IMPROVING SEMANTIC VIDEO INDEXING: EFFORTS IN WASEDA TRECVID 2015 SIN SYSTEM

Kazuya Ueki, Tetsunori Kobayashi

Faculty of Science and Engineering, Waseda University

ABSTRACT

In this paper, we propose a method for improving the performance of semantic video indexing. Our approach involves extracting features from multiple convolutional neural networks (CNNs), creating multiple classifiers, and integrating them. We employed four measures to accomplish this: (1) utilizing multiple evidences observed in each video and effectively compressing them into a fixed-length vector; (2) introducing gradient and motion features to CNNs; (3) enriching variations of the training and the testing sets; and (4) extracting features from several CNNs trained with various large-scale datasets. Using the test dataset from TRECVID's 2014 evaluation benchmark, we evaluated the performance of the proposal in terms of the mean extended inferred average precision measure. On this measure, our system's performance was 35.7, outperforming the state-of-the-art TRECVID 2014 benchmark performance of 33.2. Based on this work, our submission at TRECVID 2015 was ranked second among all submissions.

Index Terms— Semantic video indexing, video search, CNN, TRECVID, generic object recognition

1. INTRODUCTION

In this paper, we focus on the problem of semantic video indexing— i.e., the automatic assignment of semantic tags to video segments. Semantic video indexing is especially useful for video filtering, searching, and browsing.

Recently, convolutional neural networks (CNNs) have been widely used for automatic video analysis tasks, and have consistently outperformed conventional methods. Our research likewise uses CNN-based feature extraction, but we present a method for improving the performance of this method. We undertook the following four efforts:

1. Using multiple observations in each video and compressing them into a fixed-length vector.
2. Introducing gradient and motion features to CNNs.
3. Enriching variations of the training and testing sets by using flipped images during both the training and the testing phases.
4. Utilizing other CNNs that were pretrained with various large-scale image datasets.

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

To evaluate the proposed approach, we used TRECVID's 2014 benchmark data to facilitate a comparison with state-of-the-art methods.

2. RELATED WORK

TRECVID is an annual benchmarking conference [1, 2] organized by the National Institute of Standards and Technology. At TRECVID, participants work on common tasks using a common dataset and scoring metrics, such that a comparison can be made when judging the performance of the various methods proposed by the participants. TRECVID includes several tasks every year, but the one most germane to our research is Semantic INdexing (SIN) task. Almost all proposed methods consist of a fusion of various feature extraction methods [3, 4, 5, 6, 7]. These methods include conventional image classification methods (local feature extraction followd by pooling), acoustic feature extraction (with the mel-frequency cepstral coefficient), motion feature extraction (according to the dense trajectory [8]), and deep learning (with CNNs).

However, because a tremendous amount of video is needed to be trained and tested, the computational costs associated with such tasks lead to considerable bottlenecking. Thus, large-scale computing resources (i.e., supercomputers) are needed, especially when high-dimensional feature vectors are used.

Several CNN implementations with graphical processing units (GPUs) have recently been made available to the public—e.g., Theano [9], cuda-convnet [10], OverFeat [11], and Caffe [12]. These implementations are considerably useful when training a large number of samples. In this paper, therefore, we focus on the ability of the CNN, exclusively using CNN features, rather than motion features such as the dense trajectory, or local features such as a scale-invariant feature transform (SIFT) or a histogram of oriented gradients (HOG).

3. SYSTEM PERSPECTIVE

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

We used AlexNet [10], a CNN structure proposed at ILSVRC 2012. AlexNet contains five convolutional layers and three fully-connected layers. With our proposed method, the original image is first inputted to the network. Features are then extracted from the hidden layers (the 6th and 7th

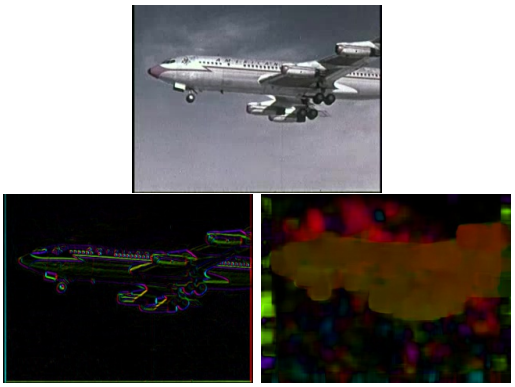

Fig. 1. Examples of gradient and optical flow images. Top: original image. Bottom-left: gradient image. Bottom-right: optical flow image.

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.

Next, SVMs are trained using the extracted feature vectors. To reduce the computational cost and the toll on memory resources, we do not concatenate the feature vectors of different layers. Rather, we create a separate SVM using features from each layer.

Finally, all of the results are combined. This involves simply adding the multiple scores from the SVMs to obtain the final results.

4. PROPOSED METHOD

4.1. 1st effort: Using multiple evidences observed in each video

Features can only be obtained from a single frame in a video. However, the use of multiple frames can offer an improvement to the performance, because different types of features can be extracted from different angles of a detection target. We first select at most N frames from a video at regular intervals. After selecting these N frames, the corresponding N images are inputted to the CNN to output the respective N feature vectors. These N feature vectors are then bound to one feature vector by element-wise max-pooling. That is, the values of the elements in the same dimension are compared across N sets, and the maximum value is 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 conducted because higher values are returned when an input image has more distinctive features.

4.2. 2nd effort: Introducing gradient and motion features to the CNNs

Using both CNN features and SIFT or dense trajectory features and integrating them with the CNN method can boost the performance of semantic video indexing [3, 4, 5, 6, 7]. This is because complementary features can be obtained using different feature extraction methods. However, after the pooling stage (with bag-of-features (BoF) or Fisher vectors), we are faced with very high-dimensional vectors. To avoid this, we substitute SIFT or dense trajectory features with CNN features.

SIFT and HOG are used to compute the edge gradient, with which the object's contour can be enhanced significantly. To substitute edge features with CNN features, we apply a Sobel filter to the images and create **gradient images**, as shown in the bottom-left of Fig. 1. In these images, the color corresponds to the orientation, and the brightness corresponds to the magnitude of the orientation gradients. We then use these images to train a new CNN.

We also substitute motion features with CNN 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 bottom-right of Fig. 1, 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 are then used to train the new CNN.

4.3. 3rd effort: Enriching variations of the training and the testing sets

Data augmentation is often used to train CNNs effectively [10]. Therefore, to enrich variations of the training and the testing sets, our approach utilizes flipped images, during both the training and testing of the SVMs. In general, there are far fewer positive samples than negative samples. Consequently, we used the flipped images for the positive samples exclusively. For testing, we simply inputted the flipped images to the CNN in the same way that we inputted the original images when extracting features from the hidden layers.

4.4. 4th effort: Utilizing CNNs pretrained with various kinds of data

Currently, many annotated images are well organized as a result of the emergence of crowd-sourcing platforms. This has helped in the creation of superior recognition models. Other databases have also been developed. At Model Zoo [16], for instance, several state-of-the-art pretrained CNNs using the Caffe toolbox have been made publicly available. Our proposed approach involves using multiple CNNs trained with various large-scale datasets to extract complementary features, in order to improve the performance of semantic video indexing.

We experimented with two training methods for multiple CNNs. The first method was to train a model using only the target data (viz., the TRECVID videos). The second involved using CNNs trained with other datasets. Because the concepts we need to detect comprise not only objects, but also scenes and events, we selected two types of so-called Places-CNNs [13]: the Places205-AlexNet model, which was trained on 205 scene categories with 2.5 million images; and

Table 1. 30 concepts evaluated in the TRECVID 2014 SIN task.

Airplane	Motorcycle
Basketball	News_Studio
Beach	Nighttime
Bicycling	Running
Boat_Ship	Singing
Bridges	Stadium
Bus	Telephones
Chair	Baby
Cheering	Flags
Classroom	Forest
Computers	George_Bush
Demonstration_Or_Protest	Lakes
Hand	Oceans
Highway	Quadruped
Instrumental_Musician	Skier

the Hybrid-AlexNet model, which was trained on 1,183 categories (205 scene categories and 978 object categories) with 3.6 million images.

5. EXPERIMENTS

5.1. Database

We evaluated the performance of the proposed approach on the TRECVID video dataset from TRECVID's 2014 SIN task. The TRECVID dataset was collected from the Internet Archive. Therefore, it contains a wide variety of objects, scenes, and events. Videos typically consist of multiple shots that are divided using automatic shot-boundary detection. Labels are provided to some (but not all) of the shots with collaborative annotation [14, 15]. The average length of each video shot is approximately 5.4 seconds. The TRECVID 2014 dataset includes 106,913 testing shots and more than 500,000 training shots (approximately 800 hours of video).

5.2. Evaluation criteria

The purpose of the proposed approach is to detect specific semantic concepts—e.g. objects, scenes, and events—in the testing videos. We used the same evaluation criteria as the TRECVID 2014 SIN task, namely by measuring the mean extended inferred average precision (MAP). At TRECVID 2014, participants evaluated the entire testing set (106,913 shots), outputted their scores, and submitted lists of the top 2,000 shots corresponding to 60 concepts. Finally, 30 of these 60 concepts were evaluated, as shown in Table 1. For the evaluation, the top pool sampled 100% of the shots ranked 1–200 across all submissions, and the bottom pool randomly sampled 11.1% of the shots ranked 201–2000. Human judges then assessed these pools in order to generate a truth judgment. In our experiments, this truth judgment was used to provide a fair comparison of the performance of the respective methods.

5.3. Experimental conditions

Our implementation of the CNNs is based on the publicly available Caffe toolbox [12]. To train the SVMs, we used approximately 30,000 shots for each concept, with roughly the same number of positive and negative samples. Indeed, there are an insufficient number of positive samples for most concepts. Therefore, we used more negative samples, rather than positive samples. Finally, after evaluating all of the testing samples, we ranked the top 2,000 samples based on their respective SVM scores.

Table 2. MAP values for different CNN models and their combinations with the TRECVID 2014 SIN dataset.

Model	`ImageNet`			`Gradient`			`OpticalFlow`			MAP
Layer	6	7	8	6	7	8	6	7	8	
Single classifier	✓									26.94
		✓								27.06
			✓							25.73
				✓						23.00
					✓					23.17
						✓				21.74
							✓			15.07
								✓		14.76
									✓	13.12
Multiple classifiers	✓	✓	✓							28.49
	✓	✓	✓	✓	✓	✓				30.25
	✓	✓	✓				✓	✓	✓	30.11
	✓	✓	✓	✓	✓	✓	✓	✓	✓	**30.97**

5.4. Experimental results with regard to the 1st effort

To determine the effectiveness of using multiple frames, we compared the performance between the use of a single frame and the use of at most 10 frames. We extracted features from the 6th, 7th, and 8th layers of a CNN trained with ImageNet. With only a single frame, the MAP values were 20.83 (6th layer), 20.10 (7th layer), and 19.41 (8th layer). With multiple frames, by contrast, the MAP values were 26.94 (6th layer), 27.06 (7th layer), and 25.73 (8th layer). Thus, utilizing multiple frames consistently improved the performance by at least six MAP points.

5.5. Experimental results with regard to the 2nd effort

In order for the CNN to be inputted with both the gradient images shown in the bottom-left of Fig. 1 and the optical flow images shown in the bottom-right of Fig. 1, we newly trained CNNs using gradient and optical flow images, respectively. The network structure used was similar to the AlexNet structure. In our network, however, the output layer contained 346 units, such that it was equivalent to the number of concepts in TRECVID. For the sake of clarity, the CNN trained with the ImageNet dataset is written as `ImageNet`, in bold Typewriter font. Likewise, the CNN trained with the TRECVID gradient images is written `Gradient`, and the CNN trained with the TRECVID gradient images is written `OpticalFlow`.

Experimental results for `ImageNet`, `Gradient`, and `OpticalFlow` are summarized in Table 2. The MAP values for `Gradient` and `OpticalFlow` are not as high as the one for

Table 3. MAP values for individual models with the TRECVID 2014 SIN dataset.

Model	Layer	Train: origignal images	Train: original + flipped images	
		Test: original images	Test: original images	Test: flipped images
`ImageNet`	6	26.94	27.35	27.11
	7	27.06	27.93	27.82
	8	25.73	26.27	26.01
`Gradient`	6	23.00	21.46	21.65
	7	23.17	22.25	23.12
	8	21.74	21.06	21.99
`OpticalFlow`	6	15.07	15.37	15.41
	7	14.76	15.10	15.13
	8	13.12	13.43	13.56
`Finetune`	6	27.24	27.73	27.44
	7	27.56	28.44	28.14
	8	26.57	27.19	26.94
`Places`	6	27.80	27.89	27.84
	7	26.90	27.51	27.68
`Hybrid`	6	29.51	29.14	28.98
	7	29.19	29.44	29.35
	8	27.91	27.90	27.84

`ImageNet`. However, from this experiment we can see that the performance improves by the fusion of complementary features, namely color, gradient, and motion features (see Section 5.8 for details about the fusion method).

5.6. Experimental results with regard to the 3rd effort

We evaluated the effectiveness of utilizing flipped images by analyzing features from the 6th layer of the `ImageNet` model. We adapted this during both the training and the testing phases. The performance after training the SVMs with only the original images was compared with that from using both the original and flipped images, resulting in MAP values of 26.94 and 27.35, respectively.

By contrast, when flipped images were used during the testing phase, the MAP was 26.18 using the SVM trained only with original images, and 27.11 using the SVM trained with both original and flipped images.

After combining these four types of classifiers, we found that the best MAP (27.99) was obtained by combining three types of classifiers, except when original images were used during the training phase and flipped images were used during the testing phase. To reduce the computational expense, we decided to perform calculations using only these three types of classifiers for the other CNNs.

5.7. Experimental results with regard to the 4th effort

In order to obtain complementary features, we not only extracted from the `ImageNet`, `Gradient`, and `OpticalFlow` models, but also from other CNNs trained with different large-scale image datasets. First, we created a new CNN (`Finetune`) by fine-tuning the `ImageNet` model on TRECVID data. This `Finetune` model was trained with 1 million images in 346 categories. These images were extracted from the positive shots of TRECVID training videos.

Additionally, scene recognition is important for semantic video indexing, because some concepts in the TRECVID dataset are related to particular scenes (e.g., Beach, Nighttime, Stadium, etc). Therefore, as mentioned in the previous section, we utilized two pretrained models provided at the Model Zoo [16]: the Places205-AlexNet model (`Places`) and the Hybrid-AlexNet model (`Hybrid`).

The MAP values for each individual model are shown in Table 3.

5.8. Fusing all of the classifiers

Finally, we integrated all the classifiers shown in Table 3. To do so, we calculated the total scores by simply summing their weighted scores. We used a weight of 2 for `ImageNet`, `Finetune`, `Places`, and `Hybrid`, and 1 for `Gradient` and `OpticalFlow`, because the original images contain more information than gradient and optical flow features.

When combining all of the features, we achieved a MAP of 35.69, which is significantly better than 33.2, achieved by the winning team at TRECVID 2014. Finally, we submitted our system to TRECVID 2015 SIN task. Our submission was ranked second among all 29 teams.

6. SUMMARY AND FUTURE WORKS

In this paper, we showed that extracting the complementary features from several CNNs is considerably effective for semantic video indexing. Our proposed approach achieved a state-of-the-art performance without needing to combine conventional image classification methods (e.g., SIFT with BoF) and motion features (e.g., the dense trajectory). In future research, we shall study the use of other CNNs, especially those with a very deep structure, and we shall train these CNNs using large-scale datasets in order to improve their performance.

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

7. REFERENCES

[1] A. F. Smeaton, P. Over, and W. Kraaij, "Evaluation campaigns and TRECVid," in *MIR '06: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, New York, NY, USA, 2006, pp. 321–330, ACM Press.

[2] P. Over, G. Awad, M. Michel, J. Fiscus, G. Sanders, W. Kraaij, A. F. Smeaton, and G. Quéenot, "TRECVID 2014 – An overview of the goals, tasks, data, evaluation mechanisms and metrics," in *Proceedings of TRECVID 2014*. NIST, USA, 2014.

[3] B. Safadi, N. Derbas, A. Hamadi, M. Budnik, P. Mulhem, and G. Quénot, "LIG at TRECVid 2014: Semantic indexing," in *TRECVID 2014*, 2014.

[4] C. G. M. Snoek, S. Cappallo, J. van Gemert, A. Habibian, T. Mensink, P. Mettes, R. Tao, D. C. Koelma, and A. W. M. Smeulders, "MediaMill at TRECVID 2014: Searching concepts, objects, instances and events in video," in *TRECVID 2014*, 2014.

[5] 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.

[6] S. Ishikawa, M. Koskela, M. Sjöberg, R. Anwer, J. Laaksonen, and E. Oja, "PicSOM experiments in TRECVID 2014," in *TRECVID 2014*, 2014.

[7] N. Inoue, Z. Liang, M. Lin, T. Hai, K. Shinoda, X. Zhang, and K. Ueki, "TokyoTech-Waseda at TRECVID 2014," in *TRECVID 2014*, 2014.

[8] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Dense trajectories and motion boundary descriptors for action recognition," *Int. J. Comput. Vis.*, vol. 103, no. 1, pp. 60–79, 2013.

[9] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio, "Theano: a CPU and GPU math expression compiler," in *Proceedings of the Python for Scientific Computing Conference (SciPy)*, June 2010, Oral Presentation.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, pp. 1106–1114, 2012.

[11] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," in *International Conference on Learning Representations (ICLR 2014)*. April 2014, CBLS.

[12] 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.

[13] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *NIPS*, 2014.

[14] S. Ayache and G. Quénot, "Video corpus annotation using active learning," in *In 30h European Conference on Information Retrieval (ECIR' 08)*, 2008, pp. 187–198.

[15] J. Blanc-Talon, W. Philips, D. C. Popescu, P. Scheunders, and P. Zemcík, "Advanced concepts for intelligent vision systems," in *Proceedings of 14th International Conference, ACIVS 2012*, 2012.

[16] "Model zoo," https://github.com/BVLC/caffe/wiki/Model-Zoo/.