Waseda at TRECVID 2015: Semantic Indexing

Kazuya Ueki and Tetsunori Kobayashi

Faculty of Science and Engineering, Waseda University,
Room 40-701, Waseda-machi 27, Shinjuku-ku, Tokyo, 162–0042 Japan
{ueki,koba}@pcl.cs.waseda.ac.jp

Abstract. Waseda participated in the TRECVID 2015 Semantic Indexing (SIN) task [6]. For the SIN task, our approach used the following processing pipelines: feature extraction using several deep convolutional neural networks (CNNs); classification of the presence or absence of a detection target by support vector machines (SVMs); and fusion of multiple score outputs. In order to improve the performance of semantic video indexing, we employed the following techniques: utilizing multiple evidences observed in each video and compressing them into a fixed-length vector; introducing gradient and motion features to CNNs; enriching variations of the training and the testing sets; and extracting features from several CNNs trained with various large-scale datasets. Through these techniques, our best run achieved a mean Average Precision (mAP) of 30.9%. This was ranked 2nd among all the participants.

1 System Description

Our method consists of three steps: extracting features with CNNs, classifying the presence or absence of a detection target, and fusing multiple classifiers.

Almost all methods submitted in the past several years have been a fusion of various feature extraction methods. These methods include conventional image classification methods (such as local feature extraction and encoding), acoustic feature extraction (with the mel-frequency cepstral coefficient), motion feature extraction (according to dense trajectories [7]), and deep learning (with CNNs).

Our CPU and memory resources are limited; we only have two off-the-shelf computers. Fortunately, these computers each have two Titan Black GPUs; therefore we decided to focus on extracting features only from CNNs rather than extracting other types of features (either local image features such as SIFT or motion features such as dense trajectories).

1.1 Feature extractions with CNNs

We used AlexNet [4], a CNN structure proposed at ILSVRC 2012 (ImageNet Large-Scale Visual Recognition Challenge). AlexNet contains five convolutional layers and three fully-connected layers as shown in Fig. 1. With our proposed method, the original image is first inputted into the network. Features are then extracted from the hidden layers (the 6th and 7th layers) and the output layer (the 8th layer). The features extracted from the 6th, 7th, and 8th layers are 4,096, 4,096, and 1,000 dimensions, respectively.

Instead of using local features such as SIFT or HOG, or motion features such as dense trajectories, we used several different CNNs to obtain complementary features. The following six types of CNNs were used in our semantic indexing system.

1. `ImageNet`

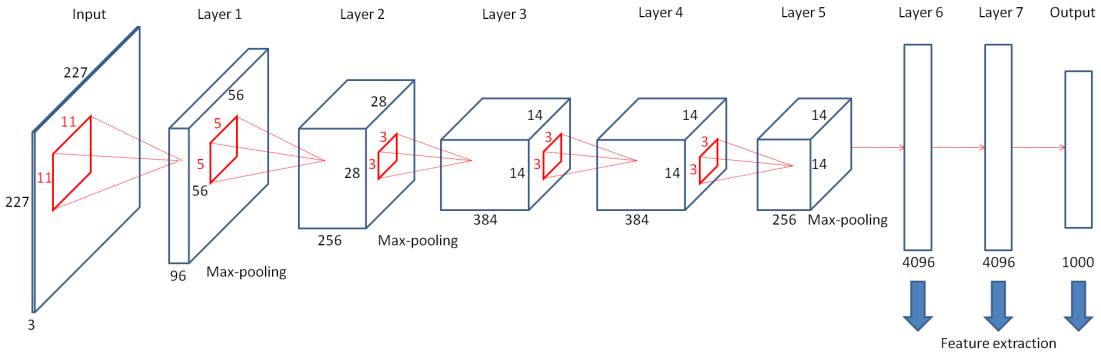

Fig. 1. CNN structure used in the feature extraction. The number of units in the output layer is different depending on individual models.

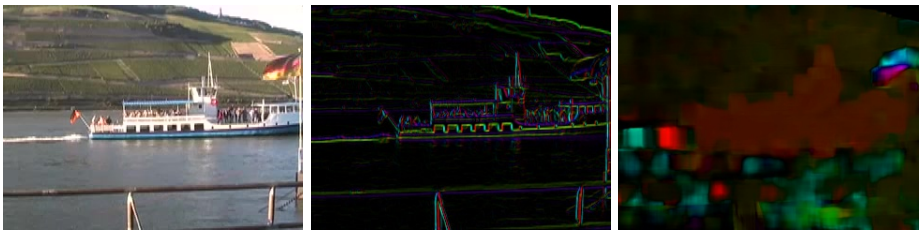

Fig. 2. Examples of gradient and optical flow images. Left: original image. Center: gradient image. Right: optical flow image.

The `ImageNet` model is provided with the Caffe (CNN) library [2]. This network was trained with the ImageNet dataset containing 1.2 million images and 1,000 categories and was used in the ILSVRC2012 benchmark.

2. `Finetune`
   We created this new CNN by fine-tuning the `ImageNet` model for TRECVID SIN task; we named this CNN the `Finetune` model. The `Finetune` model was trained with one million keyframe images from TRECVID videos. These keyframe images were randomly selected from positive shots of 346 concepts. We used the annotation provided by collaborative annotation [1]. The number of concepts annotated to TRECVID videos is 346, consequently, the number of units in the final output layer of `Finetune` model was changed to 346.
3. `Gradient`
   SIFT and HOG features are commonly used and are very effective for video retrieval. Both features compute the edge gradient, with which the object's contour can be enhanced significantly. To substitute edge features with CNN features, we first applied a Sobel filter to the images and created **gradient images**, as shown in the middle of Fig. 2. In these images, the color corresponds to the orientation, and the brightness corresponds to the magnitude of the orientation gradients. We then trained a new CNN using one million gradient images which were created from the original keyframe images. The feature vectors' dimension extracted from the output layer is also 346.
4. `OpticalFlow`
   Motion is a powerful feature for action recognition and video retrieval, and dense trajectories are one of the best features in TRECVID SIN task [3]. However, due to our limited computational resources, we decided to substitute CNN features for motion features. When using dense trajectories, the optical flow is first calculated,

before pooling the features spatially and temporally. Thus, we focused on creating **optical flow images**, as shown in the right of Fig. 2, by calculating the optical flow between two consecutive frames. In these images, the color corresponds to the orientation of the optical flow, and the brightness corresponds to the magnitude of the optical flow. As before, these images were then used to train a new CNN. The number of units in the 8th layer of CNNs is also 346.

5. `Places`
   Scene understanding has an important role for semantic video indexing; some of TRECVID's concepts are related to particular scenes (e.g., "Beach", "Nighttime", "Stadium"). Therefore, we utilized a pretrained scene recognition model: the Places205-AlexNet model (`Places`) provided by MIT Computer Science and Artificial Intelligence Laboratory [8], which is publicly available at the Model Zoo [5]. The `Places` model was trained on 205 scene categories with 2.5 million images. The feature vectors' dimension extracted from the output layer is 205.
6. `Hybrid`
   We employed another pretrained model, the Hybrid-AlexNet model (`Hybrid`), which is also provided by MIT Computer Science and Artificial Intelligence Laboratory [8]. The `Hybrid` model was trained on 1,183 categories (205 scene categories and 978 object categories) with 3.6 million images. The feature vectors' dimension extracted from the output layer is 1,183.

1.2 Feature Pooling

To employ multiple observations in each shot, we utilized multiple frames and compressed them into a fixed-length vector. We theorized that the use of multiple frames would offer a performance improvement, because different types of features can be extracted from different angles of a detection target. We first selected a maximum of 10 frames from a shot at regular intervals. After selecting these 10 frames, the corresponding 10 images were inputted to the CNN to obtain the respective 10 feature vectors. These 10 feature vectors were then bound to one feature vector by element-wise max-pooling. That is, the values of the elements in the same dimension were compared across 10 sets, and the maximum value was selected. For example, when extracting multiple features from the 6th layer of the CNN, we can create one 4,096-dimensional feature vector from multiple 4,096-dimensional vectors. This fixed-length vector must include the information from multiple frames. Max pooling is employed because higher values are returned when an input image has more distinctive features.

1.3 Classification with SVMs

After extracting features from the 6th, the 7th, and the 8th (output) layer of CNNs, the SVMs were trained using the extracted feature vectors. To reduce the computational cost and the toll on memory resources, we did not concatenate the feature vectors of different layers. Rather, we created a separate SVM using features from each layer.

To train the SVMs, we approximately used 30,000 shots for each concept, with roughly the same number of positive and negative samples. There were an insufficient number of positive samples for most concepts, therefore we used more negative samples than positive samples. After evaluating all of the testing samples, we ranked the top 2,000 samples based on their respective SVM scores.

We also considered ways to enrich the variations of the training and the testing sets. Our approach utilized flipped images during both the training and testing of the SVMs. This is similar to the data augmentation used to train CNNs effectively [4]. There were far fewer positive samples than negative samples, therefore we used the

flipped images exclusively for the positive samples. Finally, we decided to evaluate the testing set (IACC_2_C) in the following ways:

- Original images used for both training and testing.
- Both original and flipped images used for training, but only original images used for testing.
- Both original and flipped images used for training, and only flipped images used for testing.

For testing flipped images, we inputted the flipped images into the CNN in the same manner that we inputted the original images when extracting features from the hidden layers. We generated scores using these three methods, and subsequently integrated the results using late fusion.

1.4 Fusion

Finally, all of the results from different CNNs and their layers were combined using score-level fusion. This involves the summation of the scores from the various SVMs to obtain the final results. For weight optimization, we employed several methods, as we will discuss in the next section.

2 Submissions

2.1 Submitted Runs

We submitted the following four runs to the TRECVID 2015 SIN task. For these runs, we used all scores obtained from each layer of six CNNs, however, we changed the fusion weights as described below.

- **Waseda4 (priority 4)**
  We calculated the total scores by summing their weighted scores. We set a different weight for each CNN model in an ad hoc way. Specifically, we used a weight of 2 for `ImageNet`, `Finetune`, `Places`, and `Hybrid` models, and 1 for `Gradient` and `OpticalFlow` models, because we considered the original images to contain more information than gradient and optical flow features.
- **Waseda3 (priority 3)**
  There are 51 classifiers for each concept. We set 51 different weights which were shared across all concepts. We started from the weights used in the Waseda4 run and then slightly modified each weight to improve the mAP of 30 concepts on development set. Most of the weights were close to the weights of Waseda4 run.
- **Waseda2 (priority 2)**
  We set the weights in the same manner as the Waseda3 run, however we adjusted the weights to improve the mAP of 60 concepts on development set.
- **Waseda1 (priority 1)**
  Weights were adjusted in a similar manner to the Waseda2 and the Waseda3 runs, but were optimized to improve the average precision of each concept on development set.

Table 1. The mAPs for individual models with the TRECVID 2015 SIN testing set.

Model	Layer	Train: original images	Train: original + flipped images	
		Test: original images	Test: original images	Test: flipped images
`ImageNet`	6	24.02	24.14	23.75
	7	23.61	23.89	23.53
	8	18.82 [1]	19.08 [1]	18.70 [1]
`Finetune`	6	23.50	23.80	23.84
	7	23.29	23.39	23.44
	8	21.53	21.90	21.78
`Gradient`	6	20.74	19.41	19.03
	7	19.82	18.95	19.17
	8	17.71	17.26	17.35
`OpticalFlow`	6	14.21	14.43	13.99
	7	13.22	13.34	13.42
	8	13.12	13.43	13.56
`Places`	6	23.40	23.61	23.74
	7	22.29	22.41	22.20
	8	— [2]	— [2]	— [2]
`Hybrid`	6	25.12	24.75	24.34
	7	25.52	25.17	24.79
	8	23.20	22.93	22.88

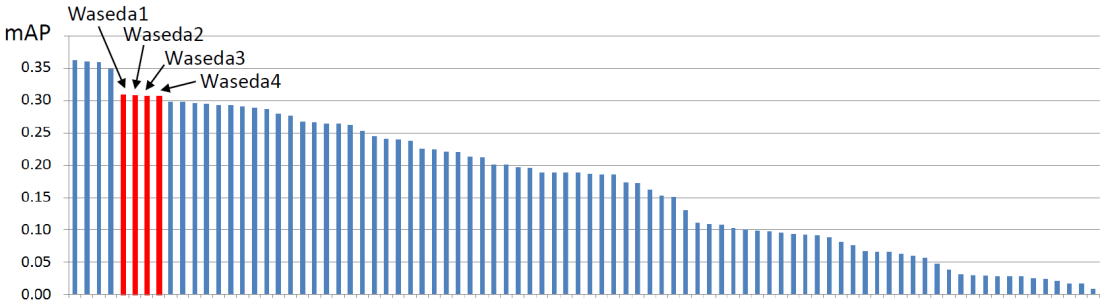

Fig. 3. Comparison of Waseda runs with the runs of other teams on IACC_2_C.

2.2 Results

The mAPs for individual models are shown in Table 1. From the table, it can be inferred that the `Hybrid` was the best feature extraction model, and the `Gradient` and `OpticalFlow` models were not good enough. However, the mAP was significantly improved by combining all models.

Fig. 3 shows the results of all runs. Our 2015 submissions ranked between 5 and 8 in a total of 86 runs. Our best run was an mAP of 30.9%, which ranked 2nd among all participants.

Fusion weight optimization did not offer significant improvements over averaging of scores.

Fig. 4 shows the average precision for each semantic concept. One of our runs achieved the best average precision for some concepts: "Cheering", "Demonstration_Or_Protest",

[1] This result includes some errors. After rectifying the errors, the mAPs were changed to 22.04, 22.20, and 21.74, respectively.

[2] We could not finish the calculation by the submission deadline. After the submission, we evaluated the performance and found that the mAPs were 19.73, 19.86, and 19.59, respectively.

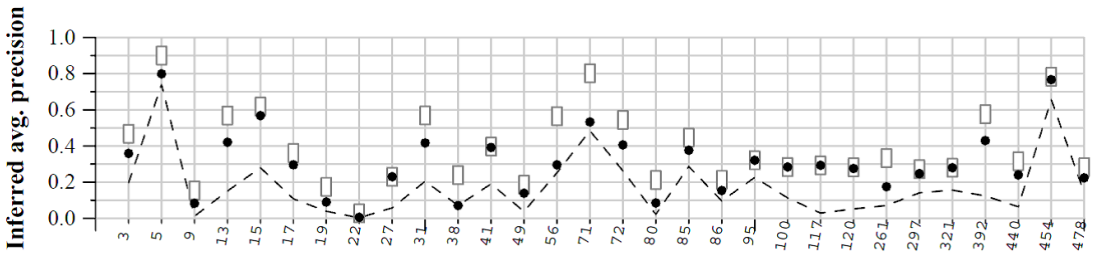

Fig. 4. Average precision of our best run (Waseda1) for each semantic concept.

"Press_Conference", "Running", "Telephones", "Throwing", and "Lakes". We deduce that the motion feature works well for concepts such as "Cheering", "Running", and "Throwing", and feature extraction from the scene classification CNN (the `Gradient` and `OpticalFlow` models) is valuable for concepts such as "Press_Conference" and "Lakes".

3 Conclusion

We used multiple CNNs to achieve feature extraction, and we combined them to improve semantic video indexing. Despite the simplicity of our method, it achieved relatively high performance. However, the performance of semantic video indexing was still extremely low. The method of the winning team achieved a mAP of 36.2%. In the future, we will investigate the root causes of this poor performance and evaluate the options for improving it.

4 Acknowledgements

This work was supported by JSPS KAKENHI Grant Number 15K00249.

References

1. S. Ayache and G. Quénot. Video corpus annotation using active learning. In *In 30h European Conference on Information Retrieval (ECIR ' 08)*, pages 187–198, 2008.
2. Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.
3. L. Jiang, X. Chang, Z. Mao, A. Armagan, Z. Lan, X. Li, S. Yu, Y. Yang, D. Meng, P. Duygulu-Sahin, and A. Hauptmann. CMU Informedia@TRECVID 2014 semanctic indexing. In *TRECVID 2014*, 2014.
4. A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25:1106–1114, 2012.
5. Model zoo. https://github.com/BVLC/caffe/wiki/Model-Zoo/.
6. P. Over, G. Awad, M. Michel, J. Fiscus, W. Kraaij, A. F. Smeaton, G. Quéenot, and R. Ordelman. TRECVID 2015 – An overview of the goals, tasks, data, evaluation mechanisms and metrics. In *Proceedings of TRECVID 2015*. NIST, USA, 2015.
7. H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Dense trajectories and motion boundary descriptors for action recognition. *Int. J. Comput. Vis.*, 103(1):60–79, 2013.
8. B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *NIPS*, 2014.