TokyoTech at TRECVID 2016

Nakamasa Inoue, Ryosuke Yamamoto,
Na Rong, Koichi Shinoda
Tokyo Institute of Technology.
{inoue, ryamamot, na}@ks.cs.titech.ac.jp
shinoda@cs.titech.ac.jp

1 Localization

This year, we introduced Faster R-CNN[1] and LSTM to our last year's system[2] which uses multi-frame score fusion and neighor score boosting. Faster R-CNN, the state-of-the-art method for object detection on still image, is used not only to detect objects but also to generate region proposals. An LSTM layer is introduced to Faster R-CNN for action concepts. Our best result was 0.2582 in I-frame F-score and 0.1393 in Mean Pixel F-score, which was ranked second among 3 teams participated.

1.1 Training Data

IACC.1.A-C data sets are used as development data[3]. They have shot-level annotations but their bounding-box information are not provided. For target concepts, we manually annotated bounding-boxes on the representative key-frames (RKFs) in each shot. For Sitting_Down, we annotated every frame in shots in order not to miss the actions done in a short period of time which are not overlapped on the RKFs. Table 1 shows the numbers of annotated frames and those of bounding-boxes for each concept.

ID	Concept name	# Annotated frames	# Bounding-boxes
1006	Animal	11545	9155
1013	Bicycling	599	1355
1016	Boy	1848	2492
1038	Dancing	2118	5199
1049	Explosion_fire	2483	2402
1071	Instrumental_Musician	4923	7229
1100	Running	945	1394
1107	Sitting_Down	124682	515
1163	Baby	898	895
1434	Skier	320	521

Table 1: The numbers of annotated frames and bounding-boxes for each concept.

1.2 Methods

We introduced Faster R-CNN[1] and LSTM to our last year's system[2] as follows. First, region proposals are generated from sparse sliding windows by Region Proposal Network of Faster R-CNN. In contrast to other object detection methods such as R-CNN[4], SPP net[5] and Fast R-CNN[6], Faster R-CNN can generate region proposals from sparse sliding windows. Second, the objects in region proposals are classified and scored by Detection Network of Faster R-CNN. We use a network with an LSTM layer for those concepts with action. Finally, Score Fusion and Score Boosting are applied for each region proposals.

1.2.1 LSTM Layer

This year, the Localization task includes concepts with actions such as Dancing, Running and Sitting_Down. It is hard to distinguish Sitting_Down from stable Sitting by a single image. In order to deal with this problem, we introduced an LSTM layer into our network. LSTM is one of recurrent neural networks which can recognize sequential data. It is suitable for detecting actions that have long and short context since it can memorize long and short term information. We replaced the second last layer of a network with an LSTM layer.

1.2.2 Score Fusion [2]

To improve robustness against varying object appearance in video, we employ multi-frame score fusion which calculates scores on four extra frames for each key-frame provided. Extra four frames are selected from P-frames succeeding each key-frame. Scores are calculated for each region proposal over the key-frame and the four P-frames, and are averaged to generate a final score.

1.2.3 Score Boosting [2]

Similar objects often appear spatially or temporally close to each other over more than one key-frame. For each key-frame, we increase the score of a region proposal which has a large overlap with a high scoring region in the adjacent key-frames as follows:

$$l'(r_{ti}) = l(r_{ti}) + \beta \max_{\{r'|r' \in R_{t\pm1} \cap l(r') \geq \mathrm{th}\}} \frac{\mathrm{size}(\mathrm{BB}(r_{ti}) \cap \mathrm{BB}(r'))}{\mathrm{size}(\mathrm{BB}(r_{ti}) \cup \mathrm{BB}(r'))}, \quad (1)$$

where t is a temporal index of a frame, R_t is a set of region proposals in t-th key-frame, r_{ti} is the i-th region proposal in R_t, $l(r_{ti})$ and $l'(r_{ti})$ are scores before and after boosting of region r_{ti}, β is a boosting multiplier, th is a detection threshold. We used the same values of β and th as last year's submission.

1.3 Experiments

We used Faster R-CNN[1] with ZF net[9], the smallest network for Faster R-CNN, implemented on *Caffe* [7] for our system. We chose it because of the shortage of GPU memory. We trained two networks; one is a ZF net without an LSTM layer used for all concepts except Sitting_Down and another is a ZF net with an LSTM layer put on its second last layer used for Sitting_Down. We used an LSTM implementation for *Caffe*[7] made by L. A. Hendricks[8]. We summarize our runs in the following.

faster

Faster R-CNN without an LSTM layer. This run is our baseline. Score Fusion and Score Boosting are not applied.

fusion

Faster R-CNN without an LSTM layer. Score Fusion is applied.

fusion.lstm

Faster R-CNN with an LSTM layer for Sitting_Down and without an LSTM layer for the other concepts. Score Fusion is applied.

boost

Faster R-CNN without an LSTM layer. Score Fusion and Score Boosting are applied.

boost.lstm

Faster R-CNN with an LSTM layer for Sitting_Down and without an LSTM layer for the other concepts. Score Fusion and Score Boosting are applied.

1.4 Results

Our result is shown in Figure 1 with the other teams' result. Our runs were second among 3 participating teams. A tendency of our scores among concepts was almost the same as other team's as shown in Figure 2 and Figure 3. Concepts with actions such as Running and Sitting_Down were difficult to detect for all teams.

As shown in Table 2, the scores of runs with LSTM were worse than those without LSTM. F-scores of runs with LSTM for Sitting_Down were zeros. The LSTM layer seems to be failed to train. Score Fusion and Score Boosting were worked well as in the last year. We achieved the best I-frame and mean pixel F-scores on Sitting_Down. This may be due to the frame-wise bounding-box annotation described in Section 1.1.

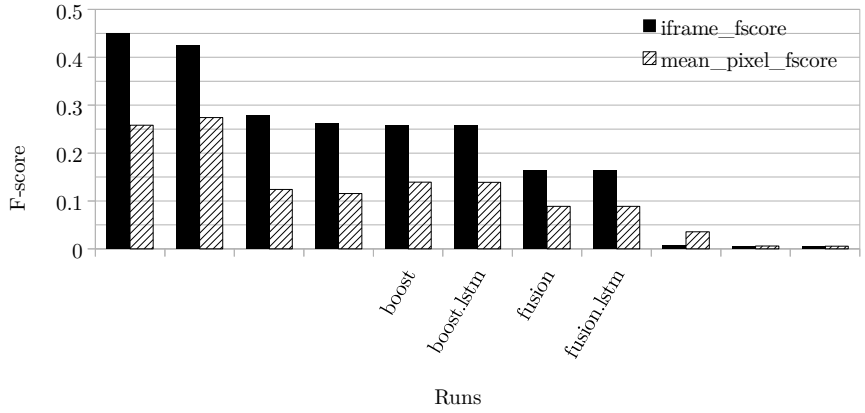

Figure 1: Overview of results of the localization task in TRECVID 2016. Runs are sorted with I-frame F-score. Our runs are labelled.

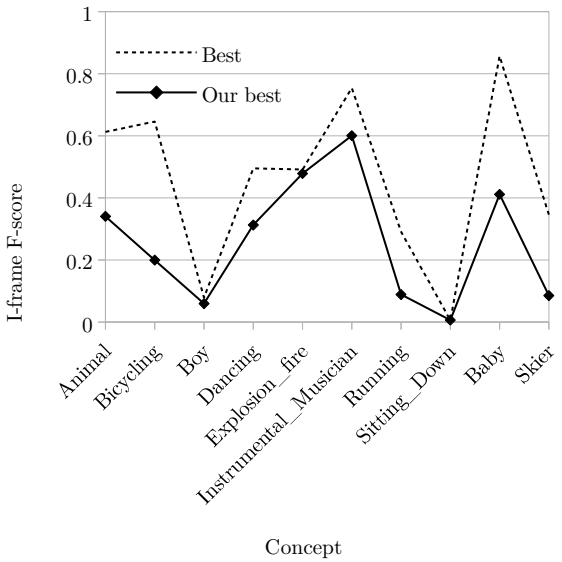

Figure 2: I-frame F-score comparison among 10 concepts. A dashed line shows the best scores among all runs from all teams. A solid line shows the best scores among all runs from us.

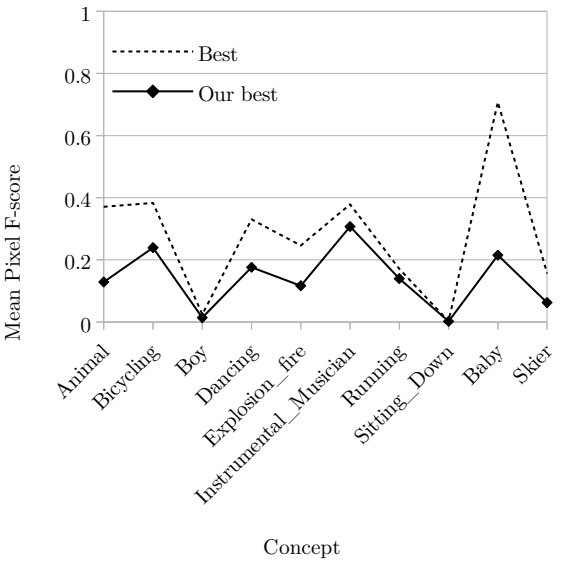

Figure 3: Mean Pixel F-score comparison among 10 concepts. A dashed line shows the best scores among all runs from all teams. A solid line shows the best scores among all runs from us.

		Sitting_Down		mean	
Name	**Methods**	**I-Frame F**	**Pixel F**	**I-Frame F**	**Pixel F**
faster	None	0.0007	0.0003	0.0723	0.0266
fusion	Fusion	0.0027	0.0010	0.1640	0.0890
fusion.lstm	LSTM, Fusion	0.0000	0.0000	0.1637	0.0889
boost	Fusion, Boost	0.0063	0.0022	0.2582	0.1393
boost.lstm	LSTM, Fusion, Boost	0.0000	0.0000	0.2576	0.1391
The best scores among all teams		0.0063	0.0022	0.4499	0.2743

Table 2: The results of each method. **Bold runs** are submitted. F stands for F-score.

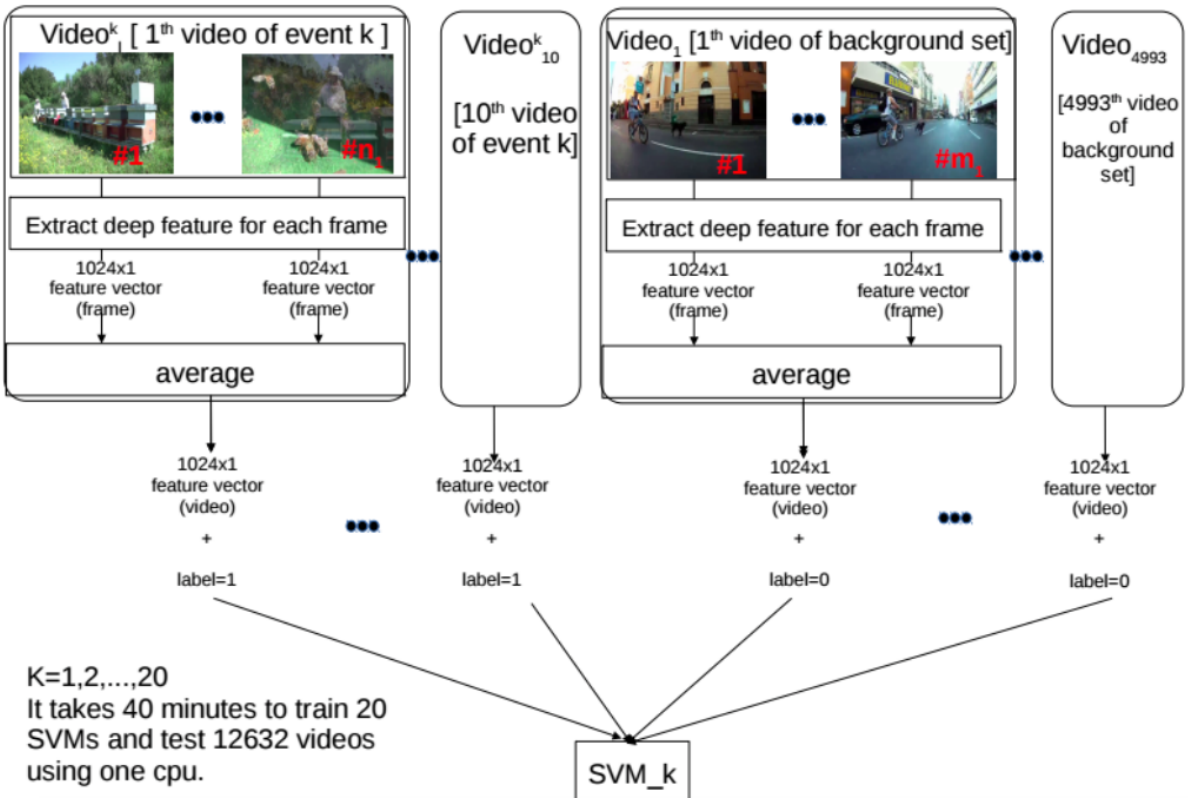

Figure 4: SVM structure

1.5 Conclusion

We proposed a localization system using Faster R-CNN with an LSTM layer. However, an LSTM layer did not work well on our experiment. Our best run achieved 0.2582 in I-frame F-score and 0.1393 in Mean Pixel F-score, which were second among 3 participating teams. We try to find why we failed to train an LSTM layer and re-train the network.

2 Multimedia Event Detection

In Multimedia Event Detection task of this year, we only use deep feature [10]. To get video representation, firstly we extract frames every two seconds, which are the key frames for each video, then we extract deep feature [10] of each frame. We use two ways to do this MED task. First, we train one SVM classifier for each event, while the the negative samples are those background videos. Second, we train only one LSTM model which has twenty classes (twenty events of PS task and the background). 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 5th among 7 teams in PS 100Ex, and 6th among 10 teams in PS 10Ex.

2.1 Deep feature

This year we only used the feature in [10], which is a representation learned from deep convolutional neural networks. It tries to leverage the complete ImageNet hierarchy for pre-training deep networks. To deal with the problems of over-specific classes and classes with few images, a bottom-up and top-down approach is used for reorganization of the ImageNet hierarchy based on all its 21,814 classes and more than 14 million images. We used the features at the pool5 layer, with a 1,024-dimensional frame representation.

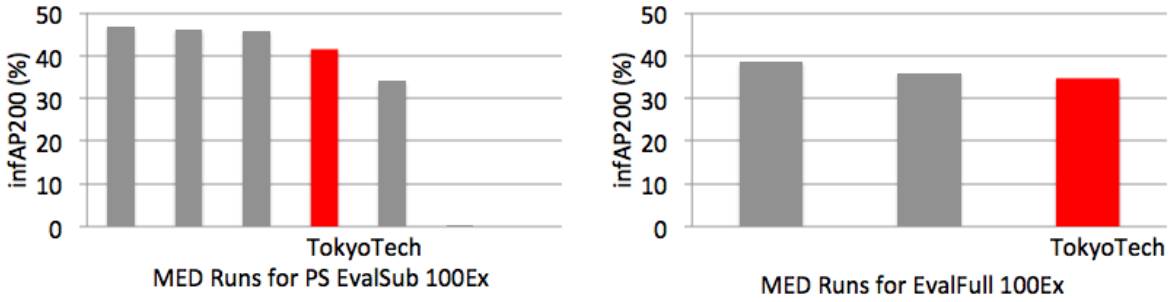

Figure 5: The comparison of infAP200 (%) in 2016 for Pre-Specified task under 10Ex

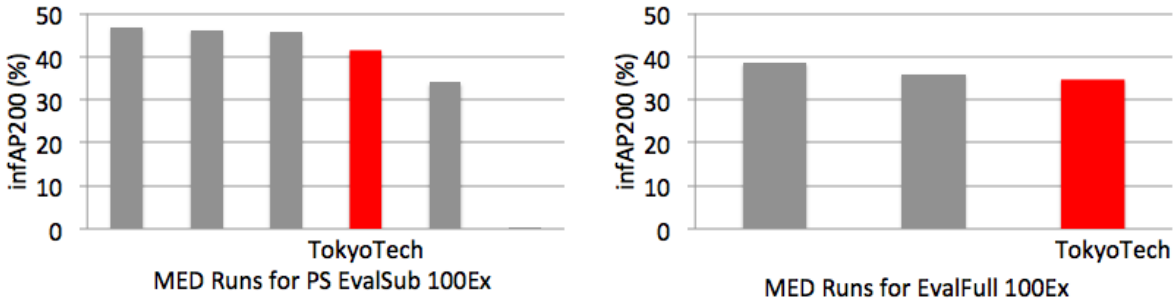

Figure 6: The comparison of infAP200 (%) in 2016 for Pre-Specified task under 100Ex

2.2 SVM

For each video, first, we extract deep features of the key frames to get frame representations, then we average them to get a video representation as shown in Figure 4. We totally train thirty SVM classifiers for thirty events of PS task and Ad-Hoc task. The positive samples are different within twenty events, while for negative samples we use same background videos for all SVM classifiers.

2.3 LSTM

Considering that each video consists of several sequential frames, which are relational, recurrent neural network (RNN) may be a good way to deal with this MED task. Long short-term memory (LSTM) is a RNN architecture (an artificial neural network), however, unlike traditional RNNs, an LSTM network is well-suited to learn from experience to classify, process and predict time series when there are very long time lags of unknown size between important events. We use nn package of Torch [11] to do the LSTM experiment, while the sequence of each video are variable. Unless SVM part, in which we train thirty classifiers for thirty events, we only train one LSTM model for 100Ex of Pre-Specified task, where the output is twenty-one classes, that is, twenty events and background.

2.4 Results

Our primary system only use deep features. This setting is common among all conditions: PS 100Ex, PS 10Ex, AH 10Ex. We also did LSTM experiment only with PS 100Ex. From Table 1, we can see that SVM results are greatly better than the LSTM results in evaluation set, however, in test dataset the gap between these two methods were not that huge.

2.5 Conclusion

This year we only consider deep feature, and two ways are used in this task, while the SVM gets better results than LSTM. With the EvalSub dataset, our result ranked 5th among 7 teams in PS 100Ex,

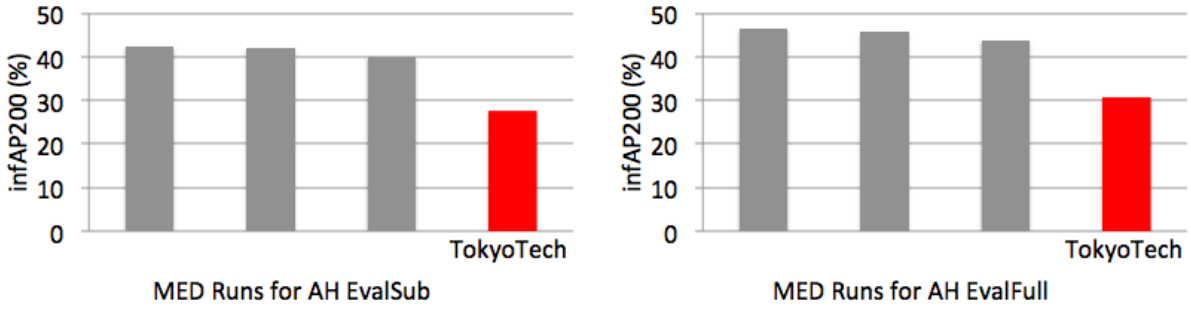

Figure 7: The comparison of infAP200 (%) in 2016 for Ad-Hoc task

Method	EvalSub	EvalFull
LSTM	1.7	0.9
SVM	41.5	31.7

Table 3: The comparison of SVM and LSTM

and 6th among 10 teams in PS 10Ex. Though it didn't get good results till now, we will focus on LSTM structure for event detection as our future work.

References

[1] S. Ren, K. He, R. Girshick, and J. Sun, Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In Proc. of *NIPS*, pp. 91–99, 2015

[2] N. Inoue, T. H. Dang, R. Yamamoto, and K. Shinoda, TokyoTech at TRECVID 2015. In Proc. of *TRECVID*, 2015

[3] G. Awad, J. Fiscus, M. Michel, D. Joy, W. Kraaij, A. F. Smeaton, G. Quenot, M. Eskevich, R. Aly, and R. Ordelman, TRECVID 2016: Evaluating Video Search, Video Event Detection, Localization, and Hyperlinking. In Proc. of *TRECVID*, 2016

[4] R. Girshick, J. Donahue, T. Darrell, and J. Malik, Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proc. of *CVPR*, pp. 580–587, 2014

[5] K. He, X. Zhang, S. Ren, and 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

[6] R. Girshick, Fast R-CNN. In Proc. of *ICCV*, pp.1440–1448, 2015

[7] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, Caffe: Convolutional Architecture for Fast Feature Embedding. In Proc. of ACM Multimedia Open Source Competition, pp. 675–678, 2014

[8] J. Donahue, L. A. Hendricks, S. Guadarrama,M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, Long-term Recurrent Convolutional Networks for Visual Recognition and Description. In Proc. of *CVPR*, pp. 2625–2634, 2015

[9] M. D. Zeiler and R. Fergus, Visualizing and understanding convolutional networks. In Proc. of *ECCV*, pp.818–833, 2014

[10] P. Mettes, D.C. Koelma, and C.G.M. Snoek, The ImageNet Shuffle: Reorganized Pre-training for Video Event Detection. In Proc. of *ICMR*, pp.175–182, 2016

[11] `http://torch.ch`