TokyoTech at TRECVID 2015

Nakamasa Inoue, Tran Hai Dang,
Ryosuke Yamamoto, Koichi Shinoda
Tokyo Institute of Technology.
{inoue, dang, ryamamot}@ks.cs.titech.ac.jp
shinoda@cs.titech.ac.jp

1 Localization

We developed a localization system using Spatio-Temporal Selective Search (ST-Selective Search) and Convolutional Neural Network (CNN) with Spatial Pyramid Pooling (SPP-net) [1].

ST-Selective Search is an extension of the conventional 2D Selective Search [2] to spatio-temporal (3D) search. It uses optical-flow features in addition to color or texture features for video segmentation to produce temporally continuous candidate bounding-boxes for object detection. Then SPP-net is adopted to detect objects from the candidates produced by ST-Selective Search. It processes a large number of bounding-boxes efficiently by sharing lower convolutional layers of the network among these boxes. To improve the robustness against noise and blur, multi-frame score fusion and neighbor score fusion are also introduced.

One of our runs achieved 0.6688 mean pixel F-score, the highest score among all teams. Another run achieved the 3rd place in harmonic mean of spatio-temporal F-scores among all 6 teams.

1.1 Method

1.1.1 3-Dimensional Segmentation

The Selective Search [2] is extended so that it will process not only intra-frame similarities such as color and texture similarities but also inter-frame similarities such as optical-flow value similarity for temporal segmentation. We call this method Spatio-Temporal Selective Search (ST-Selective Search).

In detail, the following method is used for the ST-Selective Search.

1. Split a video at every I-frame
2. Extract a set of frames from each split video
3. Extract color and optical-flow features from each pixel of each frame
4. Generate super-pixels with color similarity for each frame
5. Extract texture features from each super-pixel
6. Generate intra- and inter-frame edges connecting super-pixels
7. Hierarchically merge super-pixels connected with the strongest edge and update features of merged super-pixel in the same way as the original Selective Search
8. Repeat step 2-7 until whole video is processed
9. Repeat step 1-8 with changing values of following parameters: initial super-pixel size and weights of features, in the same way as the original Selective Search [2]

In step 1, videos are split at every I-frames into short segments containing about 12 frames since segmenting a whole video or shot is time consuming. In step 4, super-pixels are not generated over frames to avoid under-segmentation.

1.1.2 Spatial Pyramid Pooling in CNN

ST-Selective Search will produce a large number of object bounding-boxes from each frame. To reduce calculation cost, Spatial Pyramid Pooling (SPP) [1] is adopted. SPP is a special layer connecting convolutional layers working as feature extractor and fully-connected layers working as a classifier. It will pool features of a ROI (region of interest) from suitably scaled feature map from a whole image. The network adopting SPP can process a large number of objects with less time than usual networks because the system just have to calculate a few scaled deep CNNs per image and a shallow neural network for each object as shown in [1].

Following [1], a CNN proposed in [3] is used as feature extractor before the SPP layer. Support Vector Machines (SVMs) for TRECVID concepts are put as the last layer of network. They will work as 2 class (positive and negative) classifiers and one of them will be used while the test phase. Note that the other layers are shared among all TRECVID concepts.

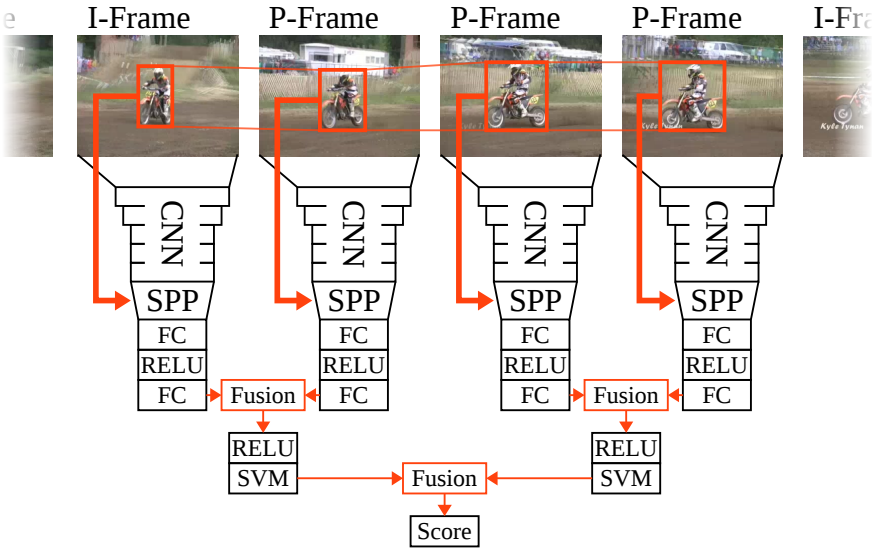

Figure 1: The diagram of our scoring method. The results of 3-Dimensional Segmentation is used on Spatial Pyramid Pooling (SPP) of each frame. Fusion is Multi-frame Score Fusion.

1.1.3 Multi-frame Score Fusion

Information of P-frames between I-frames are also useful for localization since movements of objects in P-frames can be traced by ST-Selective Search. An I-frame and 3 P-frames between I-frames are taken as input of CNNs and their CNN outputs are fused in a certain depth as shown in Figure 1.

1.1.4 Neighbor Score Boosting

Neighbor Score Boosting is introduced to improve the robustness against occlusion, blurring, noise and etc. If there are object bounding-boxes overlapping with a high-scoring object bounding-box in a neighbor frame, their scores will be boosted with the following formula.

$$BoostedScore_i = BaseScore_i + \beta \max_j \frac{area(BB_i \cap NBB_j)}{area(BB_i \cup NBB_j)} \quad (1)$$

where β is the boosting multiplier, BB is bounding-boxes in a current frame and NBB is high-scoring bounding-boxes in neighbor frames. Then scores of these objects with scores below the threshold will be boosted and some of them will be selected as positives.

1.2 Experiments

1.2.1 Experimental Conditions

On the segmentation stage, Selective Search parameters are tuned for videos in TRECVID training dataset since it is found that parameters used in the original Selective Search code is not suitable for noisy videos. On the classification stage, CNN layers pre-trained with the ILSVRC 2012 dataset provided with the SPPnet [1] model are used below the SPP layer. Fully-connected layers above the SPP layer are fine-tuned with positive samples and hard-negative samples overlapping positives for all concepts on *Caffe* [4] and SVMs are trained with positive samples and negative samples for each concept as done in [1]. The IACC.2.A dataset is used for training and the IACC.2.B dataset is used for validation to optimize parameters.

The conditions and results derived from the validation set are shown in Table 1 and in Table 2, respectively. The best threshold and the best fusion method are adopted from this experiment for this year's TRECVID submission. Finally, it is concluded that fusion layers taking average positioned after the 2nd FC layer and the last SVM is the optimum method as shown as 3Avg-5Avg in Figure 1.

For this year's TRECVID submission, the IACC.2.A and the IACC.2.B datasets are used for fine-tuning and training to acquire more accuracy. We annotated 12K I-frames for "Anchorperson" and 7K

Name	Boxes	Fusion Method	Threshold	β
Base_S	Single	None	-0.65	0.0
Base_M	Multiple	None	-0.65	0.0
Single2	Single	3Avg-5Avg	-0.65	0.0
Multiple	Multiple	3Avg-5Avg	-0.65	0.0
Multiple_Aug3	Multiple	3Avg-5Avg	-0.55	0.4
Multiple_Spat	Multiple	3Avg-5Avg	0.85	0.0

Table 1: The experimental conditions of each method. Boxes: Single will contain up to 1 bounding-box per I-frame, Multiple will contain multiple. Fusion Method: None will just use a score from the I-frame, 3Avg-5Avg will fuse scores at 3rd and 5th layers as shown in 1.1.3. Threshold: the detection threshold. β: score boosting multiplier shown in 1.1.4.

	Validation set			Test set		
Name	I-Frame F	Pixel F	F-like	I-Frame F	Pixel F	F-like
Base_S	0.4747	0.3809	0.4227			
Base_M	0.4747	0.4129	0.4416			
Single2	0.4873	0.3891	0.4327	0.6699	0.4450	0.5348
Multiple	0.4873	0.4211	0.4518	0.6699	0.4984	0.5716
Multiple_Aug3	0.4785	0.4564	0.4569	0.6683	0.5046	0.5750
Multiple_Spat	0.2239	0.3218	0.3578	0.3791	0.6688	0.4839

Table 2: The results of each method. F stands for F-score. F-like shows a integrated score of I-Frame F-score and Pixel F-score.

I-frames for "Computers" in the IACC.2.A dataset and used them for training since there are no ground truths provided for these two concepts added from this year.

We submitted following four runs. They are at the bottom of Table 1.

Single2

This run is the simplest run of our submitted runs. Scores are extracted from each frame and are fused as explained above. One object bounding-box with the highest score is selected as a positive if its score is above the threshold.

Multiple

This run is a modification of Single2. In this run, all object bounding-boxes with a score above the threshold are selected as positives since multiple bounding-boxes in one frame are allowed in TRECVID localization task.

Multiple_Aug3

In this run, Neighbor Score Boosting explained above is added to Multiple. Note that the threshold is re-optimized for Neighbor Score Boosting.

Multiple_Spat

This run optimizes parameters based on spatial (pixel) F-score in Multiple_Aug3.

1.2.2 Results

Since there are no I-frame and pixel F-scores integrated measure, we also report F-score like measure of them to compare methods. This is defined as the harmonic mean of them as follows.

$$F\text{-}like = \frac{2 \cdot I\text{-}FrameF \cdot PixelF}{I\text{-}FrameF + PixelF} \tag{2}$$

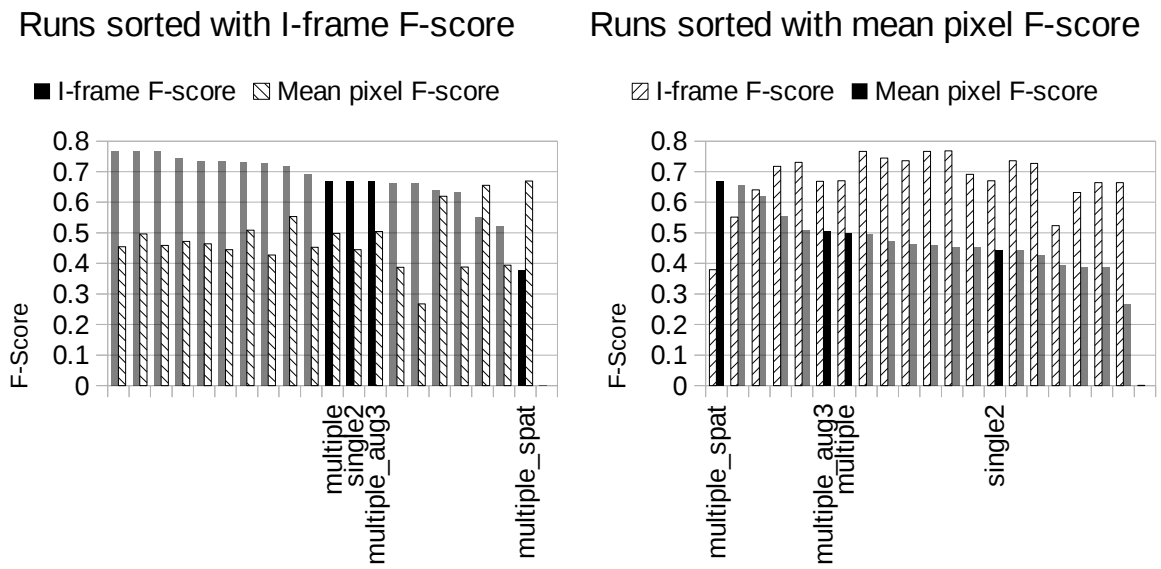

Figure 2: Overview of results of the localization task in TRECVID 2015. Our runs are colored in black. One of our runs Multiple_Spat achieved the best in mean pixel F-score.

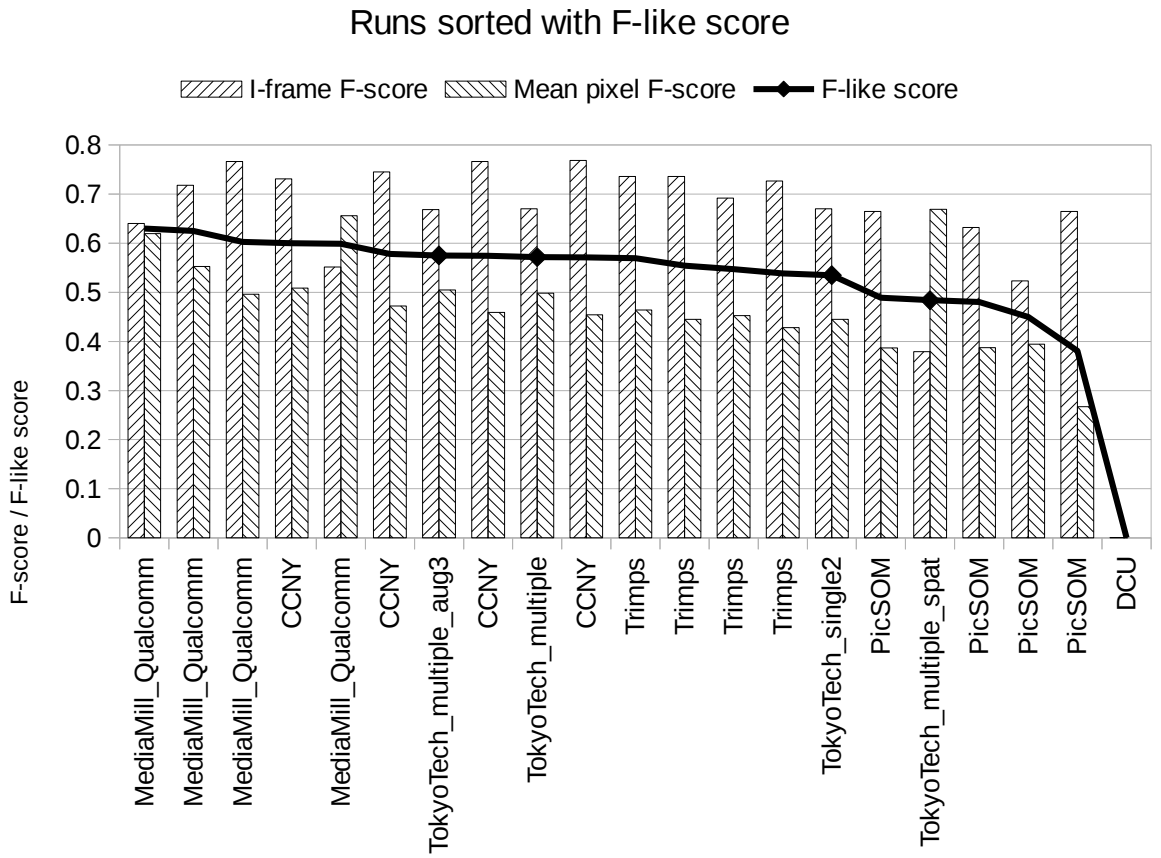

Figure 3: Overview of results of the localization task in TRECVID 2015 with F-like scores. Our runs are marked with black diamonds. Our best run is placed in 3rd among all 6 teams.

As shown in Table 2, the scores of test set are higher than these of validation set since amount of training data is larger as mentioned in Sec. 1.2.1. Our best run, Multiple_Aug3 achieved the highest score among our runs in test set as in validation set. This run is placed in the 3rd position in F-like score among all 6 teams. Multiple_Spat achieved 0.6688 mean pixel F-score, the highest score among all teams. Comparisons with other teams are shown in Figure 2 and Figure 3.

1.3 Conclusion

We proposed a localization system using ST-Selective Search and SPP-net. Our best run achieved 0.6688 mean pixel F-score, the highest score among all teams. Another run was placed in the 3rd place in harmonic mean of spatio-temporal F-scores among all 6 teams. Our future work will focus on extending convolutional networks to spatio-temporal space. We also need to increase the amount of training data to learn such networks.

2 Semantic Indexing

We propose a hybrid system of deep convolutional neural networks (CNNs) and Gaussian mixture model (GMM) supervectors [6, 7, 8] for semantic indexing [5]. This year, we introduced temporal max-pooling to CNN features extracted from multiple frames in a video shot. The 16-layered network in [9] trained on the ImageNet dataset is used to extract CNN features. Finally, it is combined with our baseline system using GMMs by score fusion. Our best result was 0.299 in terms of Mean InfAP, which was ranked third among participating teams.

2.1 Method

2.1.1 Deep Convolutional Neural Networks

Deep convolutional neural network (CNN) is introduced to extract feature vectors. A feature vector, which represent an image frame, consists of the activations on the second-last layer of a network, i.e., from the 16-layered network in [9], a 4096 dimensional feature vector is extracted from the 15th fully connected layer. Max pooling is applied to aggregate features extracted from multiple image frames. Finally, support vector machines (SVMs) are trained for each semantic concept by using these aggregated features as input.

2.1.2 GMM Supervectors

GMM supervectors [6, 7, 8] represent an image frame by a concatenation of mean vectors of an estimated GMM on low-level features. First, the following six types of visual and audio low-level features are extracted from video data.

1. SIFT features with Harris-Affine detector (Har-SIFT) [10, 11]
2. SIFT features with Hessian-Affine detector (Hes-SIFT) [11]
3. SIFT and hue histogram with dense sampling (Dense-SIFTH) [12]
4. HOG with dense sampling (Dense-HOG)
5. LBP with dense sampling (Dense-LBP) [13]
6. MFCC audio features (MFCC)

Note that principal component analysis is applied to reduce the dimensions of each type of low-level feature to 32. Second, from a set of low-level features, parameters of a Gaussian mixture models (GMMs) is estimated under the maximum a posteriori (MAP) criterion. Its probability density function (pdf) is given by

$$p(x|\theta) = \sum_{k=1}^{K} w_k \mathcal{N}(x|\mu_k, \Sigma_k), \tag{3}$$

where x is a low-level feature, $\theta = \{w_k, \mu_k, \Sigma_k\}_{k=1}^{K}$ is a set of GMM parameters, K is the number of Gaussian components (vocabulary size), w_k is a mixture coefficient, and $\mathcal{N}(x|\mu_k, \Sigma_k)$ is a Gaussian pdf

with a mean vector μ_k and a covariance matrix Σ_k. Third, a GMM supervector is extracted by combining normalized mean vectors as

$$\phi(X_{\mathrm{F}}) = \begin{pmatrix} \tilde{\mu}_1 \\ \tilde{\mu}_2 \\ \vdots \\ \tilde{\mu}_K \end{pmatrix}, \quad \tilde{\mu}_k = \sqrt{w_k^{(U)}}(\Sigma_k^{(U)})^{-\frac{1}{2}}\hat{\mu}_k. \tag{4}$$

2.1.3 Late Fusion

Support vector machines (SVMs) with the following RBF-kernel are used to train discriminative models for each semantic concepts.

$$k(X_{\mathrm{F}}, X'_{\mathrm{F}}) = \exp\left(-\gamma\|\phi(X_{\mathrm{F}}) - \phi(X'_{\mathrm{F}})\|_2^2\right), \quad \gamma = \frac{1}{\tilde{d}}, \tag{5}$$

where $\tilde{d}$ is the average distance between two GMM supervectors or CNN features. Here, annotations are obtained from the collaborative annotations [14]. Finally, trained discriminative functions are linearly combined as

$$f(X) = \sum_{\mathrm{F}\in\mathcal{F}} \alpha_{\mathrm{F}} f_{\mathrm{F}}(X_{\mathrm{F}}), \quad 0 \leq \alpha_{\mathrm{F}} \leq 1, \quad \sum_{\mathrm{F}} \alpha_{\mathrm{F}} = 1. \tag{6}$$

where $\mathcal{F}$ is a set of feature types. Combination coefficients α_{F} are optimized on a validation set.

2.1.4 Video-Clip Scores

The relationship between shots are useful for detecting semantic concepts. For example, Safadi et al. [15] proposes a re-ranking method to re-evaluate scores of video shots by using shot-score distributions. In our re-ranking method, we define a video-clip score as the maximum value of shot scores among all the shots in a video clip:

$$s_{\max} = \max_i s_i \tag{7}$$

where $s_i (i = 1, 2, \cdots, n)$ are shot scores for a video-clip that consists of n shots. Our final score for ranking shots is given by

$$s'_i = (1 - p)s_i + ps_{\max} \tag{8}$$

where p is a probability of appearance of a semantic concept in a video clip given by

$$p = r\left\langle \frac{\#(\text{positive shots in a video clip})}{\#(\text{shots in a video clip})} \right\rangle. \tag{9}$$

where r is a scaling parameter. The final score s'_i gets closer to $s_{\max}$ as the concept appear more often (e.g. an anchorperson in a news video).

2.2 Experiments

Figure 4 shows the overview of results of the semantic indexing task [5, 16, 17]. Figure 5 shows InfAP by semantic concepts. Our best result obtained by the run of TokyoTech_1 was 0.299 in terms of Mean InfAP, which is ranked 9th among all runs and is ranked 3rd among participating teams. Details of our four runs are as follows.

TokyoTech_4

This run used average weighting for late fusion. CNN features and GMM supervectors for the six types of visual and audio features are fused. This run achieved 0.287 in Mean InfAP.

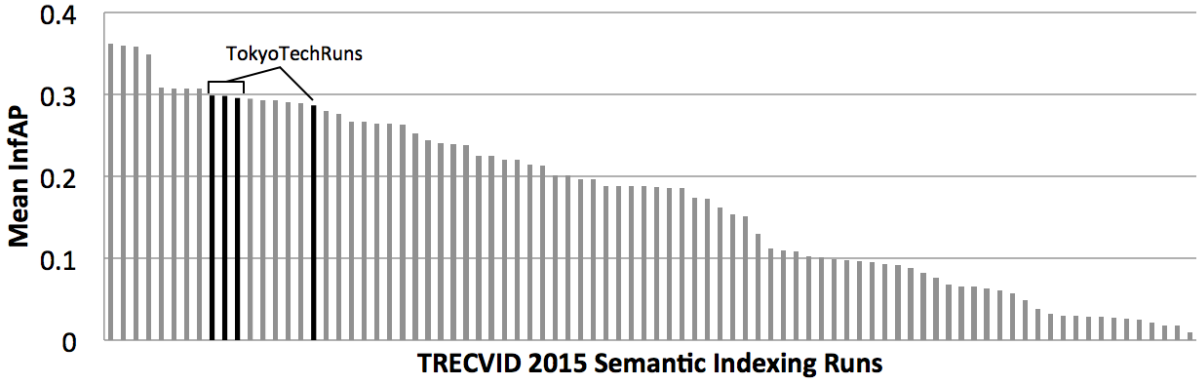

Figure 4: Overview of results of the semantic indexing task in TRECVID 2015. Our best result was Mean InfAP of 29.9%. Our four runs are colored in black.

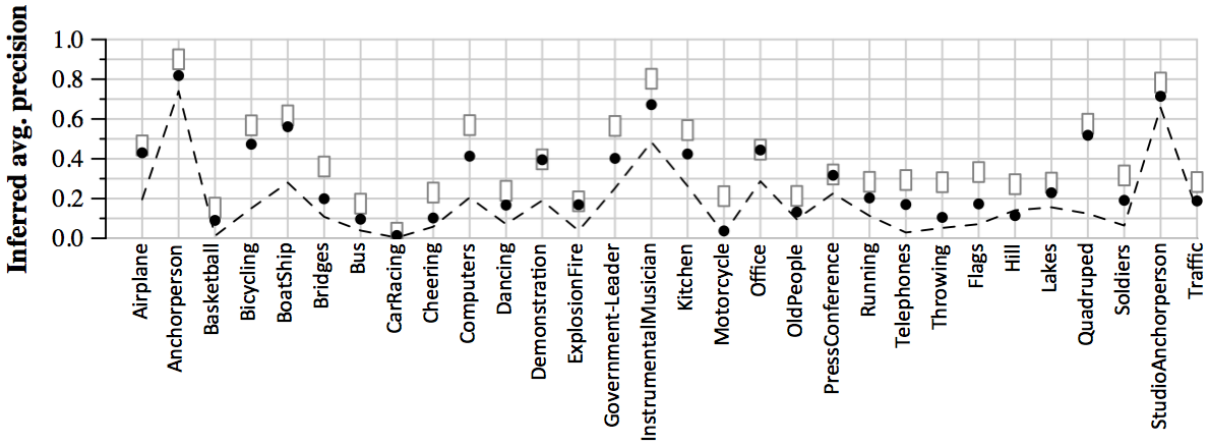

Figure 5: InfAP by semantic concept.

TokyoTech_1, 2, and 3

These runs optimized weights for late fusion on IACC_1_A dataset. The steepest descent method is used for optimization, in which Mean AP on the IACC_1_A dataset for initial weight values. TokyoTech_1, 2, and 3 stop iteration at 20th, 15th, and 1st epochs, respectively. They achieved 0.296, 0.298, and 0.299 in Mean InfAP. We also confirmed that video-clip scoring applied to all runs improved performance by 1.0% on average.

2.3 Conclusion

We proposed a high-performance semantic indexing system using CNN features and GMM supervectors. Our best result was 0.299 in terms of Mean InfAP, which was ranked third among participating teams in the semantic indexing task. Our future work will focus more on the spatio-temporal analysis based on neural networks.

3 Multimedia Event Detection

In Multimedia Event Detection task [18] in this year, we add VideoStory features to our GMM-supervector system using four types of low-level features. We submit runs under the condition with 10Ex and 100Ex for the Pre-Specified (PS) task and 10Ex for the Ad-Hoc task. With the EvalSub dataset, our result ranked 3rd among 7 teams in PS 100Ex, and 9th among 16 teams in PS 10Ex.

Type of features	MAP(%)
DT-HOG	20.04
DT-HOF	17.66
DT-MBH	21.51
DT-HOG+HOF+MBH	29.00

Table 3: The effectiveness of DT features on Kindred dataset under 100Ex

Settings	EvalFull	EvalSub
HOG+SIFT+DT	9.19	13.88
HOG+SIFT+DT+MFCC	9.64	13.73
HOG+SIFT+DT+VS	8.96	13.98

Table 4: The comparison in infAP200 (%) of our runs

3.1 VideoStory representation (VS)

VideoStory is a video representation that combines videos and their textual descriptions such as titles [23]. It is computed by learning a visual projection from low-level visual features and a textual projection from video titles simultaneously.

We follow the VideoStory algorithm to train the projections to compute features. From 38,457 videos from VideoStory46K [23] and their titles, we extract low-level visual features and term vectors, and then train textual projection and visual projection in the form of 2 matrices. Regularization parameters are optimized by cross-validation. DT-MBH [22] is used as low-level visual features. For TRECVID training and test videos, we use the projections trained on VideoStory46K to compute VideoStory representations.

3.2 GMM supervectors

We also use four other different types of features as follows.

- Dense HOG features (HOG)

  We use 32-dimensional histogram of oriented gradients (HOG) features [19]. We apply three levels of spatial pyramids: 1×1, 2×2, and 3×1 [20] [21]. PCA is applied for normalization.

- RGB-SIFT features (SIFT)

  To capture color information, we extract Scale-Invariant Feature Transform (SIFT) features [10] from each of RGB channels, then combine to a 384-dimensional feature. We also apply spatial pyramids with three levels: 1×1, 2×2, 3×1. PCA is used to reduce the dimension to 64.

- Dense trajectory features (DT)

  Dense trajectory is an motion feature for action recognition in [22]. We resize images to the width of 160, and skip every other frames of each interval. We use PCA to reduce the number of dimensions of HOG, HOF, and MBH descriptors to 32, 32, and 64, respectively.

- MFCC features (MFCC)

  To capture audio information, we use 38-dimensional Mel Frequency Cepstral Coefficient (MFCC) including ΔMFCC, $\Delta\Delta$ MCFF, Δ power, and $\Delta\Delta$ power. PCA is applied for normalization.

Gaussian Mixture Model (GMM) supervectors are extracted as in [6, 7, 8] from these features. Maximum a posteriori (MAP) adaptation and Universal Background Model(UBM) are used to make GMM supervectors. The number of Gaussian mixtures is set to 512. SVM is used to score videos for each type of features. Late fusion is applied to fuse SVM scores obtained from these features.

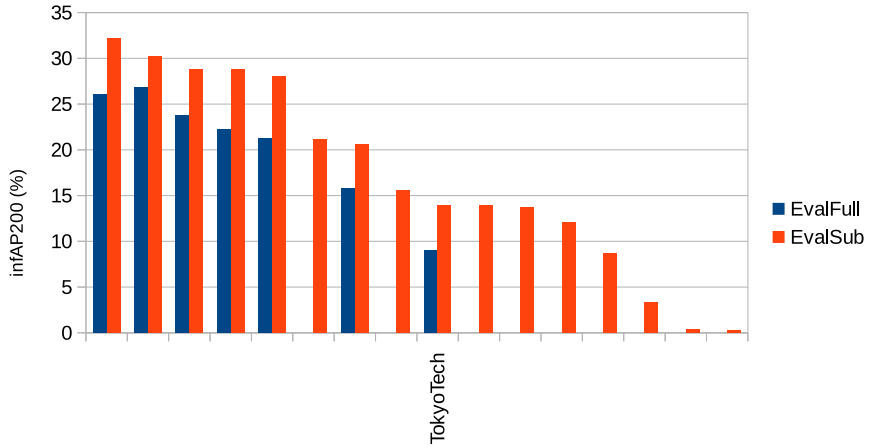

Figure 6: The comparison of infAP200 (%) in 2015 for Pre-Specified task under 10Ex

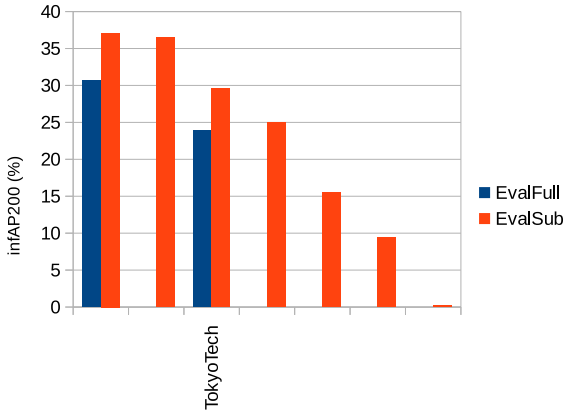

Figure 7: The comparison of infAP200 (%) in 2015 for Pre-Specified task under 100Ex

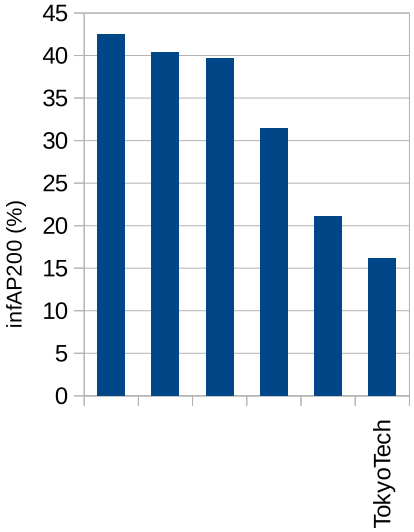

Figure 8: The comparison of infAP200 (%) in 2015 for Ad-Hoc task under 10Ex

3.3 Experimental Results

Our primary system combines HOG, SIFT, DT and VS features. This setting is common among all conditions: PS 100Ex, PS 10Ex, AH 10Ex. Comparison with other teams is showed in Figure 6, 7, and 8. We see that our system worked better for 100Ex than 10Ex because the number of training samples was not enough to train SVMs in 10Ex. DT features were the most effective among the seven types of features because they capture actions that are important to detect events. The effectiveness of combining 3 types of descriptors for DT is shown in Table 3. Audio MFCC and VS improved the performance as shown in Table 4.

VideoStory shows effectiveness in events consisting familiar concepts such as “Rock climbing”, “Fixing musical instruments”, “Parking a vehicle”, and “Tuning musical instruments”. VS does not improve the system in events such as “Giving directions to a location”, “Beekeeping”, “Wedding shower”, and “Playing fetch”, because concepts of these events are not popular in the pre-training dataset VideoStory46K. We conclude that VideoStory is a compact feature to represent concepts appearing in videos. It is needed to increase the amount of training data with textual descriptions for improving the performance.

3.4 Conclusion

This year we added VideoStory features to our system based on GMM supervectors. Our best run ranked 3rd among 7 teams in PS 100Ex EvalSub. Our future work will focus on deep learning techniques such as deep convolutional neural networks for event detection.

References

[1] K. He, X. Zhang, S. Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition. In IEEE Transactions on Pattern Analysis and Machine Intelligence, pp.1904-1916, 2015

[2] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, A. W. M. Smeulders, Selective search for object recognition. In *IJCV*, vol.104, pp.154-171, 2013

[3] M. D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks. In *ECCV*, pp.818-833, 2014

[4] Y. Jia, et al., Caffe: Convolutional Architecture for Fast Feature Embedding. Proc. ACM Multimedia Open Source Competition, 2014.

[5] 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 Proc. of *TRECVID workshop*, 2015.

[6] N. Inoue, and K. Shinoda. A Fast and Accurate Video Semantic-Indexing System Using Fast MAP Adaptation and GMM Supervectors. In IEEE trans. on Multimedia, vol.14, no.4, pages 1196–1205, 2012.

[7] N. Inoue, and K. Shinoda. A Fast MAP Adaptation Technique for GMM-supervector-based Video Semantic Indexing Systems. In Proc. of *ACM Multimedia* (short paper), 2011.

[8] N. Inoue, and et al. High-Level Feature Extraction using SIFT GMMs and Audio Models. In Proc. of *ICPR*, 2010.

[9] K. Simonyan, and A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition In Proc. of *ICLR*, 2015.

[10] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *In IJCV*, 2004.

[11] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. In *IJCV*, 60(1):63–86, 2004.

[12] J. van de Weijer and C. Schmid. Coloring local feature extraction. In Proc. of *ECCV*, vol.2, pages 334–348, 2006.

[13] X. Wang, T. X. Han, and S.Yan. An HOG-LBP Human Detector with Partial Occlusion Handling. In Proc. of *ICCV*, pages 32–39, 2009.

[14] S. Ayache, and G. Quéenot. Video Corpus Annotation using Active Learning. In Proc. of *ECIR*, pp.187–198, 2008.

[15] B. Safadi and G. Qunot. Re-ranking by Local Re-scoring for Video Indexing and Retrieval. In Proc. of *CIKM*, 2011.

[16] A. F. Smeaton, P. Over, and W. Kraaij. Evaluation campaigns and TRECVid. In Proc. of *ACM MIR workshop*, pp.321–330, 2006.

[17] A. F. Smeaton, P. Over, and W. Kraaij. High-Level Feature Detection from Video in TRECVid: a 5-Year Retrospective of Achievements. In Multimedia Content Analysis, Theory and Applications, Springer Verlag, pp.151–174, 2009.

[18] S. Strassel, A. Morris, J. Fiscus, C. Caruso, H. Lee, P. Over, J. Fiumara, B. Shaw, B. Antonishek, M. Michel. Creating HAVIC: Heterogeneous Audio Visual Internet Collection In Proc. of *LREC*, 2012.

[19] N. Dalal and B. Triggs.B. Histograms of oriented gradients for human detection. In Proc. of *CVPR*, 2005.

[20] I. Laptev, M. Marszalek, C. Schmid, B. Rozenfeld, Learning Realistic Human Actions from Movies. In Proc. of *CVPR*, pp. 1–8, 2008.

[21] S. Lazebnik, C. Schmid and J. Ponce, Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In Proc. of *CVPR*, 2006.

[22] H. Wang and C. Schmid, Action recognition with improved trajectories. In Proc. ICCV, 2013.

[23] A. Habibian, T. Mensink, C. G. M. Snoek, VideoStory: A new multimedia embedding for few-example recognition and translation of events. ACM Multimedia, 2014.