NEU_MITLL @ TRECVid 2015: Multimedia Event Detection by Deep Feature Learning

Joseph P. Robinson[1], **Edward Scott**[1], **and Yun Fu**[1,2]

[1]College of Electrical and Computer Engineering, Northeastern University, Boston, Masschusetts
[2]College of Computer and Information Science (Affiliated), Northeastern University, Boston, Masschusetts

ABSTRACT

We present a framework for multimedia event detection (MED), developed for TRECVID 2015 using convolutional neural networks (CNNs) to detect complex events by using deterministic models trained on video frame data. We used several well-known CNN models designed to detect objects, scenes, and a combination of both (i.e., , Hybrid-CNN). We also experimented with features from different networks fused together in different ways. The best results achieved was by fusing objects and scene detections at the feature-level (i.e., , early fusion), with a MAP of 16.02%. Results showed that, although there is room for improvement, our baseline framework is capable of recognizing various complex events in videos when there are only a few instances of each within a large search pool of *uneventful* videos.

1 Introduction

This report summarizes the performance of a system designed jointly by Synergistic Media Learning (SMILE) Lab of Northeastern University (NEU) and Massachusetts Institute of Technology Lincoln Laboratory (MIT-LL) for the TRECVid 2015 MED task. Specifically, we present results for the 10Ex, Pre-Specified (PS) evaluation protocol, which specifies 10 training exemplars (videos) for 20 events (E21-E40). See Section 2 or [2] for more information about the MED15 evaluation.

Our system uses pre-trained, deeply learned convolutional neural networks (CNNs) as off-the-shelf feature extractors, using outputs from either the last or second-to-last fully connected layers as feature vectors to detection events in video data. Networks used were VGG-16 [4], Places205-Alexnet, and Hybrid-CNN [6], which are further discussed in Section 3.1.

The rest of this report is organized as follows. First we give an overview of the MED15 task and provided data. We then present our system, discussing feature extraction, training, and classification, and resource usage. Results are presented for different runs, performance metrics, run-specific results, and computational resources used. This is followed by an analysis of the results, and then concluded.

2 MED Task

The goal of MED is to determine whether a particular video contains a particular complex event, such as a bike trick or rock climbing. As defined in the provided event kits, events typically coincide with the presence of certain objects in certain settings (i.e., scenes) [see Section 2.1.1].

2.1 MED data

2.1.1 Event Kits

There are 30 event kits (i.e., event types) provided for the MED15 evaluation: 20 pre-specified (PS) events and 10 Ad-Hoc events [see Table 1]. Each event kit comes with a text description that contains the event name, event definition, event explication, evidential description, and a set of training exemplar videos for the given event. Video exemplars are selected to be indicative of a particular event. However, due to the complexity of the events, all intra-class variations are unlikely to be represented in the training set. For MED15, there were 3 sets of event kits supported for the evaluation.

Pre-specified Events

E021	Bike trick	E031	Beekeeping
E022	Cleaning an appliance	E032	Wedding shower
E023	Dog show	E033	Non motorized vehicle repair
E024	Giving Directions	E034	Fixing a musical instrument
E025	Marriage Proposal	E035	Horse riding competition
E026	Renovating a home	E036	Felling a tree
E027	Rock climbing	E037	Parking a vehicle
E028	Town hall meeting	E038	Playing fetch
E029	Winning race w/out a vehicle	E039	Tailgating
E030	Working on metal crafts project	E040	Tuning a musical instrument

Table 1. A list of MED15 Pre-Specified Event Types. E021-E030 come from the MED12 collection, and E031-E040 come from MED13.

1. **0Ex:** No example video clips per event kit.
2. **10Ex:** 10 positive and up to 5 miss/ non-positive clips per event kit.
3. **100Ex:** 100 positive and up to 50 miss/ non-positive clips per event kit.

All experiments reported here used the 10Ex event kits.

2.1.2 Test Search Video Collection

MED15 participants were provided a search video set referred to as the Progress Search Set. This video collection was provided for "blind" testing and, hence, no ground truth was provided. [1] There are two sets of Pre-Specified video collections for teams to select from.

1. **MED15EvalFull:** Consisted of approximately 200,000 videos.
2. **MED15EvalSub:** Consisted of a subset of 32,000 videos from MED15EvalFull.

All tests reported here were with the MED15EvalSub collection.

3 System Overview

Our MED15 system uses pre-trained CNNs configured for and implemented with the Caffe framework [1] to serve as *off-the-shelf* feature extractors. These networks accepts images as inputs, so videos were first sampled at a rate of about one frame per second. Sample frames with an image entropy less than 0.03 are rejected, as this typically occurs when images are homogeneous in color, either with or without text overlaid (e.g., blank image, credits, logos, etc.). Such frames are not useful for our vision-based system and could add unnecessary noise to the training data. A maximum number of 160 frames per video is processed – if a video's frame count exceeds 160, then sample frames are selected at random. Deep features were extracted for each video by passing its sample-frames through the given pre-trained net. Then, a single, video-level feature was obtained by average pooling the frame-level features. The following subsection covers the pre-trained deep nets used in more detail.

3.1 Deep Features

three pre-trained CNNs were used– VGG-16, Places205-Alexnet, and Hybrid-CNN– which are described as follows:

VGG-16 [4] was trained on the ImageNet ILSVRC 2012 dataset, which consists of 1.3 million images for 1,000 object types [3].

[1] NIST scored submitted ranked lists of videos. For each submission, events were tested independently and event-specific results were returned by NIST thereafter.

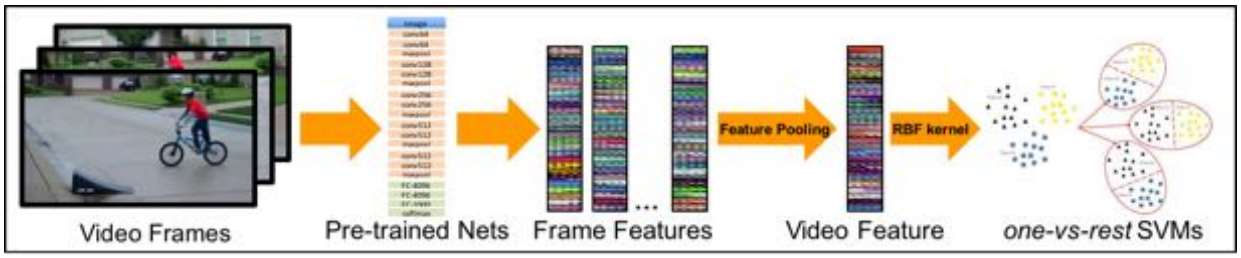

Figure 1. The work flow of our MED system: input videos are first sampled at approximately one fps, then preprocessed. Deep features are extracted from all samples of a video and average pooled to yield a video-level descriptor. RBF kernels project video descriptors to a higher dimensional feature space, from which *one-vs-rest* SVM models are based on.

Places205-Alexnet was trained on MIT's Places Database, which consists of 2.5 million images for 205 categories of Places [6]. The architecture imitates the Caffe reference network [1].

Hybrid-CNN was trained on Places Database (205 scene types) and a subset of ImageNet ILSVRC (978 object types) which, in total, consists of about 3.6 million images across 1183 class types [6]. The architecture imitates the Caffe reference network [1].

Figure 1 depicts the overall workflow of our MED system framework. Implementation was done using Caffe's open framework [1].

3.2 Model

We used support vector machines (SVM) to model events. Typically, a SVM model acts as a binary classifier with an objective to maximize the margin separating two classes. Using the method of [5], we trained *N one-vs-rest* SVM models (one per event type), i.e., , each SVM was trained on the respective in-class data for its positive samples, with all other (out-of-class) data representing negative samples.

These event-specific SVM classifiers were based on the video-level descriptors obtained by averaging pooling of the frame-level feature vectors– linear SVMs were trained on video-level descriptors non-linear kernel space via a χ^2 (RBF) kernel.[2] SVM models were stored as the event meta-data store, and the features of the test (search) videos were stored as the search-pool metadata. To search for a particular event, videos were ranked according to SVM scores.

Table 2. Results for MED15 submission. Note that *Run-1* was our evaluation submission, while the other runs were post-evaluation submissions.

Run ID	Network(s)
Run-1	VGG16-fc8 (1,000D) + Places205-fc8 (205D), averaged SVM scores.
Run-2	VGG16-fc8 (1,000D).
Run-3	Places205-fc8 (205D).
Run-4	Hybrid-CNN-fc8 (1,186D).
Run-5	VGG16-fc8 + Places205-fc8, concatenated features vectors (1,205D).
Run-6	VGG16-fc7 (4,096D).

[2]Kernel and SVM classifiers were implemented with VLFeat toolbox [5].

4 Experimentation

4.1 Description of runs

In this section, we report our results for the TRECVid's MED15 10Ex, PS subtask. We present a total of six runs: one of which was our official submission, while the other five were processed post-evaluation. Runs are summarized in Table 2.

4.2 Performance metrics

According to [2], performance was measured via mean average precision (MAP). For Q events, MAP was determined as follows:

$$MAP = \frac{1}{Q}\sum_{q=1}^{Q} AP(q),$$

AP, as a function of event q, is defined as

$$AP(q) = \frac{1}{P_Q}\sum_{tp=1}^{P_Q} Prec(tp) = \frac{1}{P_Q}\sum_{tp=1}^{P_Q} \frac{tp}{rank(tp)},$$

where P_Q is the number of training exemplars for event Q, which remains constant at 10 throughout each of our runs.

Computational resources

Development and most processing was done on a local machine equipped with an Intel Core i7-5930K CPU @ 3.50 GHz with 32 GB of memory and a 4GB GEForce GTX 970 GPU running on Ubuntu 14.04 LTS.

Table 3. Results of our official MED15 submission (i.e., *Run-1*).

	AP%	iP10	iP50	infAP200
E021	17.0	0.5	0.24	0.2389
E022	2.5	0.0	0.04	0.036
E023	26.9	0.7	0.38	0.2163
E024	1.0	0.0	0.0408	0.008
E025	0.3	0.0	0.0	0.0
E026	11.1	0.5	0.26	0.1737
E027	25.4	0.6	0.46	0.3377
E028	11.0	0.3	0.14	0.1345
E029	19.2	0.4	0.38	0.1556
E030	8.8	0.3	0.16	0.1558
E031	18.8	0.6	0.28	0.1542
E032	3.6	0.1	0.08	0.0485
E033	9.8	0.3	0.14	0.0831
E034	3.7	0.1	0.1	0.1007
E035	23.3	0.4	0.34	0.1851
E036	3.2	0.1	0.08	0.0449
E037	6.7	0.2	0.12	0.1303
E038	3.5	0.0	0.08	0.0511
E039	30.5	0.7	0.4	0.386
E040	9.6	0.1	0.18	0.0948
Mean	**11.8**	**0.295**	**0.195**	**0.1368**

5 Results

The results of our MED15 submission, as scored by NIST, are listed in Table 3.

We submitted several additional runs for scoring post-evaluation. Table 4 summarizes these results. To allow for a clearer comparison, the first column, *Run-1*, repeats the AP scores reported in Table 3.

Table 4. Submission scores for each event [AP (%)], along with the overall mean (MAP %).

	Run-1	*Run-2*	*Run-3*	*Run-4*	*Run-5*	*Run-6*
E021	17	**26.8**	5.8	21.4	19.4	17.2
E022	2.5	2.4	2.4	1.4	**4**	2.2
E023	26.9	26	14.1	24.8	33.6	**33.8**
E024	1	1	0.7	0.8	**3.3**	0.4
E025	0.3	0.3	**0.4**	0.3	0.3	0.2
E026	**11.1**	8.8	8.7	5.2	10.5	7
E027	25.4	32.5	18.1	29.7	36.8	**43.1**
E028	11	10	11.1	9.6	**17.4**	14.3
E029	19.2	16.9	22.7	12.9	**28.6**	25.4
E030	8.8	9.4	3.8	5.3	**11.9**	8.3
E031	18.8	**40.6**	8.5	37.2	25.6	23.7
E032	3.6	3	3.5	2	3.6	**4.4**
E033	9.8	10.8	2.6	4.5	**19.6**	6.3
E034	3.7	4.7	2.2	2.2	**7.3**	6.1
E035	23.3	22.5	18.3	24.9	**26**	18.9
E036	3.2	3.3	2.5	3.9	**5.9**	2.8
E037	6.7	7.6	6.4	7.3	**9.3**	5
E038	3.5	2.2	4.6	3.7	**4.9**	2.2
E039	30.5	25.2	25.7	27.8	**39.7**	14.6
E040	9.6	11.2	6.5	8.5	**12.7**	11.5
MAP	11.795	13.26	8.43	11.67	**16.02**	12.37

Table 4 shows performance in MAP of various system configurations (*Run-1* – *Run-6*, described in Table 2). *Run-1*, the official submission, returned a MAP of 11.80%. However, by tweaking feature extraction and SVM modules we were able to improve on this result. For *Run-1*, separate SVMs were trained on 1000D VGG-16 outputs and 205D Places-205 outputs. For each test video, the SVM scores were averaged into a single value – this is known as late fusion. In contrast, early fusion (*Run-5*) – concatenating the outputs of VGG-16 and Places-205 networks at the feature-level (i.e., , before passing to SVM) – increased MAP to 16.02%.

Interestingly, using only VGG-16 outputs also resulted in improved performance (*Run-2*, 13.26%) over the late-fusion hybrid model. In contrast, an off-the-shelf hybrid model (*Run-4*), performed similarly to *Run-1*. We can draw a few conclusions from this set of results. Object features (VGG-16) appear to be much more discriminative than scene features (Places-205), but scene features do provide useful information to the system. Additionally, the VGG-16/Places-205 hybrid is more discriminative than the Alexnet/Places-205 hybrid network.

6 Conclusion

We have presented an overview of a multimedia event detection system utilizing pre-trained CNN models to extract meaningful features from video data. Outputs from these models correspond to the probability that certain objects or scenes are present in a given video. The VGG-16 model proved to be the most discriminative when used independently, resulting in a MAP of 13.26%. Combining VGG-16 and Places-205 by concatenating their outputs (1000-object and 205-scene vectors, respectively) improved MAP to 16.02%. This system represents a strong

baseline and starting point for future work. The training scheme could be improved by using different models for the event-type classifiers and by augmenting the data with different feature types. The incorporation of temporal information and additional modalities such as audio and contextual cues will not only improve performance, but also move the system toward zero-shot detection capability, requiring no training exemplars to detect events in video data.

Acknowledgment

MIT Lincoln Laboratory for their financial and technical support.

References

1. Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.
2. Paul Over, George Awad, Martial Michel, Jonathan Fiscus, Wessel Kraaij, Alan F. Smeaton, Georges Quéenot, and Roeland Ordelman. Trecvid 2015 – an overview of the goals, tasks, data, evaluation mechanisms and metrics. In *Proceedings of TRECVID 2015*. NIST, USA, 2015.
3. Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, pages 1–42, April 2015.
4. K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.
5. A. Vedaldi and B. Fulkerson. VLFeat - an open and portable library of computer vision algorithms. In *ACM International Conference on Multimedia*, 2010.
6. Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In Z. Ghahramani, M. Welling, C. Cortes, N.d. Lawrence, and K.q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 487–495. Curran Associates, Inc., 2014.