Towards Multi-Camera System for the Evaluation of Motorcycle Driving Test

Giuseppe Riccardo Leone, Marco Righi, Davide Moroni, Francesco Paolucci

Towards Multi-Camera System for the Evaluation of Motorcycle Driving Test

Giuseppe Riccardo Leone*, Marco Righi†, Davide Moroni*, and Francesco Paolucci‡
*Institute of Information Science and Technologies
National Research Council of Italy, Pisa, Italy
e-mail: *name.surname*@isti.cnr.it
†Institute of Information Science and Technologies
National Research Council of Italy, Pisa, Italy
e-mail corresponding author: marco.righi@isti.cnr.it
‡Consorzio Nazionale Interuniversitario per le Telecomunicazioni (CNIT), Pisa, Italy
e-mail: francesco.paolucci@cnit.it

***Abstract*—This work describes the early stage of an interactive and accelerated AI-driven framework for Practical Driving Courses and Driving Licence Exams. The core of the project is an innovative multi-parameter AI-assisted telemetry system able to compute test scores and outcome, useful for human-neutral auditability of Driving Licence Exams. The distributed Artificial Intelligence (AI) system available at the Track Testbed will be able to perform driving behaviour classifications and will suggest specific improvements based on the analysis of vehicle trajectories acquired during the driving test. Finally, the project will target the creation of a large dataset for driving test classification of key performance parameters. The system is envisioned to have a relevant impact on all the certification, driving licence operators and regulator entities.**

***Index Terms*—Camera-base systems, edge computing, trajectory analysis, computer vision, artificial intelligence, motorcycle driving test**

I. Introduction

The state-of-the-art of Internet of Things (IoT) sensors for driving/riding is significantly evolving in the last years, driven by the automotive ecosystem and next-generation autonomous driving applications, requiring a massive amount of online data to predict, compute and maintain optimal car/motorbike trajectories and behaviour. However, in the context of human learning and training driving phases for motorbikes, such instruments are currently not adequate to provide instructive feedback to anomalous, dangerous or inaccurate manoeuvres, since they are designed to interact with a computing system rather than humans. Since 2006 (Directive 2006/126/EC of the European Parliament on driving licences), very similar exam procedures have been defined in EU Countries (and worldwide). Nevertheless, there is a significant lack of standard, measurable and auditable means of verification for motorcycle practical examinations. For example, in 2021 in the province of Naples (Italy), only 1.9% of exam failures have been experienced, while more than 30% in the province of Cagliari (Italy). In Finland, it takes 3 years to get a full licence for motorcycles, with a minimum of 37 hours of driving, while in other EU Countries there is no minimum amount defined. There is a need for a system that provides reliable and rigorous means of evaluations, guaranteeing fair and equitable treatments. The AI-RIDE project chases this goal, presenting the adoption of an accelerated, online and embedded Artificial Intelligence framework in the context of motorcycle rider training, particularly targeting the Practical Driving Courses (PDC) and Driving Licence Exam (DLE) sessions verification tools. The project will target a disruptive innovation step in the context of driving learning techniques, significantly going beyond the state-of-the-art of current instruments used in the PDC and DLE ecosystem. In addition, the Project will enable the definition of a reliable scoring system, overcoming the current basic pass/fail system while providing useful and measurable indications of typical exam errors and unsafe driving procedures. The outcome of a driving licence exam depends on several factors, such as the performance time, the trajectory precision, the speed management, the driver's body posture and the motorbike position. Most of such factors' performance may be computed using data analytic tools resorting to video frames from external cameras; video cameras installed along the circuit track may provide information to AI systems to

Fig. 1. The motorcycle driving licence test takes place on predefined paths delimited with traffic cones. Original image from [1]

extract information about position, speed, and trajectory. For the Italian law [2], the driving licence exam consists of three phases: the first two aimed at demonstrating driving skills on a "safe" circuit closed to traffic, and, only after passing these steps, the final test on a public road. Our research project focuses on the first two tests with a known *a priori* scenario where it is possible to put cameras and sensors suitable for the automatic evaluation of the candidate's driving skills. In particular, these tests take place on predefined paths with standard measures delimited with traffic cones (see Fig. 1): one path is smaller with close passages and low speed and the other is larger and requires higher speeds. On both of them there is a slalom at the beginning, then a curve, and finally a straight section. On the long track, the vehicle must be stopped in a specific space. Everything must be done respecting time constraints.

The tests must be carried out without committing penalties that otherwise jeopardise the success of the exam. These penalties are:

- irregularly coordinate driving, demonstrating poor skill;
- touch one or more cones;
- skip a cone during the slalom phase or exit the course;
- put one foot on the ground;
- (long track only) stop the motorcycle with the front wheel which has not passed the first alignment or which has passed the second alignment;
- take more than 25 seconds to complete the long track;
- take less than 15 seconds to complete the short track;

In the following sections of this paper we will show how we designed the system to be able to recognise each penalty and perform the outcome of a driving licence exam. In Section II we talk about the state-of-the-art of technologies that our system will rely on. In Section III we describe the logic that we are planning to use to deal with every single point of the penalties list. In Section IV we show the preliminary tests and results behind the choice of the hardware and software components of the system. Finally, in Section V we summarise what we have done till now and the future work we are planning to do.

II. Related work

Although the announcement reported in [3] is discussing some generic AI-based automated procedures under implementation in Singapore and expected to be available in 2024, to the best of our knowledge, no complete solutions based on AI and computer vision are available on the market or have been reported in the scientific literature to automate training and testing procedures for motorbike driving lessons and exams. Nevertheless, our system will rely on and put together some functionalities widely used in the computer vision domain: object detection, background subtraction and tracking.

A. Object detection

In our scenario, we have only three objects to recognise: the cones, the motorcycle and the person on top of it. Several methods based on deep learning have been proposed in the literature for this task. Conventionally, two main types of networks for object detection can be identified: two-stage ones and single-stage ones. The former divide the detection task into two steps: first a set of candidate regions corresponding to sub-windows of the image in which it is likely there is an object of interest are generated and, then, these regions are tested using architectures borrowed from the object classification task producing in output a vector of size equal to the number of C classes to be analysed. The i-th entry of this vector corresponds to the probability that in the region there is an object of class i ($0 < i < C$). Among the best-known methods are Region-Based Convolutional Network (R-CNN) [6], which allows locating objects by training a model using a small amount of annotated data; Spatial Pyramid Pooling (SPP) net, Fast R-CNN [7], which guarantees better mean average precision than the previous ones and Faster R-CNN [8] that generates region proposals in a less expensive way than both R-CNN and Fast R-CNN. On the other hand, single-stage networks simultaneously produce candidate regions and their classification, often efficiently representing an *a priori* number of detectable objects with certain predetermined scales and aspect ratios, called anchors. The actual positioning and size of the regions represented by an anchor are obtained by refining the initial data of the anchor by estimating appropriate offset factors by means of regression. Among them, Single Shot MultiBox Detector (SSD) [9], which is based on a convolutional feed-forward network capable of producing a set of predetermined cardinality of bounding boxes, each characterised by a vector of confidence values about the presence of objects inside it. Then, a non-maximum suppression procedure allows for identifying a subset of these bounding boxes, corresponding to the objects in the image. The first layers of the SSD network are based on standard architectures used for the classification of images (truncated, however, before any classification level), layers which are generally referred to as the backbone. Auxiliary levels are then added to produce, also by regression, refinements on the position and size of the bounding boxes. A further well-known method is constituted by YOLO (You Look Only Once) [10]; after the first publication in 2016 many improved versions have been presented and since YOLOv3 [11] the algorithm is computationally suitable for real-time (can even reach 155

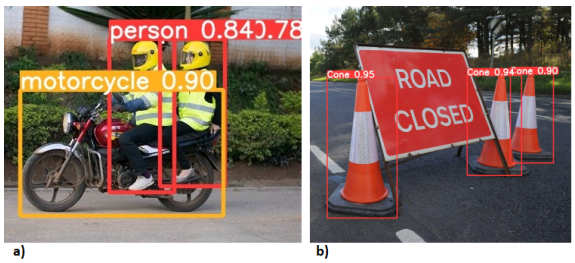


Fig. 2. Motorcycles, persons and cones detection with Yolov5; a) original image from [4] b) original image from [5]

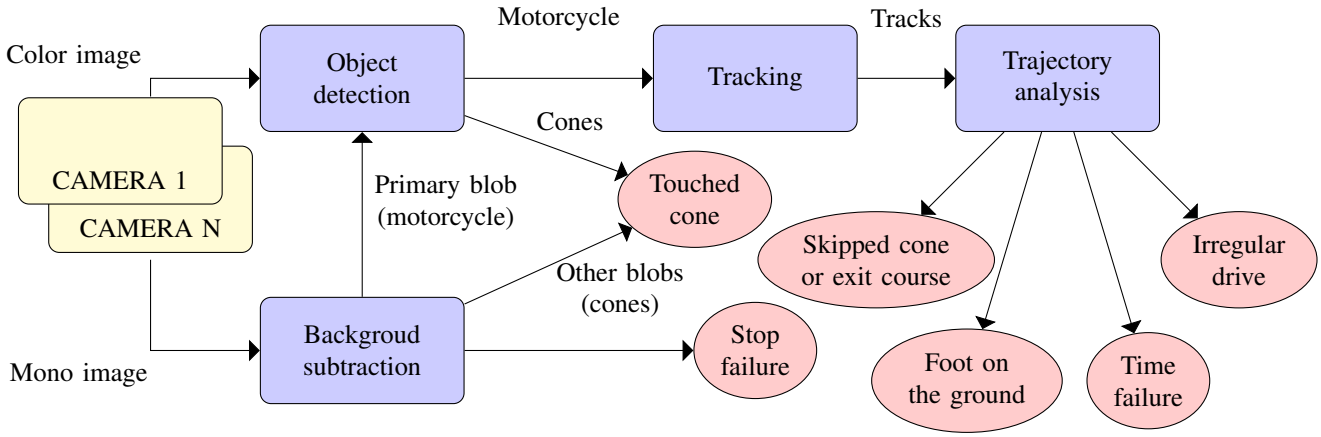


Fig. 3. The processing pipeline: the blue rectangles are the image processing steps, while the red ovals indicate the penalties we are looking for.

fps). Nowadays YOLOv5 [12] is the most used and supported detector by the computer vision community. In Fig. 2 you can see a couple of examples that could be useful for this research. The authors of YOLOv7 [13] [14] state that its performances surpass all known object detectors in both speed and accuracy in the range from 5 FPS to 160 FPS. Single-stage methods have been appreciated for their computational convenience, which makes them an ideal choice for real-time applications. However, the recorded performances have often been lower than with two-stage methods. To this end, many recent works, such as [15], [16], have tried to bring some successful features of two-stage detectors into single-stage detectors, without altering their computational convenience.

B. Background subtraction

Background subtraction is one of the fundamental image processing tasks frequently used in applications such as video surveillance, human activity recognition and autonomous navigation. It might be regarded as a binary image segmentation aimed at separating the background from the foreground, usually consisting of moving objects of interest. Among the classical methods for achieving background subtraction, we mention Gaussian Mixture Models GMM [17] and *codebooks* [18], both of which require a small number of video frames to understand the statistics both of foreground and background pixels. Other low complexity methods have been tested in specific scenarios, such as traffic and parking monitoring [19], [20]. The advent of deep learning has given some new development also in Background Subtraction (BGS). For instance, in [21], semantic segmentation extracted at a lower pace by using deep learning paradigms is integrated into a real-time BGS achieving state-of-the-art performance. Similarly, other supervised approaches both *video-agnostic* and *video-group-optimized* have been proposed [22] taking advantage of spatial and temporal information that is processed thanks to end-to-end convolutional neural networks. In our scenario, videos are from one know category and, therefore, video-group-optimized algorithms are the natural candidates for achieving BGS which might be beneficial for detecting possible contacts between the bikers and the cones as well as for assessing the area of final stopping and checking it is within the first and second alignments.

C. Tracking

Single-camera tracking means assigning an identification number (ID) to all the moving elements present in a given frame and recognising the same subjects in the following frames by carrying forward the assigned IDs; the image coordinates of the tracked object are projected at ground level $(z = 0)$ to obtain the bi-dimensional (x, y) position; the temporal sequence of these coordinates is the track of that moving object. In our scenario, there is only one object to track, the motorcycle with the rider on top of it. In this case, it is possible to use a robust background subtraction algorithm to achieve very good results because there are no occlusions along the track. Nevertheless, deep learning has made it possible to obtain great performances also in tracking [23], [24]. A reference site that shows the results of the best algorithms is [25]: the winner of the CVPR19 competition proposed in [26] is still the top algorithm today. It uses a detection based on Faster R-CNN and a similarity estimation based on features of size 128 produced by the patches of the converted image in the HSV color space. In the second place, there is [27] whose code in open-source format is freely usable and modifiable.

III. Methods

In this section, we describe the logic that we are planning to use to compute the overall score of the observed driving test. The basic idea relies on the assumption that the motorcycle is the only moving element in the scene. Every penalty occurs always nearby the motorcycle (touching or skipping a cone, driver's foot on the ground, failing the stop in the designed area). With background subtraction, we can easily identify

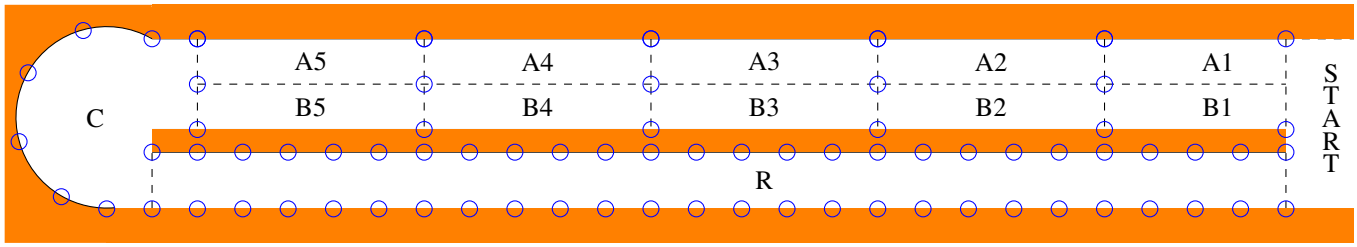


Fig. 4. Skipped cone or exit the course: the track is divided into virtual zones to understand if the path follows the right sequence or it the motorcycle enters in any forbidden area.

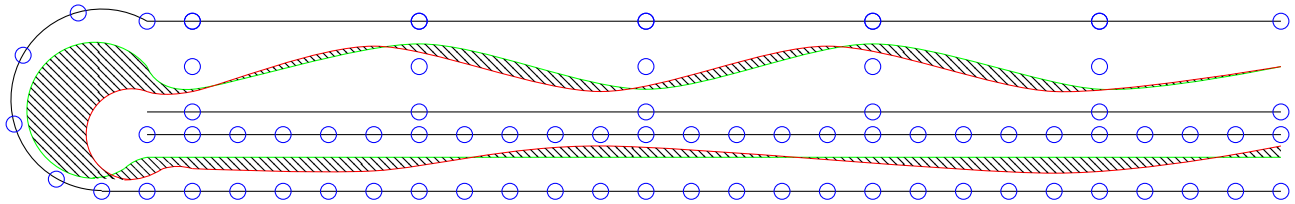


Fig. 5. Trajectories analysis: the blue symbols are the cones, the green line is the reference path, the red line is the path of the test drive, the northwest lines the areas containing the error

the area of interest to analyse in the current frame. In this way, there is no need to process the whole image but only a small part of it. The area should be big enough to include the relevant scenario around the motorbike, like the nearby cones or the circuit edges. Fig. 3 depicts the main processing pipeline applied to the video stream of each camera. The blue rectangles are the image processing steps of the pipeline, while the red ovals indicate the penalties we are looking for. The Trajectory analysis module is build on top of the Tracking one to implement custom logic according to the specific penalty definition. As soon as a single penalty is detected the exam ends with a failure. Only irregular driving is not connected to a specific event but it is decided by the human examiner. Relying on the computer vision techniques described in section II, in the following paragraphs we will discuss how we deal with every single point of the penalties list.

A. Touching a cone

Background subtraction is the perfect technique to discover this penalty: when a cone's position changes over time after it has been touched also the background changes correspondingly. This is the flag to start the verification procedure. As you can observe in Fig. 3 the *Touched cone* oval has input from the *Background subtraction* module, but also from the *Object detection* one. This is to be sure that we are dealing with a cone and to avoid false positives due to other unexpected moving elements (e.g. cat or dog on the circuit). A very tough situation is when the cone is slightly touched and it does not change position after this event. We will dedicate particular attention to this case.

B. Putting a foot on the ground

When there is an unexpected stop during the test and the rider puts the foot on the ground this is confirmed by the motorcycle's speed which goes down to zero and this is a parameter that *Trajectory analysis* can easily measure. Unfortunately, there is also the case when the rider slightly touches the ground without stopping the vehicle, because he/she is losing balance. This is maybe the harder task to deal with only external cameras. Due to the perspective and the high speed of the gesture, it can be very hard also for the human eye to understand if the foot has actually touched the ground. We will try some experiments with a dedicated camera at ground level, but the presence of the cones should create many visual occlusions in this setup. As further work, we plan to compute the driver pose estimation that could be useful for such a task.

C. Skipping a cone or exiting the course

This penalty is flagged by the *Trajectory analysis* module. The track and the surrounding area are divided into virtual zones as shown in Fig. 4: if the driver starts at the right of the first cone the correct path is determined by the sequence START - A1 - A2 - B2 - B3 - A3 - A4 - B4 - B5 - A5 - C - R - START; the examiner often allows to start also at the left of the first cone. In this case the correct sequence is START - B1 - B2 - A2 - A3 - B3 - B4 - A4 - A5 - B5 - C - R - START. With the virtual zone approach, it is straightforward to accept both possible sequences that depend on the side chosen at the beginning of the test. Furthermore, it is relatively easy to understand if the driver exits from the course. The edges

of the circuits are also virtual delimiters and an outside zone is a forbidden place (the filled area in Fig. 4); as soon as the motorcycle's position is in that area the penalty is flagged.

D. Driving irregularly or showing poor skill

Some interesting works has been made about the assessment of the car driving style [28]. In our case the test has two main factors to consider: the spatial precision and the temporal/spatial precision of the driver. We provide three independent *scores* to measure the quality of the trajectory followed by the candidate biker.
The first step of the system setup requires the computation of the reference (or optimal) trajectory. The reference trajectory can be computed as follow:

- by the analysis of some drives performed by one or more driving instructors;
- directly by humans choice settings.

The points of optimum trajectory are projected in a three-dimensional space and each point is characterised by three attributes:

- x: the x position projected in the bi-dimensional space of the camera
- y: the y position projected in the bi-dimensional space of the camera
- t: the time the motorbike is in the point (x, y), the time is expressed in milliseconds.

We transform the (x, y, t) coordinate space into $(\overline{x}, \overline{y}, t)$ where $(\overline{x}, \overline{y})$ is the rectified space. The rectification transforms the camera space into a metric space. This transformation permits us to have an absolute measure of track in terms of kinematics. The score related to the simply spatial position is computed by subtracting the calculus of two areas. As shown in Fig. 5, we can easily calculate the area of the trajectory considering it as a closed path by connecting its start and end point. Equation 1 describes the calculus of the area.

$$A_t = u \cdot \forall m, n \in I_g, \sum_{\{x_m, y_n\}} 1 \tag{1}$$

where Ig is the area delimited by the perimeter of the closed path and u is the size in meters of each pixel in the rectified space. $^{\mathrm{A_t}}/_{\mathrm{u}}$ is simply the number of the pixels in the rectified space. We consider now A_{ref}. We calculate A_{ref} on the reference trajectory. We compare now A_{t_i} with A_{ref}. For each i-th test dive is calculated a new A_{t_i}. The score "a", (S^a) is computed as described in equation 2.

$$S^a_{t_i} = [A_{ref} - (A_{ref} \bigcap A_{t_i})] + [A_{t_i} - (A_{ref} \bigcap A_{t_i})] \tag{2}$$

The best score is zero.
The aim of the next score considers the spatial/temporal data and provides an analysis of the harmoniously of the driving. The analysis of spatial/temporal data is more complicated because the various sets of acquired data do not have synchronisation. We focus our interest on the speed and acceleration of the motorbike. In fact, we label each point of the reference trajectory (considering the rectified space) with its proper speed and relative acceleration. The test drive has to respect some parameters and too much high speed or acceleration has to be considered a penalty. In practice this score is composed of various attributes described by equations 3, 4, 5 and 6.
In order to perform this calculus, we reshape [29] the shortest acquired data or the data of the reference trajectory to the longer one. This elaboration permits us to have two arrays aligned in the space from the beginning of the test to the end of the test and to have a relation with the time that is used to calculate kinematic features.
Let's consider:

- $Speed_{rp} = r^s_j$ is the reshaped speed array of the reference trajectory
- $Speed_{td} = t^s_j$ is the reshaped speed array of the test drive

By reshaping arrays we reshape time values too so we can easily calculate the acceleration by dividing the speed by the time.

- $Acc_{rp} = r^a_j$ is the reshaped acceleration array of the reference trajectory
- $Acc_{td} = t^a_j$ is the reshaped acceleration array of the test drive

Let's define N equal to the length of the defined arrays. The score is composed of these values:

$$S^c_{speed-max} = max\left(r^s_j - t^s_j\right) \quad \forall i = j \in \{0, \ldots, N)\} \qquad [^{\mathrm{m}}/_{\mathrm{s}}] \tag{3}$$

$$S^c_{acc-max} = max\left(r^a_j - t^a_j\right) \quad \forall i = j \in \{0, \ldots, N)\} \qquad [^{\mathrm{m}}/_{\mathrm{s}^2}] \tag{4}$$

The computation of the equations provides an average behaviour and uses the RMS definition [30] on the arrays previously defined.

$$S^c_{speed-rms} = \sqrt{\frac{1}{N} \cdot \sum_{j=0}^{N} \left(r^s_j\right)^2} - \sqrt{\frac{1}{N} \cdot \sum_{j=0}^{N} \left(t^s_j\right)^2} \qquad [^{\mathrm{m}}/_{\mathrm{s}}] \tag{5}$$

$$S^c_{acc-rms} = \sqrt{\frac{1}{N} \cdot \sum_{j=0}^{N} \left(r^a_j\right)^2} - \sqrt{\frac{1}{N} \cdot \sum_{j=0}^{N} \left(t^a_j\right)^2} \qquad [^{\mathrm{m}}/_{\mathrm{s}^2}] \tag{6}$$

The last parameter we are going to evaluate is the Hausdorff distance [31] [32] among the reference trajectory and the actual trajectory during a test drive.

E. Stopping outside the target area

In the long track test, it is mandatory to stop the motorcycle with the front wheel inside a narrow target area of 0.5m delimited by four cones. If the front wheel has not passed the first alignment or if it has passed the second alignment there is a failure penalty. We think that a dedicated low-cost camera and a solid background subtraction method should be enough to verify the position of the wheel in the designed area,

as depicted in Fig. 3 where the *Stop Failure* oval is connected only with the *Background subtraction* module.

F. Time constraints

There is a penalty if the driver takes more than 25 seconds to accomplish the long track or less than 15 seconds for the short one. The task consists in comparing the travel time of the test with these time constraints. The start signal is manually given by the examiner that pushes the button to start the official chronometer. For the stop signal, we use two different strategies depending on the test we are evaluating. For the long track test, we already analysed if the motorcycle has correctly stopped in the designed area (see the previous paragraph). If this test has been successful we can consider as the stop timestamp the first frame where the motorcycle is not moving. For the shorter track, the motorcycle should not stop at the end but we can still use the virtual zones depicted in Fig. 4. The moment when the motorcycle returns correctly in the START area represents the stop timestamp.

IV. Materials

The activities of this project started less that a month ago, at the beginning of August 2022. Therefore it is too early to provide quantitative test results; in this section we describe the very first object detection experiments based on the videos taken at the Testbed track, with the goal of choosing the number, the positions and the features of the cameras to cover the two circuits and to be able to deal with all the problems that we described in section III.

A. The data acquisition set up

We made a couple of acquisition days at the Testbed Track located in Pontedera during some students' training. As you can see in Fig. 6 the camera positions are marked by the label CX where X is the number of the camera. All the cameras were placed at the height of 6 meters and there is a total of 6 cameras: 4 for the short track in every direction and 2 for the long one only at the end and the beginning of it. This is not the final number of cameras, we will need more on the side of the long track. This is just a sub-minimal layout to begin with the research. We used two different cameras and they are shown by different colours in Fig. 6: the green is related to the short track only and it is made with a low-cost Full HD webcam (resolution 1920x1080 at 30 fps), while the red indicates the GoPro Hero 10 action camera. The last one is a very interesting commercial product because it is designed mainly for sport activities outdoor and therefore it is waterproof and stabilised, it can record many hours of video standalone, it can transfer data via wifi, and, above all, it offers very high resolution and frame rate capabilities: it is possible to record video up to 240 fps shooting with 2704×1520 or 1920×1080 resolution and 120 fps with the 4K resolution (3840x2160). An industrial camera with the same capabilities has a price ten times the GoPro price. Unfortunately, we discovered a big downside: the device is too powerful and, after 4-5 minutes of stand-still recording, has overheating issue and turns itself off. That day was very hot with a temperature of 37° Celsius. We made a lot of tries also indoors at 25° Celsius and the problem happened again after 8-9 minutes of recording. Clearly, this cannot be the camera we need in the final set-up, but still, it is very useful for some video sessions to experiment with different resolutions and frame rates.

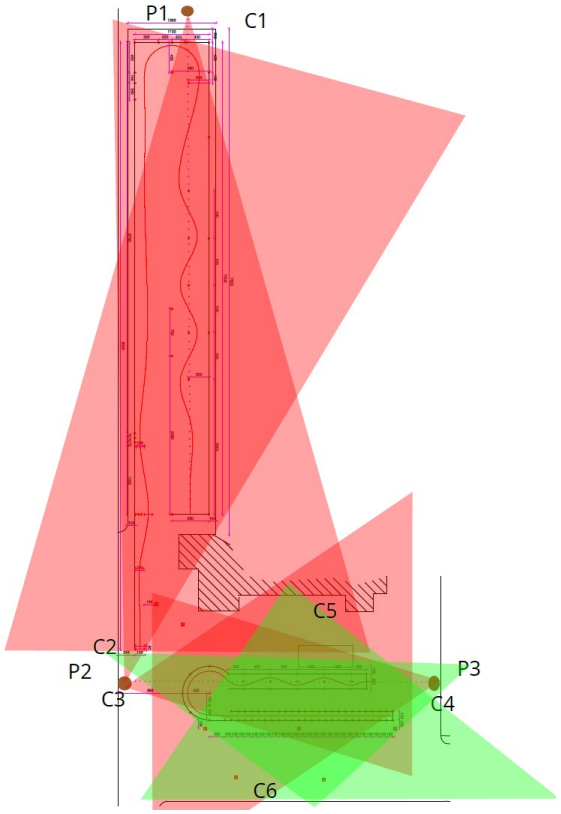
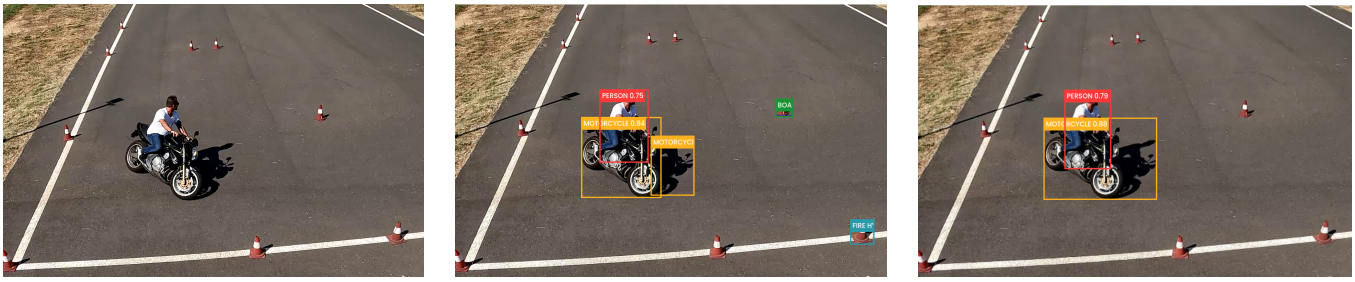


Fig. 6. The map of the Pontedera Testbed Track and the positions of the six cameras used during the acquisition days. The label CX indicates the position of camera X, while PX points show where to install the supporting poles for the cameras

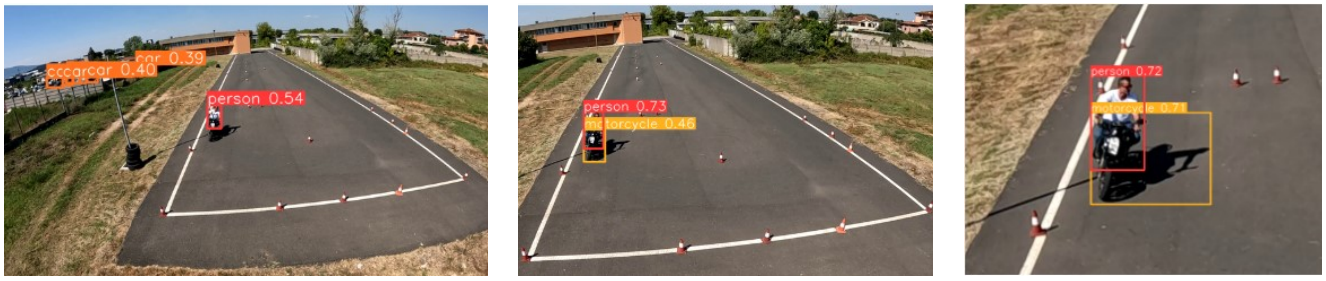
B. Early object detection experiments

In this research, we have only a few elements to detect: the pilot, the motorbike and the traffic cones. We already shown in Fig. 2 some examples. As previously described in section II we focus our attention on the YOLO object detector for its good real-time performance and the wide support from the computer vision community. Several interesting tools provide a YOLO approach to the faced problems. We do not intend to provide a YOLO review, our focus is to put in evidence the tools useful to our theoretical and experimental. For these reasons, we will cite only some tools that provide a well-documented API and a stable source code usable on the Linux OS. OpenVINO [33] is a tool provided by Intel provided with the step-by-step demo, python tutorials, DL Workbench and samples. It reports numerous examples using Docker container [34] or the programming language Python [35]. Open Model Zoo [36] for OpenVINO toolkit delivers a wide variety of free, pre-trained deep learning models and demo applications that provide full



(a) (b) (c)

Fig. 7. First results with YOLOv5: (a) source image, (b) one false positive due to the shadow and two classification errors for the cones; (c) correct result



(a) (b) (c)

Fig. 8. The smaller the better: (a) poor results on original image 2704x1520 - process time 20 sec, (b) medium precision with partial crop 1731x1133 - process time 10 sec, (c) good results on smaller crop 543x442 - process time 4 sec

application templates to help you implement deep learning in Python, C++, or OpenCV Graph API (G-API) [37]. Models and demos are available in the Open Model Zoo GitHub repo [36] and licensed under Apache License Version 2.0. OpenVINO is provided with super-resolution libraries based on An Attention-Based Approach for Single Image Super Resolution paper [38]. The use of super-resolution algorithms could be very important in our context due to the wide areas to cover, the high frame per second required by our objectives, and the need to keep the costs of the devices down. On the other hand, we try to minimise the computation time. During deep experimentation, we will decide which is the best strategy.

The Ultralytics [39] online framework presents interesting functionalities that allow testing the capabilities of YOLOv5 [40]. We used this tool to make some object detection tests and the results are shown in Fig. 7. In the middle picture, there are two different types of error: the bad classification and the false positive. For the bad classification of the cones there is nothing to worry about, because the cones class it is not in the standard default model used by YOLO. This problem will be solved by upgrading the existing model with the cone class or, maybe better, building a custom dataset containing only the three classes of this domain. The false positive is actually the shadow of the motorcycle which is recognised as a second motorcycle. Also in this case there is no big deal: it happens only in some frames and we already know that there is only one motorcycle on the circuit. This new detection will be discarded by the tracking algorithm which will choose only the element coherent with the previous frame.

Another interesting result is shown in Fig. 8: starting from the original image (a) and then selecting part of it (b) or just the interesting area we get an expected faster response due to smaller resolution, but also better performance of the detection results. This confirms the idea that we should select the area of interest around the motorcycle with background subtraction and then perform object detection only on this small image.

V. Conclusions and Further Work

AI-Ride is a research project based on computer vision and artificial intelligence techniques, that aims to build an interactive and accelerated AI-driven framework for Practical Driving Courses and Motorcycle Driving Licence Exams. In this paper we focus on the multi-parameter AI-assisted telemetry system able to compute test scores and outcomes of driving test, useful for human-neutral auditability. The distributed AI multi camera system will be able to recognised all the penalties that jeopardise a driving licence exams by means of a computer vision logic that process the streams from the cameras. It will also perform driving behaviour classifications and will suggest specific improvements based on the analysis of vehicle trajectories acquired during the driving test. A lot of work will be done in the next months: we showed the very first experiment we made at the Track Testbed in Pontedera, mainly to decide how many cameras we need to cover all the circuits and the technical characteristics to achieve the best results. The system relies on background subtraction, object detection and tracking to solve all the user cases. Nevertheless, some penalty like *Foot on the ground* is very tough to discover. Human pose estimation could be useful for this task; both the posture of

the rider and the motorcycle can add valuable information in the evaluation of the driver's skill.

ACKNOWLEDGMENT

This work is partially supported by the VEDLIoT project funded by the European Union's Horizon 2020 research and innovation programme under grant agreement No 957197. Special thanks to Filippo Cugini (Consorzio Nazionale Interuniversitario per le Telecomunicazioni, Pisa, Italy) and Marco Abbondandolo (Autoscuola Gerardo snc, Pontedera, Italy) for the support during the data acquisition days.

REFERENCES

[1] Patente.it, "Percordi d'esame motocicli," http://www.patente.it/img/tabelloni/47-Percorso-d'esame-motocicli.jpg, 2022.

[2] Ministero dei Trasporti, "Decreto 26/09/2018 - prove di valutazione per conseguimento patenti a1, a2 e a," https://www.gazzettaufficiale.it/eli/id/2018/10/12/18A06493/sg, 2018, last reviewed September 26, 2022.

[3] Motopinas, "Singapore using ai for motorcycle driving tests," https://www.motopinas.com/motorcycle-news/singapore-using-ai-for-motorcycle-driving-tests.html, 2021.

[4] A. Fredrick, "Object detection with yolov5 and pytorch," https://www.section.io/engineering-education/object-detection-with-yolov5-and-pytorch/, 2021.

[5] I. Katsamenis, E. Karolou, A. Davradou, E. Protopapadakis, A. Doulamis, N. Doulamis, and D. Kalogeras, "Tracon: A novel dataset for real-time traffic cones detection using deep learning," https://github.com/ikatsamenis/cone-detection, 05 2022.

[6] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *CVPR*, 2014.

[7] R. Girshick, "Fast r-cnn," in *ICCV*, 2015.

[8] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE TPAMI*, vol. 39, no. 6, 2017.

[9] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *ECCV*, 2016.

[10] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.

[11] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv*, 2018.

[12] G. J. et al., "Yolov5 classification models, apple m1, reproducibility, clearml and deci.ai integrations," https://zenodo.org/record/7002879#.YwUUqHZByUk, 2022.

[13] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," 2022. [Online]. Available: https://arxiv.org/abs/2207.02696

[14] "Yolo darknet," (visited on 21 August 2022). [Online]. Available: https://github.com/pjreddie/darknet

[15] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li, "Single-shot refinement neural network for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4203–4212.

[16] X. Yang, J. Yan, Z. Feng, and T. He, "R3det: Refined single-stage detector with feature refinement for rotating object," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 4, 2021, pp. 3163–3171.

[17] Z. Zivkovic, "Improved adaptive gaussian mixture model for background subtraction," in *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, vol. 2. IEEE, 2004, pp. 28–31.

[18] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, "Real-time foreground–background segmentation using codebook model," *Real-time imaging*, vol. 11, no. 3, pp. 172–185, 2005.

[19] M. Magrini, D. Moroni, G. Palazzese, G. Pieri, G. Leone, and O. Salvetti, "Computer vision on embedded sensors for traffic flow monitoring," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*. IEEE, 2015, pp. 161–166.

[20] G. Amato, P. Bolettieri, D. Moroni, F. Carrara, L. Ciampi, G. Pieri, C. Gennaro, G. R. Leone, and C. Vairo, "A wireless smart camera network for parking monitoring," in *2018 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2018, pp. 1–6.

[21] A. Cioppa, M. Van Droogenbroeck, and M. Braham, "Real-time semantic background subtraction," in *2020 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2020, pp. 3214–3218.

[22] M. O. Tezcan, P. Ishwar, and J. Konrad, "Bsuv-net 2.0: Spatio-temporal data augmentations for video-agnostic supervised background subtraction," *IEEE Access*, vol. 9, pp. 53 849–53 860, 2021.

[23] G. Ciaparrone, F. L. Sánchez, S. Tabik, L. Troiano, R. Tagliaferri, and F. Herrera, "Deep learning in video multi-object tracking: A survey," *Neurocomputing*, vol. 381, 2020.

[24] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *ICIP*, 2017.

[25] MOT, "Multiple object tracking benchmark," https://motchallenge.net, 2022.

[26] D. Mykheievskyi, D. Borysenko, and V. Porokhonskyy, "Learning local feature descriptors for multiple object tracking," in *ACCV*, 2020.

[27] X. Zhou, V. Koltun, and P. Krähenbühl, "Tracking object as points," in *ECCV*, 2020.

[28] B. Jachimczyk, D. Dziak, J. Czapla, P. Damps, and W. J. Kulesza, "Iot on-board system for driving style assessment," *Sensors*, vol. 18, no. 4, 2018. [Online]. Available: https://www.mdpi.com/1424-8220/18/4/1233

[29] M. Bonakdarpour, S. Chatterjee, R. F. Barber, and J. Lafferty, "Prediction rule reshaping," 2018. [Online]. Available: https://arxiv.org/abs/1805.06439

[30] R. C. Gonzalez and R. E. Woods, *Digital image processing*, 3rd ed. Upper Saddle River, N.J.: Prentice Hall, 2008. [Online]. Available: http://www.amazon.de/Digital-Image-Processing-Rafael-Gonzalez/dp/013168728X/ref=sr_1_6?s=books-intl-de&ie=UTF8&qid=1330928076&sr=1-6

[31] O.-K. Kwon, D.-G. Sim, and R.-H. Park, "New Hausdorff distances based on robust statistics for comparing images," in *Proceedings of 3rd IEEE International Conference on Image Processing*, vol. 3. IEEE, Sep. 1996, p. 21–24 vol.3. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=560359;https://ieeexplore.ieee.org/document/560359/

[32] N. Tran, B. Rohrer, and S. Warnick, "Alignment Distance in Path Control," in *2007 IEEE International Conference on Control Applications*. IEEE, Oct. 2007, p. 1468–1473. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4389443;https://ieeexplore.ieee.org/document/4389443/

[33] "Openvino documentation," (visited on 18 August 2022). [Online]. Available: https://docs.openvino.ai/latest/index.html

[34] "Docker," (visited on 18 August 2022). [Online]. Available: https://www.docker.com

[35] "Python," (visited on 18 August 2022). [Online]. Available: https://www.python.org/

[36] "Openvino open model zoo," (visited on 18 August 2022). [Online]. Available: https://github.com/openvinotoolkit/open_model_zoo

[37] "Opencv graph api," (visited on 18 August 2022). [Online]. Available: https://docs.opencv.org/4.x/d0/d1e/gapi.html

[38] Y. Liu, Y. Wang, N. Li, X. Cheng, Y. Zhang, Y. Huang, and G. Lu, "An attention-based approach for single image super resolution," 2018. [Online]. Available: https://arxiv.org/abs/1807.06779

[39] "Ultralytics framework," (visited on 21 August 2022). [Online]. Available: https://www.ultralytics.com/

[40] "Yolo icevision," (visited on 21 August 2022). [Online]. Available: https://github.com/airctic/yolov5-icevision