–2802, 2017.

[80] Mayank Bansal, Alex Krizhevsky, and Abhijit Ogale. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. *arXiv preprint arXiv:1812.03079*, 2018.

[81] Andrej Karpathy's presentation at pytorch 2019. `https://www.youtube.com/watch?v=oBkltKXtDE`. Accessed: 2021-30-06.

[82] Jean Rehbinder, Huda Haddad, Stanislas Deby, Benjamin Teig, André Nazac, Tatiana Novikova, Angelo Pierangelo, and François Moreau. Ex vivo mueller polarimetric imaging of the uterine cervix: a first statistical evaluation. *Journal of biomedical optics*, 21(7):071113, 2016.

[83] Angelo Pierangelo, André Nazac, Abdelali Benali, Pierre Validire, Henri Cohen, Tatiana Novikova, Bicher Haj Ibrahim, Sandeep Manhas, Clément Fallet, Maria-Rosaria Antonelli, et al. Polarimetric imaging of uterine cervix: a case study. *Optics express*, 21(12):14120–14130, 2013.

[84] Tatiana Novikova, Angelo Pierangelo, Antonello De Martino, Abdelali Benali, and Pierre Validire. Polarimetric imaging for cancer diagnosis and staging. *Optics and photonics news*, 23(10):26–33, 2012.

[85] M Anastasiadou, A De Martino, D Clement, F Liège, B Laude-Boulesteix, N Quang, J Dreyfuss, B Huynh, A Nazac, L Schwartz, et al. Polarimetric imaging for the diagnosis of cervical cancer. *physica status solidi c*, 5(5):1423–1426, 2008.

[86] Angelo Pierangelo, Abdelali Benali, Maria-Rosaria Antonelli, Tatiana Novikova, Pierre Validire, Brice Gayet, and Antonello De Martino. Ex-vivo characterization of human colon cancer by mueller polarimetric imaging. *Optics express*, 19(2):1582–1593, 2011.

[87] Angelo Pierangelo, Sandeep Manhas, Abdelali Benali, Clément Fallet, Jean-Laurent Totobenazara, Maria Rosaria Antonelli, Tatiana Novikova, Brice Gayet, Antonello De Martino, and Pierre Validire. Multispectral mueller polarimetric imaging detecting residual cancer and cancer regression after neoadjuvant treatment for colorectal carcinomas. *Journal of biomedical optics*, 18(4):046014, 2013.

[88] Tatiana Novikova, Jean Rehbinder, Jérémy Vizet, Angelo Pierangelo, Razvigor Ossikovski, André Nazac, Abdelali Benali, and Pierre Validire. Mueller polarimetry as a tool for optical biopsy of tissue. In *2018 International Conference Laser Optics (ICLO)*, pages 553–553. IEEE, 2018.

[89] Xianping Fu, Zheng Liang, Xueyan Ding, Xinyue Yu, and Yafei Wang. Image descattering and absorption compensation in underwater polarimetric imaging. *Optics and Lasers in Engineering*, 132:106115, 2020.

[90] Yalong Gu, Carlos Carrizo, Alexander A Gilerson, Parrish C Brady, Molly E Cummings, Michael S Twardowski, James M Sullivan, Amir I Ibrahim, and George W Kattawar. Polarimetric imaging and retrieval of target polarization characteristics in underwater environment. *Applied optics*, 55(3):626–637, 2016.

[91] Haofeng Hu, Lin Zhao, Xiaobo Li, Hui Wang, and Tiegen Liu. Underwater image recovery under the nonuniform optical field based on polarimetric imaging. *IEEE Photonics Journal*, 10(1):1–9, 2018.

[92] Heng Tian, Jingping Zhu, Shuwen Tan, Yunyao Zhang, Yang Zhang, Yingchao Li, and Xun Hou. Rapid underwater target enhancement method based on polarimetric imaging. *Optics & laser technology*, 108:515–520, 2018.

[93] M Dubreuil, P Delrot, Isabelle Leonard, Ayman Alfalou, Christian Brosseau, and Aristide Dogariu. Exploring underwater target detection by imaging polarimetry and correlation techniques. *Applied optics*, 52(5):997–1005, 2013.

[94] Alex Gilerson, Carlos Carrizo, Alberto Tonizzo, Amir Ibrahim, Ahmed El-Habashi, Robert Foster, and Samir Ahmed. Polarimetric imaging of underwater targets. In *Ocean Sensing and Monitoring V*, volume 8724, page 872403. International Society for Optics and Photonics, 2013.

[95] Xiaobo Li, Haofeng Hu, Lin Zhao, Hui Wang, Yin Yu, Lan Wu, and Tiegen Liu. Polarimetric image recovery method combining histogram stretching for underwater imaging. *Scientific reports*, 8(1):1–10, 2018.

[96] Haofeng Hu, Lin Zhao, Bingjing Huang, Xiaobo Li, Hui Wang, and Tiegen Liu. Enhancing visibility of polarimetric underwater image by transmittance correction. *IEEE Photonics Journal*, 9(3):1–10, 2017.

[97] Feng Xu and Y-Q Jin. Imaging simulation of polarimetric sar for a comprehensive terrain scene using the mapping and projection algorithm. *IEEE Transactions on Geoscience and Remote Sensing*, 44(11):3219–3234, 2006.

[98] Shane R Cloude and Konstantinos P Papathanassiou. Polarimetric sar interferometry. *IEEE Transactions on geoscience and remote sensing*, 36(5):1551–1565, 1998.

[99] Anthony Freeman and Stephen L Durden. A three-component scattering model for polarimetric sar data. *IEEE transactions on geoscience and remote sensing*, 36(3):963–973, 1998.

[100] Yu Zhou, Haipeng Wang, Feng Xu, and Ya-Qiu Jin. Polarimetric sar image classification using deep convolutional neural networks. *IEEE Geoscience and Remote Sensing Letters*, 13(12):1935–1939, 2016.

[101] V Alberga. A study of land cover classification using polarimetric sar parameters. *International Journal of Remote Sensing*, 28(17):3851–3870, 2007.

[102] Zhimian Zhang, Haipeng Wang, Feng Xu, and Ya-Qiu Jin. Complex-valued convolutional neural network and its application in polarimetric sar image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 55(12):7177–7188, 2017.

[103] Ronghua Shang, Jianghai He, Jiaming Wang, Kaiming Xu, Licheng Jiao, and Rustam Stolkin. Dense connection and depthwise separable convolution based cnn for polarimetric sar image classification. *Knowledge-Based Systems*, 194:105542, 2020.

[104] Yoav Y Schechner, Srinivasa G Narasimhan, and Shree K Nayar. Polarization-based vision through haze. *Applied optics*, 42(3):511–525, 2003.

[105] Sarit Shwartz, Einav Namer, and Yoav Y Schechner. Blind haze separation. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1984–1991. IEEE, 2006.

[106] Wenfei Zhang, Jian Liang, Liyong Ren, Haijuan Ju, Zhaofeng Bai, and Zhaoxin Wu. Fast polarimetric dehazing method for visibility enhancement in hsi colour space. *Journal of Optics*, 19(9):095606, 2017.

[107] Fei Liu, Lei Cao, Xiaopeng Shao, Pingli Han, and Xiangli Bin. Polarimetric dehazing utilizing spatial frequency segregation of images. *Applied optics*, 54(27):8116–8122, 2015.

[108] Jian Liang, Liyong Ren, Haijuan Ju, Wenfei Zhang, and Enshi Qu. Polarimetric dehazing method for dense haze removal based on distribution analysis of angle of polarization. *Optics express*, 23(20):26146–26157, 2015.

[109] Kai Berger, Randolph Voorhies, and Larry H Matthies. Depth from stereo polarization in specular scenes for urban robotics. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1966–1973. IEEE, 2017.

[110] Dizhong Zhu and William A. P. Smith. Depth from a polarisation + rgb stereo pair. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[111] Luwei Yang, Feitong Tan, Ao Li, Zhaopeng Cui, Yasutaka Furukawa, and Ping Tan. Polarimetric dense monocular slam. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3857–3866, 2018.

[112] Moein Shakeri, Shing Yang Loo, Hong Zhang, and Kangkang Hu. Polarimetric monocular dense mapping using relative deep depth prior. *IEEE Robotics and Automation Letters*, 6(3):4512–4519, 2021.

[113] Xin Shen, Artur Carnicer, and Bahram Javidi. Three-dimensional polarimetric integral imaging under low illumination conditions. *Optics letters*, 44(13):3230–3233, 2019.

[114] Olivier Morel, Christophe Stolz, Fabrice Meriaudeau, and Patrick Gorria. Active lighting applied to three-dimensional reconstruction of specular metallic surfaces by polarization imaging. *Applied optics*, 45(17):4062–4068, 2006.

[115] Rindra Rantoson, Christophe Stolz, David Fofi, and Fabrice Mériaudeau. 3d reconstruction by polarimetric imaging method based on perspective model. In *Optical Measurement Systems for Industrial Inspection VI*, volume 7389, page 73890C. International Society for Optics and Photonics, 2009.

[116] Xiao Xiao, Bahram Javidi, Genaro Saavedra, Michael Eismann, and Manuel Martinez-Corral. Three-dimensional polarimetric computational integral imaging. *Optics express*, 20(14):15481–15488, 2012.

[117] Osamu Matoba and Bahram Javidi. Three-dimensional polarimetric integral imaging. *Optics letters*, 29(20):2375–2377, 2004.

[118] Daniel F Huber, Louis Denes, Martial Hebert, Milton Gottlieb, Boris Kaminsky, and Peter Metes. A spectro-polarimetric imager for intelligent transportation systems. 1997.

[119] Marc Blanchon, Olivier Morel, Yifei Zhang, Ralph Seulin, Nathan Crombez, and Désiré Sidibé. Outdoor scenes pixel-wise semantic segmentation using polarimetry and fully convolutional network. In *14th International Conference on Computer Vision Theory and Applications (VISAPP 2019)*, 2019.

[120] Ning Li, Yongqiang Zhao, Quan Pan, Seong G Kong, and Jonathan Cheung-Wai Chan. Full-time monocular road detection using zero-distribution prior of angle of polarization. In *European Conference on Computer Vision*, pages 457–473. Springer, 2020.

[121] Yun Luo, Jeffrey Remillard, and Dieter Hoetzer. Pedestrian detection in near-infrared night vision system. In *2010 IEEE Intelligent Vehicles Symposium*, pages 51–58. IEEE, 2010.

[122] Omer Tsimhoni, Jonas Bärgman, and Michael J Flannagan. Pedestrian detection with near and far infrared night vision enhancement. *Leukos*, 4(2):113–128, 2007.

[123] Yihong Chen, Zheng Zhang, Yue Cao, Liwei Wang, Stephen Lin, and Han Hu. Reppoints v2: Verification meets regression for object detection. *Advances in Neural Information Processing Systems*, 33, 2020.

[124] Yingfeng Cai, Ze Liu, Hai Wang, and Xiaoqiang Sun. Saliency-based pedestrian detection in far infrared images. *IEEE Access*, 5:5013–5019, 2017.

[125] Raluca Didona Brehar, Mircea Paul Muresan, Tiberiu Mariţa, Cristian-Cosmin Vancea, Mihai Negru, and Sergiu Nedevschi. Pedestrian street-cross action recognition in monocular far infrared sequences. *IEEE Access*, 9:74302–74324, 2021.

[126] Karol Piniarski and Paweł Pawłowski. Efficient pedestrian detection with enhanced object segmentation in far ir night vision. In *2017 Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA)*, pages 160–165. IEEE, 2017.

[127] Xiaobiao Dai, Yuxia Duan, Junping Hu, Shicai Liu, Caiqi Hu, Yunze He, Dapeng Chen, Chunlei Luo, and Jianqiao Meng. Near infrared nighttime road pedestrians recognition based on convolutional neural network. *Infrared Physics & Technology*, 97:25–32, 2019.

[128] Urban Meis, Wemer Ritter, and Heiko Neumann. Detection and classification of obstacles in night vision traffic scenes based on infrared imagery. In *Proceedings of the 2003 IEEE International Conference on Intelligent Transportation Systems*, volume 2, pages 1140–1144. IEEE, 2003.

[129] Claire Burke, Maisie Rashman, Serge Wich, Andy Symons, Cobus Theron, and Steve Longmore. Optimizing observing strategies for monitoring animals using

drone-mounted thermal infrared cameras. *International Journal of Remote Sensing*, 40(2):439–467, 2019.

[130] Roland Kays, James Sheppard, Kevin Mclean, Charlie Welch, Cris Paunescu, Victor Wang, Greg Kravit, and Meg Crofoot. Hot monkey, cold reality: surveying rainforest canopy mammals using drone-mounted thermal infrared sensors. *International journal of remote sensing*, 40(2):407–419, 2019.

[131] Tae-Yun Lee, Vladimir Skvortsov, Myung-Sik Kim, Seung-Hoon Han, and Min-Ho Ka. Application of w-band fmcw radar for road curvature estimation in poor visibility conditions. *IEEE Sensors Journal*, 18(13):5300–5312, 2018.

[132] Matthias Serfling, Roland Schweiger, and Werner Ritter. Road course estimation in a night vision application using a digital map, a camera sensor and a prototypical imaging radar system. In *2008 IEEE Intelligent Vehicles Symposium*, pages 810–815. IEEE, 2008.

[133] Chia-Chi Tsai, Yi-Ting Lai, Yuan-Fu Li, and Jiun-In Guo. A vision radar system for car safety driving applications. In *2017 International Symposium on VLSI Design, Automation and Test (VLSI-DAT)*, pages 1–4. IEEE, 2017.

[134] Abhilasha Srivastava, Abhishek Goyal, and Shobha Sundar Ram. Radar cross-section of potholes at automotive radar frequencies. In *2020 IEEE International Radar Conference (RADAR)*, pages 483–488. IEEE, 2020.

[135] Ashraf Abosekeen, Aboelmagd Noureldin, and Michael J Korenberg. Utilizing the acc-fmcw radar for land vehicles navigation. In *2018 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, pages 124–132. IEEE, 2018.

[136] Haiyan Guan, Yongtao Yu, Jonathan Li, and Pengfei Liu. Pole-like road object detection in mobile lidar data via supervoxel and bag-of-contextual-visual-words representation. *IEEE Geoscience and Remote Sensing Letters*, 13(4):520–524, 2016.

[137] Bichen Wu, Xuanyu Zhou, Sicheng Zhao, Xiangyu Yue, and Kurt Keutzer. Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 4376–4382. IEEE, 2019.

[138] Bichen Wu, Alvin Wan, Xiangyu Yue, and Kurt Keutzer. Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1887–1893. IEEE, 2018.

[139] Jingrong Chen, Hao Xu, Jianqing Wu, Rui Yue, Changwei Yuan, and Lu Wang. Deer crossing road detection with roadside lidar sensor. *Ieee Access*, 7:65944–65954, 2019.

[140] Zijian Zhu, Xu Li, Jianhua Xu, Jianhua Yuan, and Ju Tao. Unstructured road segmentation based on road boundary enhancement point-cylinder network using lidar sensor. *Remote Sensing*, 13(3):495, 2021.

[141] Yongtao Yu, Jonathan Li, Haiyan Guan, Fukai Jia, and Cheng Wang. Learning hierarchical features for automated extraction of road markings from 3-d mobile lidar point clouds. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 8(2):709–726, 2014.

[142] You Li and Javier Ibanez-Guzman. Lidar for autonomous driving: The principles, challenges, and trends for automotive lidar and perception systems. *IEEE Signal Processing Magazine*, 37(4):50–61, 2020.

[143] KA Cyran and T Niedziela. Infrared images in automatic recognition of the type of road obstacle in a fog. *Archives of Transport*, 18(4):29–38, 2006.

[144] Yoichiro Iwasaki, Masato Misumi, and Toshiyuki Nakamiya. Robust vehicle detection under various environmental conditions using an infrared thermal camera and its application to road traffic flow monitoring. *Sensors*, 13(6):7756–7773, 2013.

[145] Jatin Talwar, Love Kumar, and Gurmohan Singh. Infrared vision for fog detection system to improve road visibility. In *2019 International Conference on Communication and Electronics Systems (ICCES)*, pages 882–886. IEEE, 2019.

[146] Kelsey M Judd, Michael P Thornton, and Austin A Richards. Automotive sensing: assessing the impact of fog on lwir, mwir, swir, visible, and lidar performance. In *Infrared Technology and Applications XLV*, volume 11002, page 110021F. International Society for Optics and Photonics, 2019.

[147] Jean Dumoulin, Vincent Boucher, and Florian Greffier. Numerical and experimental evaluation of road infrastructure perception in fog and/or night conditions using infrared and photometric vision systems. In *Infrared Spaceborne Remote Sensing and Instrumentation XVII*, volume 7453, page 74530T. International Society for Optics and Photonics, 2009.

[148] L Colace, F Santoni, and G Assanto. A near-infrared optoelectronic approach to detection of road conditions. *Optics and Lasers in Engineering*, 51(5):633–636, 2013.

[149] Mats Riehm, Torbjörn Gustavsson, Jörgen Bogren, and Per-Erik Jansson. Ice formation detection on road surfaces using infrared thermometry. *Cold Regions Science and Technology*, 83-84:71–76, 2012.

[150] Patrik Jonsson and Mats Riehm. Infrared thermometry in winter road maintenance. *Journal of Atmospheric and Oceanic Technology*, 29(6):846–856, 2012.

[151] Patrik Jonsson. Remote sensor for winter road surface status detection. In *SENSORS, 2011 IEEE*, pages 1285–1288. IEEE, 2011.

[152] Robin Heinzler, Philipp Schindler, Jürgen Seekircher, Werner Ritter, and Wilhelm Stork. Weather influence and classification with automotive lidar sensors. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 1527–1534. IEEE, 2019.

[153] Matti Kutila, Pasi Pyykönen, Hanno Holzhüter, Michele Colomb, and Pierre Duthon. Automotive lidar performance verification in fog and rain. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 1695–1701. IEEE, 2018.

[154] Filip Majer, Zhi Yan, George Broughton, Yassine Ruichek, and Tomáš Krajník. Learning to see through haze: Radar-based human detection for adverse weather conditions. In *2019 European Conference on Mobile Robots (ECMR)*, pages 1–7. IEEE, 2019.

[155] Rodrigo Pérez, Falk Schubert, Ralph Rasshofer, and Erwin Biebl. Single-frame vulnerable road users classification with a 77 ghz fmcw radar sensor and a convolutional neural network. In *2018 19th International Radar Symposium (IRS)*, pages 1–10. IEEE, 2018.

[156] Kun Qian, Shilin Zhu, Xinyu Zhang, and Li Erran Li. Robust multimodal vehicle detection in foggy weather using complementary lidar and radar signals. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 444–453, 2021.

[157] Christian Szegedy, Alexander Toshev, and Dumitru Erhan. Deep neural networks for object detection. 2013.

[158] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[159] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1904–1916, 2015.

[160] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.

[161] Vladimir Vapnik, Isabel Guyon, and Trevor Hastie. Support vector machines. *Mach. Learn*, 20(3):273–297, 1995.

[162] Dumitru Erhan, Christian Szegedy, Alexander Toshev, and Dragomir Anguelov. Scalable object detection using deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2147–2154, 2014.

[163] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013.

[164] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[165] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[166] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[167] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.

[168] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[169] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[170] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8697–8710, 2018.

[171] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019.

[172] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: object detection via region-based fully convolutional networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 379–387, 2016.

[173] Bharat Singh, Mahyar Najibi, and Larry S Davis. Sniper: efficient multi-scale training. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 9333–9343, 2018.

[174] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8759–8768, 2018.

[175] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, et al. Hybrid task cascade for instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4974–4983, 2019.

[176] Yue Cao, Jiarui Xu, Stephen Lin, Fangyun Wei, and Han Hu. Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[177] Yanghao Li, Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. Scale-aware trident networks for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6054–6063, 2019.

[178] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014.

[179] Xin Lu, Buyu Li, Yuxin Yue, Quanquan Li, and Junjie Yan. Grid r-cnn. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7363–7372, 2019.

[180] Jiangmiao Pang, Kai Chen, Jianping Shi, Huajun Feng, Wanli Ouyang, and Dahua Lin. Libra r-cnn: Towards balanced learning for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 821–830, 2019.

[181] Jiale Cao, Hisham Cholakkal, Rao Muhammad Anwer, Fahad Shahbaz Khan, Yanwei Pang, and Ling Shao. D2det: Towards high quality object detection and instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11485–11494, 2020.

[182] Hongkai Zhang, Hong Chang, Bingpeng Ma, Naiyan Wang, and Xilin Chen. Dynamic r-cnn: Towards high quality object detection via dynamic training. In *European Conference on Computer Vision*, pages 260–275. Springer, 2020.

[183] Guanglu Song, Yu Liu, and Xiaogang Wang. Revisiting the sibling head in object detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11563–11572, 2020.

[184] Siyuan Qiao, Liang-Chieh Chen, and Alan Yuille. Detectors: Detecting objects with recursive feature pyramid and switchable atrous convolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10213–10224, 2021.

[185] Christian Szegedy, Scott Reed, Dumitru Erhan, Dragomir Anguelov, and Sergey Ioffe. Scalable, high-quality object detection. *arXiv preprint arXiv:1412.1441*, 2014.

[186] Lichao Huang, Yi Yang, Yafeng Deng, and Yinan Yu. Densebox: Unifying landmark localization with end to end object detection. *arXiv preprint arXiv:1509.04874*, 2015.

[187] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017.

[188] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.

[189] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[190] Ivan Krasin, Tom Duerig, Neil Alldrin, Vittorio Ferrari, Sami Abu-El-Haija, Alina Kuznetsova, Hassan Rom, Jasper Uijlings, Stefan Popov, Andreas Veit, et al. Openimages: A public dataset for large-scale multi-label and multi-class image classification. *Dataset available from https://github. com/openimages*, 2(3):18, 2017.

[191] Songtao Liu, Di Huang, and Yunhong Wang. Learning spatial fusion for single-shot object detection. *arXiv preprint arXiv:1911.09516*, 2019.

[192] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[193] Chien-Yao Wang, Hong-Yuan Mark Liao, Yueh-Hua Wu, Ping-Yang Chen, Jun-Wei Hsieh, and I-Hau Yeh. Cspnet: A new backbone that can enhance learning capability of cnn. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 390–391, 2020.

[194] Cheng-Yang Fu, Wei Liu, Ananth Ranga, Ambrish Tyagi, and Alexander C Berg. Dssd: Deconvolutional single shot detector. *arXiv preprint arXiv:1701.06659*, 2017.

[195] Shifeng Zhang, Longyin Wen, Xiao Bian, Zhen Lei, and Stan Z Li. Single-shot refinement neural network for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4203–4212, 2018.

[196] Qijie Zhao, Tao Sheng, Yongtao Wang, Zhi Tang, Ying Chen, Ling Cai, and Haibin Ling. M2det: A single-shot object detector based on multi-level feature pyramid network. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 9259–9266, 2019.

[197] Chenchen Zhu, Fangyi Chen, Zhiqiang Shen, and Marios Savvides. Soft anchor-point object detection. *arXiv preprint arXiv:1911.12448*, 2019.

[198] Chenchen Zhu, Yihui He, and Marios Savvides. Feature selective anchor-free module for single-shot object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 840–849, 2019.

[199] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9627–9636, 2019.

[200] Wei Ke, Tianliang Zhang, Zeyi Huang, Qixiang Ye, Jianzhuang Liu, and Dong Huang. Multiple anchor learning for visual object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10206–10215, 2020.

[201] Xiaosong Zhang, Fang Wan, Chang Liu, Xiangyang Ji, and Qixiang Ye. Learning to match anchors for visual object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[202] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10781–10790, 2020.

[203] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[204] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.

[205] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021.

[206] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European conference on computer vision*, pages 483–499. Springer, 2016.

[207] Abdullah Rashwan, Agastya Kalra, and Pascal Poupart. Matrix nets: A new deep architecture for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[208] Kaiwen Duan, Lingxi Xie, Honggang Qi, Song Bai, Qingming Huang, and Qi Tian. Corner proposal network for anchor-free, two-stage object detection. *arXiv preprint arXiv:2007.13816*, 2020.

[209] Sergio Alberto Rodriguez Florez, Vincent Frémont, Philippe Bonnifait, and Véronique Cherfaoui. Multi-modal object detection and localization for high integrity driving assistance. *Machine vision and applications*, 25(3):583–598, 2014.

[210] Jovan Radak, Bertrand Ducourthial, Véronique Cherfaoui, and Stéphane Bonnet. Detecting road events using distributed data fusion: Experimental evaluation for the icy roads case. *IEEE transactions on intelligent transportation systems*, 17(1):184–194, 2015.

[211] Jörg Wagner, Volker Fischer, Michael Herman, and Sven Behnke. Multispectral pedestrian detection using deep fusion convolutional neural networks.

[212] Andreas Pfeuffer and Klaus Dietmayer. Optimal sensor data fusion architecture for object detection in adverse weather conditions. In *Proceedings of International Conference on Information Fusion*, pages 2592 – 2599, 2018.

[213] Vishwanath A Sindagi, Yin Zhou, and Oncel Tuzel. Mvx-net: Multimodal voxelnet for 3d object detection. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 7276–7282. IEEE, 2019.

[214] Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 922–928. IEEE, 2015.

[215] Oier Mees, Andreas Eitel, and Wolfram Burgard. Choosing smartly: Adaptive multimodal fusion for object detection in changing environments. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pages 151–156. IEEE, 2016.

[216] Karasawa Takumi, Kohei Watanabe, Qishen Ha, Antonio Tejero-De-Pablos, Yoshitaka Ushiku, and Tatsuya Harada. Multispectral object detection for autonomous vehicles. In *Proceedings of the on Thematic Workshops of ACM Multimedia 2017*, pages 35–43. ACM, 2017.

[217] Kiwoo Shin, Youngwook Paul Kwon, and Masayoshi Tomizuka. Roarnet: A robust 3d object detection based on region approximation refinement. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 2510–2515. IEEE, 2019.

[218] Xinxin Du, Marcelo H Ang, Sertac Karaman, and Daniela Rus. A general pipeline for 3d detection of vehicles. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3194–3200. IEEE, 2018.

[219] Zhixin Wang and Kui Jia. Frustum convnet: Sliding frustums to aggregate local point-wise features for amodal 3D object detection. In *IEEE/RSJ Int. Conf. Intelligent Robots and Systems*. IEEE, 2019.

[220] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[221] Lukas Schneider, Manuel Jasch, Björn Fröhlich, Thomas Weber, Uwe Franke, Marc Pollefeys, and Matthias Rätsch. Multimodal neural networks: Rgb-d for

semantic segmentation and object detection. In *Scandinavian Conference on Image Analysis*, pages 98–109. Springer, 2017.

[222] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.

[223] Dayan Guan, Yanpeng Cao, Jiangxin Yang, Yanlong Cao, and Michael Ying Yang. Fusion of multispectral data through illumination-aware deep neural networks for pedestrian detection. *Information Fusion*, 50:148–157, 2019.

[224] Simon Chadwick, Will Maddern, and Paul Newman. Distant vehicle detection using radar and vision. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8311–8317. IEEE, 2019.

[225] Mario Bijelic, Tobias Gruber, Fahim Mannan, Florian Kraus, Werner Ritter, Klaus Dietmayer, and Felix Heide. Seeing through fog without seeing fog: Deep multimodal sensor fusion in unseen adverse weather. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11682–11692, 2020.

[226] Jaekyum Kim, Junho Koh, Yecheol Kim, Jaehyung Choi, Youngbae Hwang, and Jun Won Choi. Robust deep multi-modal learning based on gated information fusion network. In *Asian Conference on Computer Vision*, pages 90–106. Springer, 2018.

[227] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *Computer Vision and Pattern Recognition (CVPR), 2018 IEEE Conference on*. IEEE, 2018.

[228] Danfei Xu, Dragomir Anguelov, and Ashesh Jain. Pointfusion: Deep sensor fusion for 3d bounding box estimation. In *Computer Vision and Pattern Recognition (CVPR), 2018 IEEE Conference on*. IEEE, 2018.

[229] Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, and Steven L Waslander. Joint 3d proposal generation and object detection from view aggregation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–8. IEEE, 2018.

[230] Zining Wang, Wei Zhan, and Masayoshi Tomizuka. Fusing bird's eye view lidar point cloud and front view camera image for 3d object detection. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1–6. IEEE, 2018.

[231] Ming Liang, Bin Yang, Shenlong Wang, and Raquel Urtasun. Deep continuous fusion for multi-sensor 3d object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 641–656, 2018.

[232] Jian Dou, Jianru Xue, and Jianwu Fang. Seg-voxelnet for 3d vehicle detection from rgb and lidar data. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 4362–4368. IEEE, 2019.

[233] Ming Liang, Bin Yang, Yun Chen, Rui Hu, and Raquel Urtasun. Multi-task multi-sensor fusion for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7345–7353, 2019.

[234] Sourabh Vora, Alex H. Lang, Bassam Helou, and Oscar Beijbom. Pointpainting: Sequential fusion for 3d object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[235] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[236] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[237] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[238] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4. *International Journal of Computer Vision*, pages 1–26, 2020.

[239] Xinyu Huang, Peng Wang, Xinjing Cheng, Dingfu Zhou, Qichuan Geng, and Ruigang Yang. The apolloscape open dataset for autonomous driving and its application. *IEEE transactions on pattern analysis and machine intelligence*, 42(10):2702–2719, 2019.

[240] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4990–4999, 2017.

[241] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[242] Markus Enzweiler and Dariu M Gavrila. Monocular pedestrian detection: Survey and experiments. *IEEE transactions on pattern analysis and machine intelligence*, 31(12):2179–2195, 2008.

[243] Piotr Dollár, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian detection: A benchmark. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 304–311. IEEE, 2009.

[244] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2636–2645, 2020.

[245] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.

[246] Longyin Wen, Dawei Du, Zhaowei Cai, Zhen Lei, Ming-Ching Chang, Honggang Qi, Jongwoo Lim, Ming-Hsuan Yang, and Siwei Lyu. Ua-detrac: A new benchmark and protocol for multi-object detection and tracking. *Computer Vision and Image Understanding*, 193:102907, 2020.

[247] Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, et al. Argoverse: 3d tracking and forecasting with rich maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8748–8757, 2019.

[248] Yu-Dong Zhang, Lenan Wu, and Geng Wei. A new classifier for polarimetric sar images. *Progress In Electromagnetics Research*, 94:83–104, 2009.

[249] Gerard Margarit, Jordi J Mallorqui, and Xavier Fabregas. Single-pass polarimetric sar interferometry for vessel classification. *IEEE transactions on geoscience and remote sensing*, 45(11):3494–3502, 2007.

[250] Michal Shimoni, Dirk Borghys, Roel Heremans, Christiaan Perneel, and Marc Acheroy. Fusion of polsar and polinsar data for land cover classification. *International Journal of Applied Earth Observation and Geoinformation*, 11(3):169–180, 2009.

[251] J Scott Tyo, Bradley M Ratliff, and Andrey S Alenin. Adapting the hsv polarization-color mapping for regions with low irradiance and high polarization. *Optics letters*, 41(20):4759–4762, 2016.

[252] Samia Ainouz. *Analyse et traitement d'images multidimensionnelles de polarisation.* PhD thesis, Strasbourg 1, 2006.

[253] Amjad Almahairi, Sai Rajeswar, Alessandro Sordoni, Philip Bachman, and Aaron Courville. Augmented cyclegan: Learning many-to-many mappings from unpaired data. *arXiv preprint arXiv:1802.10151*, 2018.

[254] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, pages 6626–6637, 2017.

[255] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[256] Rachel Blin, Samia Ainouz, Stéphane Canu, and Fabrice Meriaudeau. Road scenes analysis in adverse weather conditions by polarization-encoded images and adapted deep learning. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 27–32. IEEE, 2019.

[257] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[258] Soufiane Belharbi, Clément Chatelain, Romain Hérault, Sébastien Adam, Sébastien Thureau, Mathieu Chastan, and Romain Modzelewski. Spotting l3 slice in ct scans using deep convolutional network and transfer learning. *Computers in biology and medicine*, 87:95–103, 2017.

[259] J. Nie, J. Yan, H. Yin, L. Ren, and Q. Meng. A multimodality fusion deep neural network and safety test strategy for intelligent vehicles. *IEEE Transactions on Intelligent Vehicles*, pages 1–1, 2020.

[260] Shuo Gu, Tao Lu, Yigong Zhang, Jose M Alvarez, Jian Yang, and Hui Kong. 3-d lidar+ monocular camera: An inverse-depth-induced fusion framework for urban road detection. *IEEE Transactions on Intelligent Vehicles*, 3(3):351–360, 2018.

[261] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-nms–improving object detection with one line of code. In *Proceedings of the IEEE international conference on computer vision*, pages 5561–5569, 2017.

[262] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.

[263] Kaiming He, Jian Sun, and Xiaoou Tang. Guided image filtering. In *European conference on computer vision*, pages 1–14. Springer, 2010.

[264] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1, pages 323–325. MIT press Cambridge, 2016.

[265] Huazhe Xu, Yang Gao, Fisher Yu, and Trevor Darrell. End-to-end learning of driving models from large-scale video datasets. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2174–2182, 2017.

[266] Rachel Blin, Samia Ainouz, Stephane Canu, and Fabrice Meriaudeau. A new multimodal rgb and polarimetric image dataset for road scenes analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 216–217, 2020.

Appendix A

Physical property $I_0 + I_{90} = I_{45} + I_{135}$

This appendix aims at showing that the intensities associated with a polarizer verify the physical constraint:

$$I_0 + I_{90} = I_{45} + I_{135} \ .$$

If I is a polarimetric image, it satisfies the following equation (see equations (1.7) and (1.5)):

$$I = A\tilde{A}I \ , \tag{A.1}$$

with:

$$A = \frac{1}{2}\begin{bmatrix} 1 & \cos(2\alpha_1) & \sin(2\alpha_1) \\ 1 & \cos(2\alpha_2) & \sin(2\alpha_2) \\ 1 & \cos(2\alpha_3) & \sin(2\alpha_3) \\ 1 & \cos(2\alpha_4) & \sin(2\alpha_4) \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \end{bmatrix} \ , \tag{A.2}$$

and:

$$\tilde{A} = (A^\top A)^{-1}A^\top = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 0 & -2 & 0 \\ 0 & 2 & 0 & -2 \end{bmatrix} \ . \tag{A.3}$$

By replacing the values of A (equation (A.2)) and $\tilde{A}$ (equation (A.3)) in equation (A.1):

$$\begin{bmatrix} I_0 \\ I_{45} \\ I_{90} \\ I_{135} \end{bmatrix} = \frac{1}{4}\begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \end{bmatrix}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 0 & -2 & 0 \\ 0 & 2 & 0 & -2 \end{bmatrix}\begin{bmatrix} I_0 \\ I_{45} \\ I_{90} \\ I_{135} \end{bmatrix} \ .$$

By denoting $M = A\tilde{A}$, equation (A.1) becomes:

$$I = MI \ ,$$

$$\begin{bmatrix} I_0 \\ I_{45} \\ I_{90} \\ I_{135} \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 3 & 1 & -1 & 1 \\ 1 & 3 & 1 & -1 \\ -1 & 1 & 3 & 1 \\ 1 & -1 & 1 & 3 \end{bmatrix} \begin{bmatrix} I_0 \\ I_{45} \\ I_{90} \\ I_{135} \end{bmatrix} .$$

Thus, the set I_{polar} of polarimetric images is:

$$I_{\text{polar}} = \{I | I = MI\} = \{I | (M - \text{Id})I = 0\} ,$$

that is:

$$I_{\text{polar}} = \text{Ker}(M - \text{Id}) .$$

Since:

$$M - \text{Id} = \frac{1}{4} \begin{bmatrix} -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} ,$$

I is a polarimetric image if:

$$\frac{1}{4} \begin{bmatrix} -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \begin{bmatrix} I_0 \\ I_{45} \\ I_{90} \\ I_{135} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} ,$$

that is $I_0 - I_{45} + I_{90} - I_{135} = 0$.

Appendix B

Details on literature

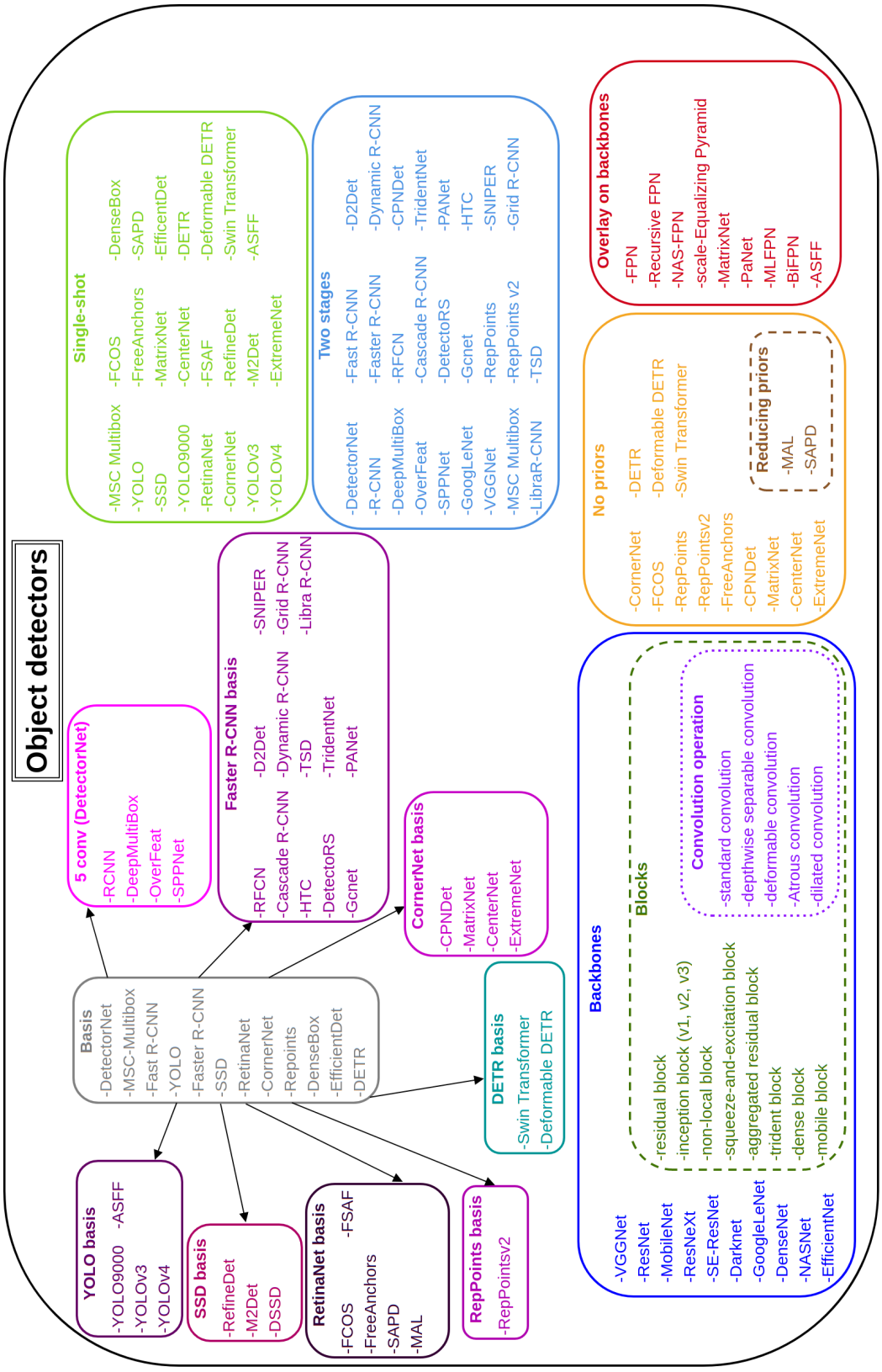

Figure B.1: Summary of object detectors and of their main properties.

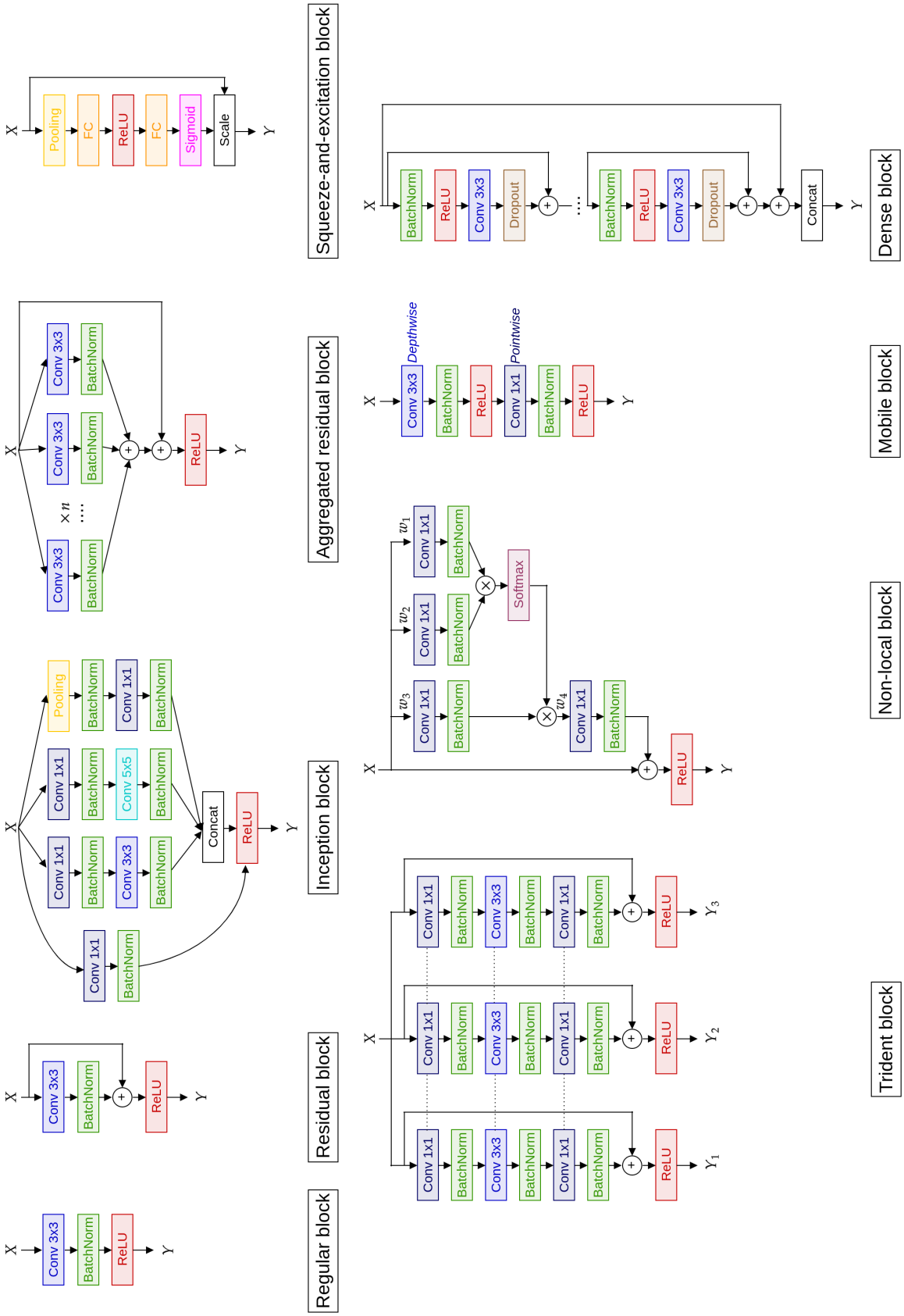

Figure B.2: Illustration of different convolutional blocks.

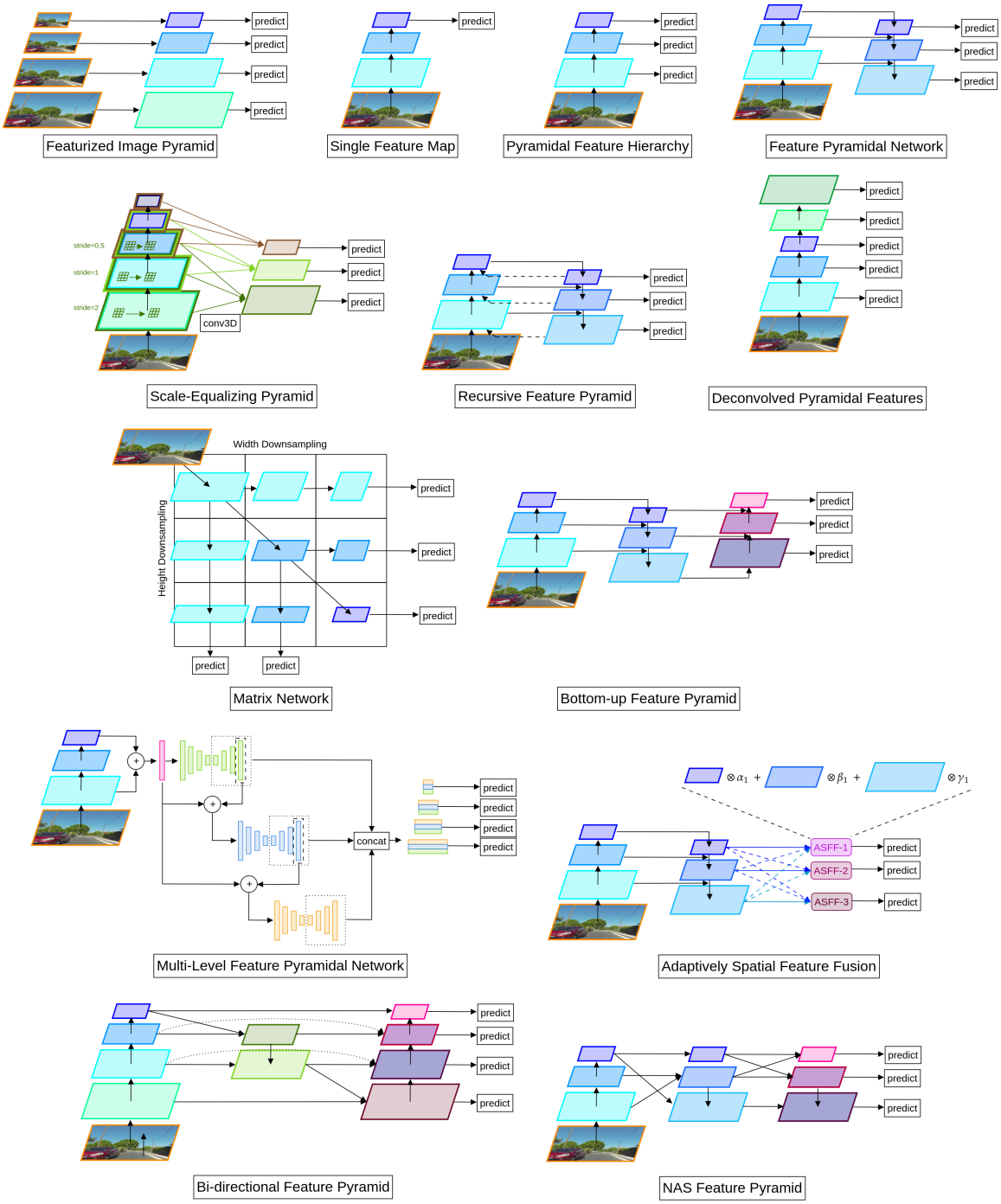

Figure B.3: Illustration of the different operations on the backbone's feature maps.

Appendix C

Generation of polarimetric images from the KITTI dataset

Figure C.1: Examples of generated polarimetric images from the KITTI dataset. From top to bottom, original image, I_0, I_{45}, I_{90} and I_{135}.

Figure C.2: Examples of generated polarimetric images from the KITTI dataset. From top to bottom, original image, I_0, I_{45}, I_{90} and I_{135}.

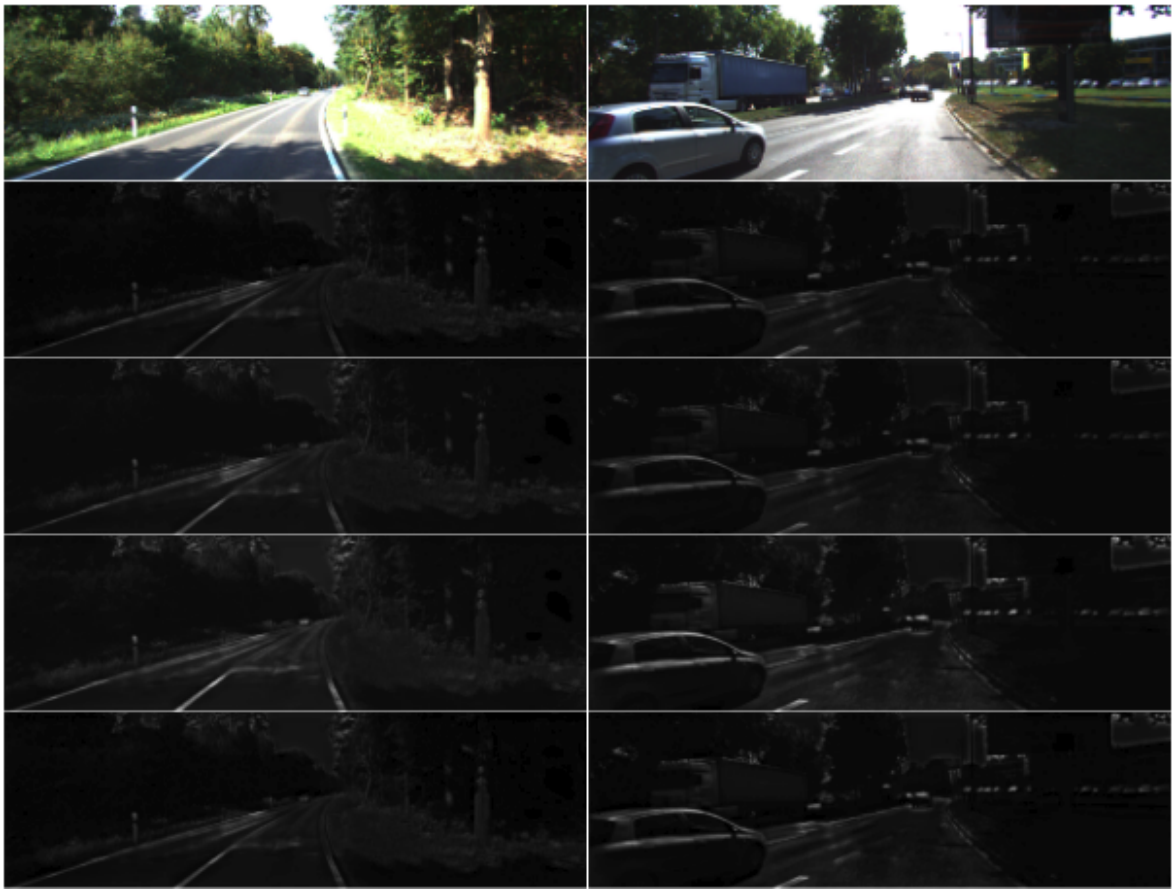

Figure C.3: Examples of generated polarimetric images from the KITTI dataset. From top to bottom, original image, I_0, I_{45}, I_{90} and I_{135}.