NII-HITACHI-UIT at TRECVID 2016

Duy-Dinh Le [1], Sang Phan [1], Vinh-Tiep Nguyen [3], Benjamin Renoust [1],
Tuan A. Nguyen [2], Van-Nam Hoang [5], Thanh Duc Ngo [4], Minh-Triet Tran [3],
Yuki Watanabe [6], Martin Klinkigt [6], Atsushi Hiroike [6],
Duc A. Duong [4], Yusuke Miyao [1], and Shin'ichi Satoh [1]

[1] National Institute of Informatics, Japan
[2] The University of Tokyo, Japan
[3] University of Science, VNU-HCMC, Vietnam
[4] University of Information Technology, VNU-HCMC, Vietnam
[5] Hanoi University of Science and Technology, Hanoi, Vietnam
[6] Hitachi, Ltd., Tokyo, Japan

1 TRECVID 2016 Instance Search Task: Searching Specific Persons in Specific Locations

Abstract. This paper presents our proposed system for TRECVID Instance Search Task with novel compound query type. A query is to search a specific person in a specific location. The system consists of two main stages. First, we use Bag-of-Visual Word model with geometric verification to search for shots with the query location. Using deep features encoded by a Convolutional Neural Network (CNN), we filter out irrelevant ones. Second, we employ VGG-Face model with a linear kernel classifer to find out shots containing query person among filtered shots from previous stage. The final result shows that, the proposed system achieved very high recognition rate. Compared to other teams, our system is ranked at the third place for fully automatic run.

1.1 Introduction

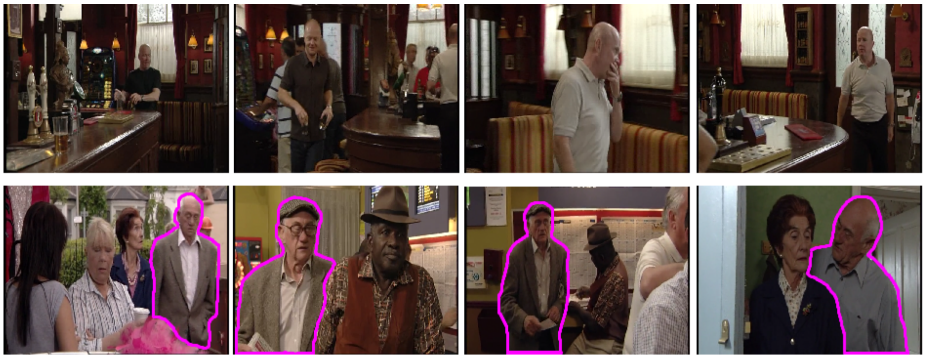


Fig. 1: A query topic includes location examples (first row images) and person examples (second row images) marked by magenta boundaries. Programme material copyrighted by BBC.

This year, TRECVID Instance Search task (INS) changes format of queries from searching a single instance such as object, person, location to a new one: retrieving a specific person at a specific location. This type of query has many applications in practice such as: surveillance systems, personal video archive management. Figure 1 gives an example of this type of query. Images in the first row are examples of a pub that a user want to search. These images cover multiple views of a location with many irrelevant or noisy objects such as humans, temporary decorations. These objects may cause low retrieval accuracy due to noisy features. Images in the second row are examples of the person that the user also need to find if he appears at the pub.

For location search, this problem can be solved to some extent by using Bag-of-Visual-Word (BOW) model[1] and its extensions e.g. geometric consistency checking [2], query expansion[3]. However, the performance of searching specific person is still very low due to the limited capacity of representation of the BOW model. This year, we propose a system which leverages both BOW and CNN based feature for retrieving this new type of query. For location search, we combine BOW based and CNN based features to improve the performance. For person search, we use VGG-face feature[4] for recognizing the first video shot that the human appears. In stead of using distance metric such L_2, we propose to use a linear kernel method to learn high-level feature encoded by deep CNN.

1.2 Framework Overview

This section describes our framework used in our experiments and its configurations. Query topic of the framework includes location and person examples. Output of the system is a rank list of video shots sorted in descending order of similarity score to both location and person examples.

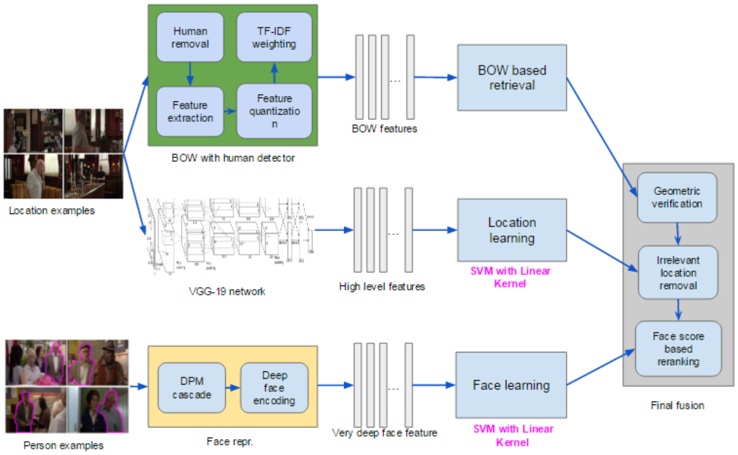


Fig. 2: Framework overview.

Our proposed system includes three main parts: BOW based retrieval, location verification and face recognition for re-ranking. Figure 2 sketches out the work flow of main components in our INS system. From location example images, we use Faster RCNN[5] with pretrained network on PASCAL VOC 2007 to remove human regions which contain a lot of noisy features. These features encode the person's clothes texture which sometime increases the bad effect of burstiness phenomena. Using BOW model, remained region are encoded by a very sparse feature vector which would be used to build inverted index and retrieval top ranked list of location.

However, BOW model is a non-structured model which does not take into account the spatial relationship between visual words. To remove irrelevant shot, we combine both RANSAC based algorithm and learning based approach for high level feature vector produced by a very deep CNN network VGG-19[6].

The second main part of this year query is person instance examples. From the person examples including color images and its masks which help the system to separate interested person from irrelevant objects. We localize and describe face examples by using DPM cascade[7] and deep face model [4] included in MatConvNet library. After this stage, each person is represented by a set of deep feature vectors. These vectors are then used to learn for a classifier which will be used to re-rank the ranked list returned for location retrieval stage.

Location Search First, we retrieve shots containing the query location. Our approach is to fuse rank lists of both holistic and local feature based searching systems. For the local feature approach, we use the configuration of the state-of-the-art Bag-of-Visual-Word (BOW) framework for image retrieval. Local features of each key frame of a shot are extracted using Hessian-Affine detector and rootSIFT feature descriptor. All features gathered from database video frames are clustered using approximate K-Mean algorithm (AKM) with a very large number of codewords. Then, these features are quantized using last pretrained codebook with hard-assignment strategy. Finally, each frame is represented by a very sparse BOW feature using TF-IDF weighting scheme. For compact representing, we sum up all BOW vectors of frame of a shot to a single one. These single feature vectors are used to build inverted index which significantly boost the speed of retrieval. Top K shots returned from BOW model are then used to reranking in the next steps. One important parameter in this initial step is the threshold of top K list. K should be vary when changing query location example. When we visualize the z-score normalized distance of all query example, we found that they have the same distribution as shown in Figure 3. Intuitively, we fixed the cut off threshold for top K is -2.5.

For the holistic feature based approach, each video frame is represented by a single high level feature vector which is the output of a fully connected layer of CNN network. We use a very deep pretrained network, i.e. VGG-19, and remove the last layer which previously used for classification task. Video frames are resized and normalized before transferring to the feed forward network. The output of the network is a 4096 dimensional feature vector which is used to represent a whole video frame. Comparing two video frames is equivalent to comparing their representing feature vectors. However, using symmetric metric such as Euclidean distance (L_2) gets low performance because it takes into account all components of feature vector evenly. In fact, for each location, some of the components are important. A learning method is proposed to magnify the role of these key components.

Face feature learning for reranking The second main part of the query is person identification. Face recognition is a very popular approach to identify a

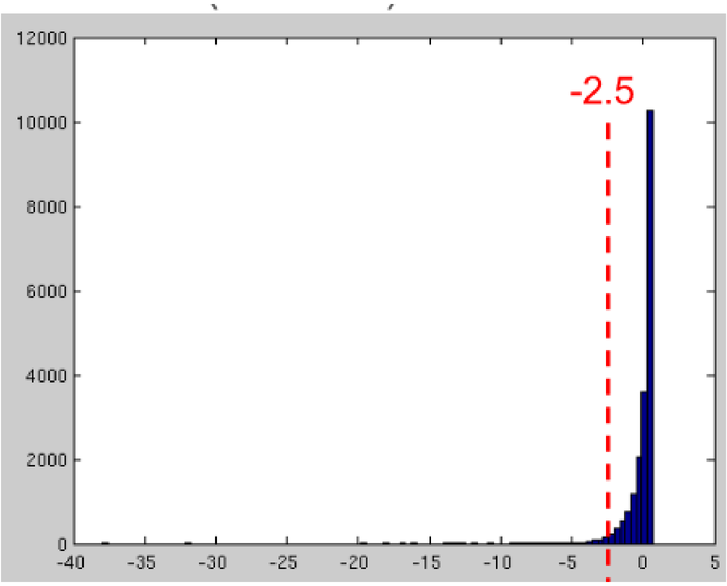


Fig. 3: Distribution of z-score normalized distance.

person. First, DPM cascade detector[8] is applied to point out locations of faces in maximum 5 keyframes per shot. Then, face images are described by a deep feature using VGG-Face descriptor[4]. After this module, each face will be represented by a 4096 dimensional feature vector. Although this feature is designed to best fit with L_2 distance metric, there still has a big gap in performance. This could be explained that, the face feature vector does not have the same weight for all components. For each face, the weights of components are different. Therefore, we propose to learn these features by a large margin classifier with linear kernel such as Linear SVM[9]. Each face candidate of a frame of a shot after transferred to classified will be scored by a value. Where positive value is equivalent to classified as positive example, and vice versa.

1.3 Our runs submitted to TRECVID INS 2016

We submitted 4 automatic runs using only image examples of locations and queries. Table 2 shows run IDs, descriptions and performances in mean average precision of 4 runs where their priority is sorted from the highest to lowest. The final result shows that, the proposed system achieved very high recognition rate but lower at recall (as shown in Figure 4). Compared to other teams, our performance is ranked at third place for fully automatic run.

There is a big gap from our best run to the first rank team. There are many reasons to explain this gap. First, we did not use any video query examples for improving the searching performance. Moreover, in many cases, target persons

Table 1: Description of submitted runs for TRECVID INS 2016

RUN-ID	Description	MAP
RUN1	RANSAC and CNN verification + Face reranking using SVM with linear kernel	0.244
RUN2	Fusion of RANSAC and CNN verification + Face reranking using L_2 dist.	0.229
RUN3	RANSAC and CNN verification + Face reranking using L_2 dist.	0.179
RUN4	RANSAC verification + Face reranking using L_2 dist.	0.231

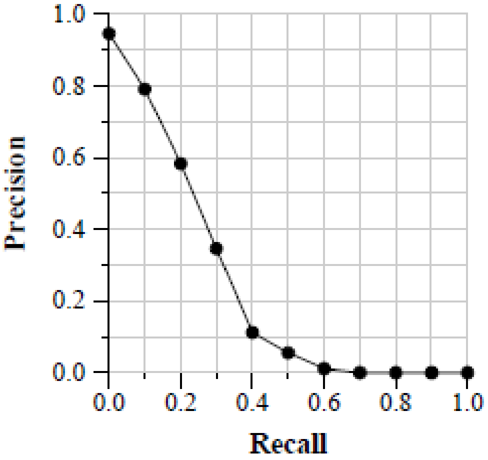


Fig. 4: Precision-recall curve of RUN1.

put their backs to the camera so that our system can not detect their faces. Therefore the final rank lists missed a lot of relevant shots.

2 TRECVID 2016 Ad-hoc Video Search: Enriching Semantic Features using Multiple Neural Networks

Abstract. Ad-hoc Video Search is a very challenging problem due to the semantic gap. A rich source of semantic information is video metadata e.g. title, summary, or textual transcript provided by video owners. However, such amount of semantic information is still far from enough to fully describe video content as it can be observed by human being. Hence, it causes low accuracy in searching videos with complex query. In this paper, we introduce an approach to enrich semantic description and presentation of a video at frame-level by using multiple neural networks pretrained on different domains. The semantic description and presentation includes object concepts, location related concepts, scene attributes, and object relationships. Compared to other teams, our system got the first rank for fully automatic run.

2.1 Introduction

With the rapid growth of video data from many sources such as social sites, broadcast TVs, films, one of the most fundamental demand is to search a particular video in huge video databases. In some cases, users did not see any target video shots before. No visual example is provided. The input query could be a text string with ad-hoc description about the content they want to search. Fig 1. gives an example of this query type, "finding shots of a man lying on a tree near a beach". In the second scenario, people already saw the target video shot and the task of the system is to to find exactly that one. In the Video Browser Showdown 2017, the dataset contains 4593 videos collected from the internet with 144GB in storage and 600 hours in duration. The participants need to solve two tasks: Ad-hoc Video Search (AVS) and Known-Item Search (KIS) corresponding to two types of query as mentioned above. In this paper, we only mention about AVS task.

To deal with AVS query type, when users describe what they are looking for by using verbal description, high-level features (i.e. semantic based features) are usually extracted to match with human language. The result of last year Video Browser Showdown has shown that, leveraging high level feature using deep convolutional neural network (CNN) is one of the-state-of the-art methods [10]. Although the performance of these neural networks are increasing every year, the number of concepts used for training is limited. On the other hand, query topics given by users are unpredictable. In this paper, we proposed combine multiple concepts from multiple datasets including ImageNet[11], Visual Genome[12], MIT Places[13] and SUN Attribute[14] to hopefully cover most popular topics that users may be interested in.

2.2 Semantic Extraction

In this section, we propose to extract semantic features to match with ad-hoc query given by users. Because the users may pay attention to any aspects of a video frame, the set of semantic concepts is unknown. Figure 5 shows an example in which users may be interested in varying from single objects e.g. the man, the beach, the coconut tree to their complex relations e.g. the man lying on the tree, the tree next to the beach.

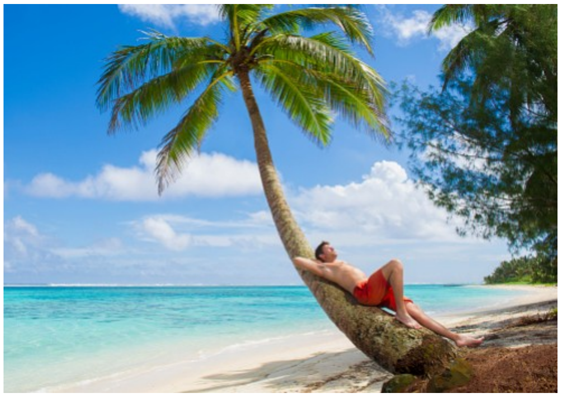


Fig. 5: Users may be interested in single objects e.g. the man, the beach, the coconut tree, or the complex relations between objects e.g. the man lying on the tree, the tree next to the beach.

Since the number of concepts is unlimited and the query of the user is unpredictable, to increase the recall of the system, we propose to extract as much semantic description and presentation of a video at frame-level as possible. The proposed system includes two main stages: i)semantic extraction using deep models trained on large scale datasets, and ii) semantic features indexing using inverted file. Figure 6 illustrates our proposed framework with two main stages.

Semantic Extraction. This is the most important part of our proposed system to detect main semantic concepts in a video frame. Inspired by recent success of deep learning techniques, in this paper, we attempt to leverage the powerful of deep features in semantic search task. Compared to low-level feature based approach, deep features (also known as high-level features) are closer to semantic based query representation and require lower storage cost. In this system, semantic concepts includes:

- Main Objects: ones that appear in a large enough region of the video frame with assumption that the higher salient object gives the higher score from the output activation of the pretrained deep convolutional neural network. In this paper, we use VGG-16 network proposed by K. Simonyan and A. Zisserman [6] to extract main objects. This is one of the state-of-the-art models for object classification on ImageNet. We sample the original video frame to overlapping 224x224 patches then transfer to the pretrained feed forward network. Feature maps from the output activation are aggregated together using average pooling approach. Five objects which give highest scores will be used to represent a video frame.

- Scene Attributes: includes indoor/outdoor labels, building, park, kitchen etc.. In our system, the attributes are extracted from the state-of-the-art models trained on MIT scene and SUN attribute dataset [13].
- Object Relationship: to describe relationships between objects, we propose to use dense captioning approach which is based on a Convolutional Neural Network-Recurrent Neural Network (CNN-RNN) to generate many sentences from the detected objects[15].
- Metadata: provided by video sharing users. Metadata includes title, summary content, and tags. They are usually about the main topic of the videos but not in detail. However, such information is helpful to improve the performance of the system by combining with other semantic concepts as mentioned above.

Building Inverted Index. After extracting semantic features, the searching task is now equivalent to text based retrieval task. This stage is to index semantic text returned from the previous stage. A standard *TF-IDF* scheme is used to calculate weight of each word. In the online searching stage, the system computes similarity scores between query text and video semantic features using inverted index structure.

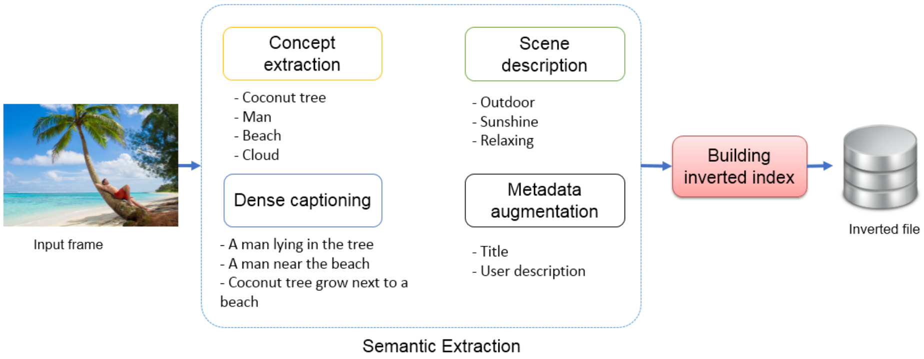


Fig. 6: Proposed system for searching based on semantic description.

2.3 Results

We submitted 4 automatic runs to this year Ad-hoc Video Search task. Table 2 shows run IDs, descriptions and performances in mean average precision of 4 runs where their priority is sorted from the highest to lowest. Our best run got first rank in fully automatic setting.

Table 2: Description of submitted runs for TRECVID AVS 2016

RUN-ID	Description	MAP (%)
RUN4	All concepts, sentences, scene attributes, metadata are merged together and index by Lucene library. We also use similarity metric of Lucene for retrieval.	**5.43**
RUN3	Similar to RUN4 but number of ImageNet concepts are extracted at maximum 5 per shot.	4.58
RUN2	This run use RUN4 as an initial rank list. We rerank topK rank list from RUN4 using TF-IDF weighting.	4.32
RUN1	Similar to RUN2 with a slightly difference in final score formula.	4.35

3 NII at TRECVID 2016 Localization

Abstract. In this paper, we present our work in TRECVID 2016 Localization (LOC). In this task, we employed the object detection with region proposal networks. We applied this framework to different network architectures and dataset combinations: provided by NIST and others public dataset. Results show the advantage/disadvantage of each configuration to the final system.

3.1 Problem

For this year, LOC task introduces 10 new concepts to localize including Animal, Bicycling, Boy, Dancing, Explosion fire, Instrumental Musician, Running, Sitting Down, Skier, Baby. It's noticeable that most of the concept is related to human and human activities when some of them can be only determined by a sequence of frames (Sitting, Running). It is quite different when comparing with the previous year when the concept was mostly static like object, animal (without action). Another difference in this year's task comes from the dataset. The development data compromises IACC.1.A-C datasets and contains only shot-level annotations while test dataset compromises IACC.2.A-C, which is all test and development dataset from previous years.

3.2 Dataset

LOC task is provided with more than 1800 shots which contains concepts. Since this year dataset includes annotation at shot-level only, preparing a object-level bounding boxes is an inevitable step. To do this, we created a annotation tools which allows user draw a bounding boxes in some key frames of video shots and generate the bounding box for the image between those key frames in the sequence using interpolation. However, for some concepts, the number of shots is very small eg: around 10 shots with Skier and some is very hard even for annotate because of the complex context or having multiple object in the scene eg: lot of people is dancing in the pub.

Other datasets In order to enrich the training data, we have collected datasets from other sources which has bounding boxes at object-level. Those datasets are:

- VOC 2012[1] for detection task, including those TRECVID's concepts: Instrumental Playing, Person, Boy, Running, Bicycling
- Activity dataset[2] for concepts: dancing, running
- ILSVRC2015 - VID dataset[3] for concept animal
- UCF101 - Action Recognition Data Set[4] for Skier and Bicycling concepts

[1] http://host.robots.ox.ac.uk/pascal/VOC/voc2012/

[2] http://vhosts.eecs.umich.edu/vision//activity-dataset.html

[3] http://image-net.org/challenges/LSVRC/2015/#vid

[4] http://crcv.ucf.edu/data/UCF101.php

– UCF Action dataset[5] for concept Running and Person

Table 3 shows the number of training images for each dataset.

Table 3: Number of training images for each dataset

	TRECVID	VOC	Activity dataset	ILSVRC	UCF101	UCFACtion
Images	54000	4568	1243	3463	3322	1076

3.3 LOC Framework

We tried to adopted framework from [16] - the winner team at ILSVRC 2015 [11] in VID challenge. This framework includes Image Object Proposal, Object Scoring, High-confidence Tracking, Tubelet Perturbation and Max Pooling, Temporal Convolution and Re-scoring. However, the tracking step is based on [17] when applying to TRECVID dataset shows low accuracy. It's mainly because with TRECVID dataset, most of the image has low quality and the changing between different scenes in the shot is often occurred which cause losing the tracker. So finally, we decided to apply only object re-scoring based on the result of object detection to remain the dominate concept in the shot.

Since most of the concept in this year is related to human and human's activities (8/10 concepts), we proposed to add an extra class named "people" to put all the person who isn't listed in 10 concepts.

We also trained the system with a different configuration. We tested with two networks: VGG-16 [18], and Resnet-50 [19] on two dataset TRECVID, TRECVID+Other dataset that mentioned in 3.2

3.4 Result and Conclusion

We used Faster R-CNN [20] with Python wrapper to perform object detection on image. As mention before, we fine tune on VGG-16 and Resnet-50 model with two different datasets. After that, we keep only object with score larger than 0.8 then make a suppressing and re-scoring the object.

We submitted three runs: NII_Hitachi_UIT.run_1 with Res-net fine tuned on TRECVID dataset only, NII_Hitachi_UIT.run_2 and NII_Hitachi_UIT.run_3 is using VGG-16 with TRECVID+other datasets. However, due to the restrict of time, with run 1, we only submit with the result is taken from a small portion of the testing dataset (about 200k/2 millions of images).

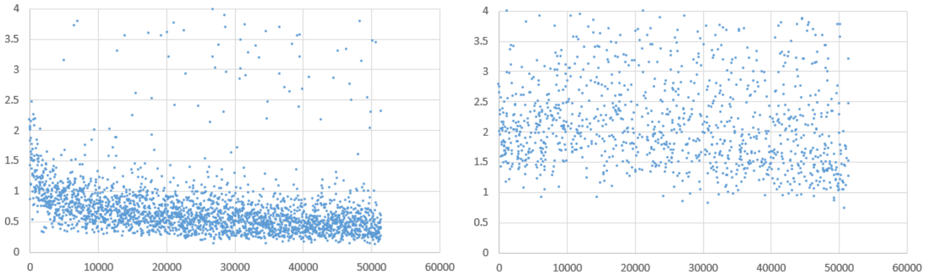
As can be seen in Fig. 7, when using both TRECVID and others datasets, the training loss is cannot converge as it has when using only TRECVID dataset.

[5] http://crcv.ucf.edu/data/UCF_Sports_Action.php

This true for both VGG-16 and Resnet-50 network. That's why our second and third run show very bad results (mostly zero).

For comparison between VGG-16 and Resnet-50, Resnet-50 show slightly higher in accuracy than VGG-16. But it's running time is much lower (1 FPS vs 7FPS). We also re-run our first submit using Resnet-50 with all image in the testing dataset and using ground truth file provided by NIST to evaluate the system. Fig.8 shows the results, our run still has low fscore in iframe level but has high fscore in mean pixel level, especially in Boy and Instrumental_musician concepts.

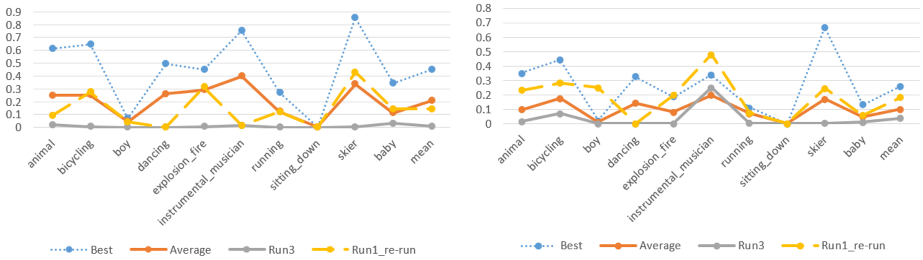
In conclusion, we have shown in this section the bad effect of using the out-domain dataset to training the system as well as the benefit of Resnet with object detection task, which is helpful when apply to TRECVID dataset. The final result is still low because we did not exploit the benefit of sequences frame to determine human action.



(a)

(b)

Fig. 7: Training loss with Resnet-50 over iteration when using different dataset (a) TRECVID (b) TRECVID+others



(a)

(b)

Fig. 8: Final result: (a) iframe fscore (b) mean pixel fscore

4 NII-Hitachi-UIT@TRECVID2016 Surveillance Event Detection

Abstract. In this paper, we present a retrospective system for surveillance event detection (SED) task in TRECVID2016. The system detects people using high-precision head detector trained by deep learning, and tracks head regions using generic object tracker. Then the system classifies person actions in tracked image sequences using spatiotemporal features obtained from 3D convolutional neural network. In SED system development, we introduced state-of-the-art deep learning methods, and fine-tuned pre-trained models by Gatwick airport videos with extra annotations to improve performances of components.

4.1 SED system overview

Fig.9 shows an overview of our SED system.

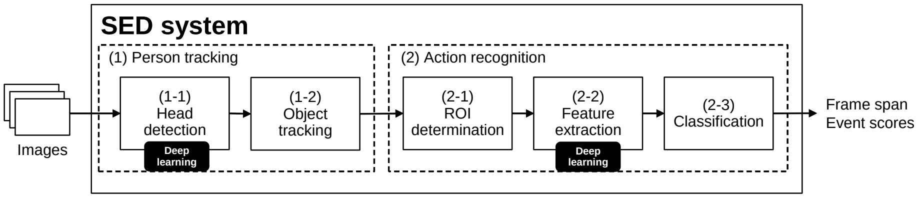


Fig. 9: NII_Hitachi_UIT surveillance event detection system.

Our SED system consists of two major components: (1) person tracking module and (2) person action recognition module. For the input image sequences the system returns frame-span and classification scores as detection results.

In the person tracking module, the system detect regions by head detector and tracks them with a generic object tracker. We introduced a deep learning method to detect objects proposed in [21]. This method achieves high-accuracy head detection in crowded scenes, by a combination of CNN-based feature extraction and LSTM-based bounding box determination. After head detection, detected head regions are associated across multiple frames using generic object detection method proposed in [22], and we can obtain temporal coordinates information of detected people.

In the action recognition module, for each detected people, the system determines ROI (Region Of Interest) images for feature extraction, and extracts spatiotemporal features from sequential ROI images to classify the action in the sequence. In ROI determination, we use head coordinates to calculate the whole body region. Also we use the same ROI in a sequence to avoid an effect of global camera motion. For event detection task in TRECVID, Improved Dense Trajectory (IDT)[23] with Fisher Vector coding [24] is known as good features to represent action in videos. In our system, we introduced 3D convolutional neural network [25] to extract spatiotemporal features. The network provides one fixed-dimensional vectors from a sequence. We used SVM solvers to learn action recognition modules.

4.2 Performance evaluation

In SED system development, we introduced state-of-the-art deep learning algorithms, and fine-tuned pre-trained models by Gatwick airport videos with extra annotation data to improve performances of person tracking and action recognition.

4.3 Fine-tuning of head detection

In our system, we use head detector by a combination of CNN and LSTM learning[21]. We used publically available pre-trained models and fine-tuned them with extra annotations (11,970 images, 82,583 head region coordinates).

Fig.10 shows performance evaluation of pre-trained and fine-tuned head detector for each camera. Average hit-rate across the camera was improved by 16.1 % using fine-tuned model. Detecting small target in scenes with depth is still difficult, and more improvement is required.

Using fine-tuned head detector and generic object tracker, we extract 120,781 sequences with 8,564,221 head regions from 10 hours evaluation videos (EVAL16).

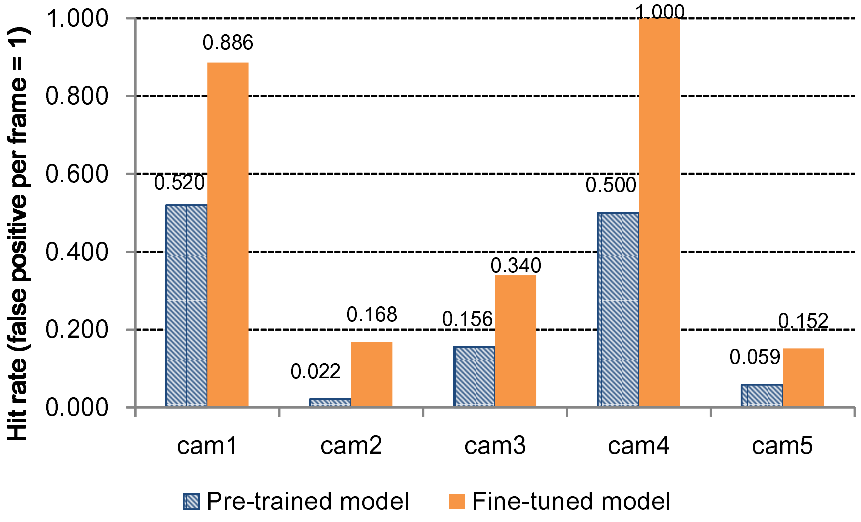


Fig. 10: Performance evaluation of head detection.

4.4 Fine-tuning of spatiotemporal features

In our system, 3D convolutional neural network proposed in [25] is used to extract spatiotemporal features for action classification. The network has five 3-dimensional convolution layers with 3d-pooling layers and three fully connected layers. The network accepts multiple images (e.g. 16 frames) and can extract spatiotemporal features. We fine-tuned Sport-1M pre-trained models training videos: 559,814 frames, 10,272 positive (10-actions defined in SED task) sequences and 12,249 negative (no action) sequences.

Fig.11 shows performance comparison of action classification using pre-trained and fine-tuned models. In this experiment, we used 4,096 outputs of intermediate fully connected layer (FC7) as spatiotemporal features, and trained SVM classifiers (RBF-kernel). Average classification accuracy was improved by 16.3 % using fine-tuned features.

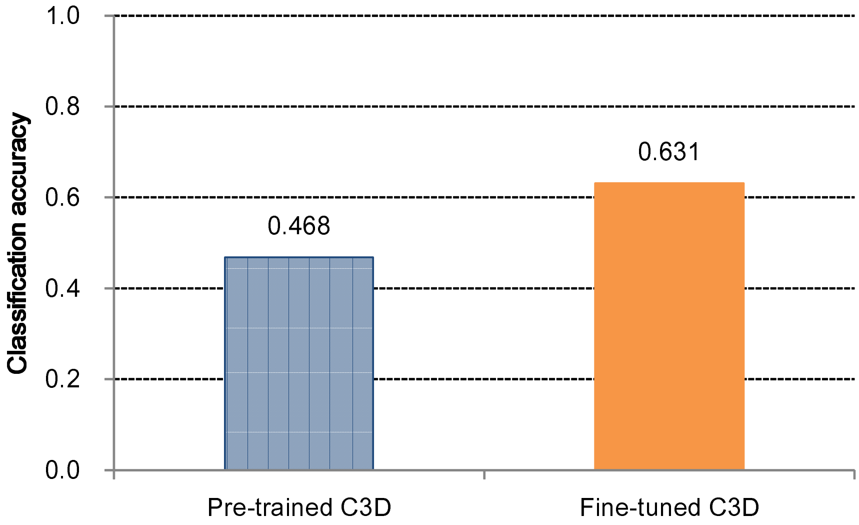


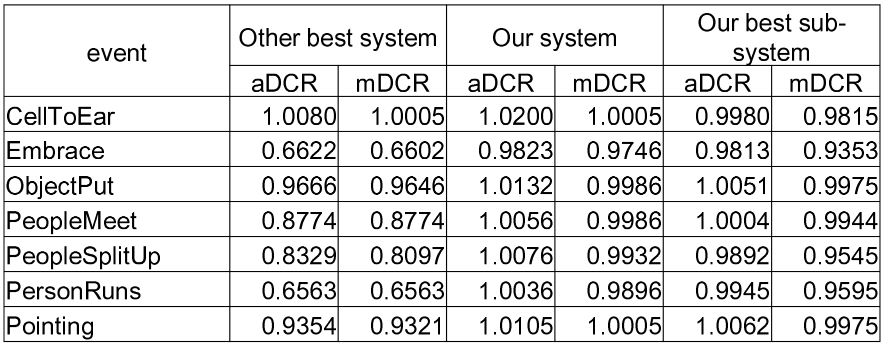
Fig. 11: Performance evaluation of action recognition.

4.5 Results

In this paper, we have presented SED system based on person tracking and person action recognition. We fine-tuned head detector and spatiotemporal feature extractor using deep learning methods.

Table 4 shows our evaluation results for EVAL16 provided by NIST, along with the best performance achieved by other participants. Although fine-tuning improves performance of each component, the overall performance is not accurate enough. Further consideration will be needed to yield any findings about ROI determination policy and a combination (or switching) of spatiotemporal and still image features.

Table 4: Evaluation results of surveillance event detection task (EVAL16).



event	Other best system		Our system		Our best sub-system	
	aDCR	mDCR	aDCR	mDCR	aDCR	mDCR
CellToEar	1.0080	1.0005	1.0200	1.0005	0.9980	0.9815
Embrace	0.6622	0.6602	0.9823	0.9746	0.9813	0.9353
ObjectPut	0.9666	0.9646	1.0132	0.9986	1.0051	0.9975
PeopleMeet	0.8774	0.8774	1.0056	0.9986	1.0004	0.9944
PeopleSplitUp	0.8329	0.8097	1.0076	0.9932	0.9892	0.9545
PersonRuns	0.6563	0.6563	1.0036	0.9896	0.9945	0.9595
Pointing	0.9354	0.9321	1.0105	1.0005	1.0062	0.9975

5 NII-Hitachi-UIT@TRECVID2016 Multimedia Event Detection

Abstract. With our system we analyze how much information the temporal component of videos carry to classify the MED events. Our system utilizes 3D convolution over the consecutive frames generating high level feature descriptors. Feature descriptors of several non-uniformly distributed timespans in the video are feed into a LSTM for final classification.

5.1 Introduction

With DeepLearning remarkable performance has been achieved over the last years for plain image classification. However, the temporal component is less researched which also hold for the submissions done in the MED task of TRECVid. We analyzed how much information the temporal component of the videos carries. One major challenge in this aspect is, that the meta data provided for the videos in MED only tells that a certain event happen in an video, but the concrete timestamps are not known. The discrepancy between unrelated and important information for an event can become arbitrary large. Simply taking all frames and extracting motion feature from them will most likely not result in a good performance. A Long Short-Term memory (LSTM) [26] seems like a good fit for this problem as it can decide which information is important and, therefore, should be remembered and which information should be skipped or forgotten. However, LSTM have not yet successfully applied for more than several hundreds of time steps. In a video with thousands of frames, some frames must be either skipped or other ways to combine them must be found. Skipping holds the risk, that important frames required to classify the event are skipped. Another approach of skipping would be to uniformly skip 1 out of 2 or 2 out of 3 frames, etc. However, some samples in the TRECVid corpus are several thousand frames long and even removing 80% of the frames do not fundamentally address our problems. Skipping more than ten consecutive can also become critical, as important motion information might get lost over so many frames. We decided to extract temporal information from consecutive frames by 3-dimensional convolution at full frame rate.

5.2 3D-Convolution

We constructed a convolutional network of 8 convolution layers followed by two Maxout layers and the logistic regression. The input for the first convolutional layers is of size 106 x 106 pixels and 22 consecutive frames. The convolutional layers compress the images to a final size of the feature maps of 3 x 3 and 2 time steps. Both Maxout layers have a 512 dimensional output. This construction follows the results published by Facebook [25], beside that the last two fully connected layers have been replaced by Maxout layers. To avoid overfitting random cropping of a 128x128 pixels image to the 106 x 106 pixels network input has been applied. Furthermore, the 22 frames input for the network are pooled out of a larger sequence of 31 frames.

5.3 LSTM

After pre-training the 3D-convolutional network the final logistic regression was dropped and the output of the last Maxout unit is used as a 512 dimensional feature descriptor. 20 such descriptors from randomly pooled timestamps of the video, but in consecutive sequence, have been feed into a LSTM unit with 256 memory cells. By averaging out the output of the LSTM in each timestep the input for a following Maxout unit is formed and a logistic regression makes the final classification of the event.

5.4 Experiments

For training of the 3D convolutional network the UCF101 dataset [27] have been utilized. This is the only outside dataset which has been utilized in our experiments. For the EX10 dataset the "DeepFeatures" of this 3D-convolutional network have been extracted and feed into the LSTM. For training this LSTM no other data beside the TRECVid corpus has been utilized.

During classification 20 uniformly distributed timespans each covering 22 frames are feed into the 3D-convolution network and the LSTM. The average performance achieved with our system is 0.67% suggesting that motion alone is not enough to achieve high performance. However, results from previous TRECVid competitions suggest that temporal information is orthogonal to spatial information and directly adds on top of systems working with spatial information.

6 TRECVID 2016 Video-to-Text: Learning Semantic Embedding for Video with 3D Convolutional Networks

Abstract. We present in this paper our results and analyses on Video-to-Text (V2T) task, which is a new pilot task in TRECVID 2016.
For the matching task, we employ 3D Convulutional Neural Networks (3D-CNN) to generate a video representation, and use ParagraphVector approach to build a representation for sentence with the same dimensionality. We minimize the distance between two representation to build an unique representation for *semantic video embedding.*
For the description generation task, we use a multimedia embedding that is a combination of multimedia features extracted from frames, spatial-temporal volumes and also from audio segments. Moreover, we also employ the temporal attention mechanism in the language model to generate better video descriptions.

6.1 Subtask 1: Matching and Ranking

Problem V2T task [28] promotes the following problem: Given N videos $\{\mathbf{v}_1, \mathbf{v}_2, \dots \mathbf{v}_N\}$, and N description sentences of these videos $\{\mathbf{c}_1, \mathbf{c}_2, \dots \mathbf{c}_N\}$, the task is generating N pairs $(i, j_i), i = 1, 2, \dots, N$ so that, sentences $\mathbf{c}_{j_i}$ describes the content in video $\mathbf{v}_i$.

Dataset V2T task is provided with 200 videos for training and 1915 videos for blind evaluation. Each videos has two descriptions, which is divided into two sets (A and B). Participants is asked to submit four runs in each set A and B, therefore, there are eight runs in total. To train the 3D-CNN [25], we use the provided training set of 200 videos and the Youtube2Text dataset [29], which contains 1970 videos. To train the ParagraphVector, we use the SBU image captions dataset [30], which consists of one million image captions, and the captions provided with Youtube2Text (YT2T) and the V2T training set.

Methodology Our approach is based on the ECNN [31], which also uses a 2D-CNN to extract deep features from each frames in video, then use a mean pooling to generate the input of the 2D-CNN, and also adopts a ParagraphVector with

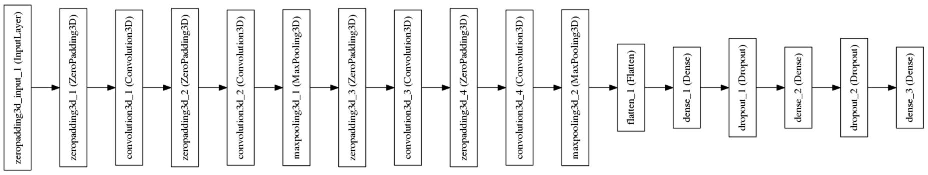

zeropadding3d_input_1 (InputLayer)
zeropadding3d_1 (ZeroPadding3D)
convolution3d_1 (Convolution3D)
zeropadding3d_2 (ZeroPadding3D)
convolution3d_2 (Convolution3D)
maxpooling3d_1 (MaxPooling3D)
zeropadding3d_3 (ZeroPadding3D)
convolution3d_3 (Convolution3D)
zeropadding3d_4 (ZeroPadding3D)
convolution3d_4 (Convolution3D)
maxpooling3d_2 (MaxPooling3D)
flatten_1 (Flatten)
dense_1 (Dense)
dropout_1 (Dropout)
dense_2 (Dense)
dropout_2 (Dropout)
dense_3 (Dense)

Fig. 12: Our 3D-CNN architecture

Table 5: V2T results on several test sets

Run name	Train set	Test set A	Test set B
Run 1 (Model A)	0.181	0.005	0.005
Run 2 (Model B)	–	0.006	0.004
Run 3 (Model C)	–	0.004	0.004
Run 4 (Model D)	–	0.005	0.003

Bag-of-Words approach to generate sentence representation. Finally, the ranking loss to minimize the distance between these representations to learn the optimal embedding space. In this submission, we replace the 2D-CNN with a 3D-CNN which is illustrated in Figure 12. The input of the 3D-CNN is a $M \times D \times D$ tensor, which is generated as follows: First, a video is divided into M chunks, from each chunk, we take a frame from the middle of chunk and resize it to has size $D \times D$. The final input tensor is a stack of these M resized frames.

Result and discusssion **Implementation details** To learn the 3D-CNN, we employ the 3D Convolutional Layer, which is available in Keras[6]. To train the ParagraphVector [32], we use the implementation in gensim[7]. Our implementation is publicly available[8]. **Result** We submitted the following four runs: 1. **Model A**: 3D-CNN w/o 3D-Dropout + Doc2Vec model learned on SBU captions; 2. **Model B**: 3D-CNN with 3D-Dropout + Doc2Vec model learned on SBU captions; 3. **Model C**: 3D-CNN with 3D-Dropout + Doc2Vec model learned on SBU captions + Youtube2Text captions (+ Early stopping); and 4. **Model D**: 3D-CNN with 3D-Dropout + Doc2Vec model learned on YT2T captions + V2T captions (+ Early stopping). Early stopping is a technique to judge the validation loss in training process. After n epochs which do not improve the validation loss, the training process is terminated. Model C and D was trained on a merged set of YT2T dataset and V2T train set, thus, the total number of training video is 2170. The mean Inverted Rank (MIR) is used to evaluate the raked list return by our algorithm. The results on blind test sets A and B, as well as our preliminary results on a train set (a subset of 20 videos which is not used in training processes) are presented in Table 5. Compared to other teams' results, our results is ranked at 6-th over 7 participant teams. **Discussion** MIR of our method is approximately 1/200, i. e., averagely, the ground truth items are ranked at 200-th over 1915 items. Compared to the result on traing set, our method is not generalized well on test set. As our observations, the reason for this degradation is the training processes of Model B, C, D were overfitting. Training losses at the end of epochs were decreased, but inside each epochs, the losses were varying and were unstable.

[6] https://keras.io/layers/convolutional/#convolution3d

[7] https://radimrehurek.com/gensim/models/doc2vec.html

[8] https://github.com/marker68/video-matching

6.2 Subtask 2: Description Generation

Problem In this subtask, our system is required to generate a natural language sentence to describe a given video, without mining knowledge of the provided descriptions in the previous task.

Methods We use three features: ResNet [19], C3D [25], and audio MFCC which represents for three main different streams in video. For each feature, we apply a linear layer to learn an embedded vector that has 512 dimension. We combine these three features by concatenating its embedded vector that results in a 1,536-dimensional vector.

We follow the temporal attention network described in [33] for video description generation task. This is a modified version of the encoder-decoder network that presented in [34], where the decoder can adaptively learn a weighting combination of features over all frames, instead of just summing as in the former work. We decompose a video into 6 equally-sampled chunks for the temporal attention experiments. Our network is illustrated in Fig. 13.

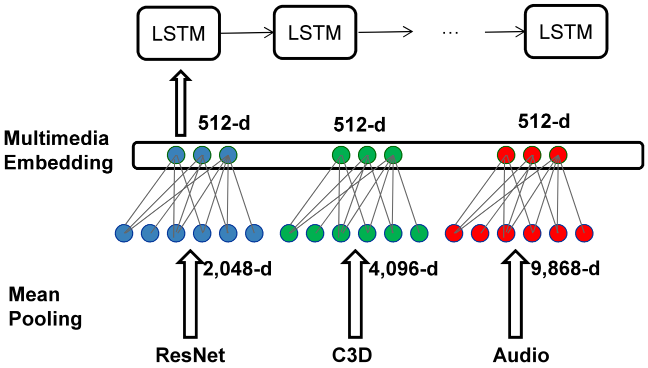


Fig. 13: Our video description architecture

Results We train our captioning model on MSRVTT dataset [35], which contain around 6,500 videos. We use the provided 200 training videos for validation and test the selected model on the provided test set.

Results of our description task is presented in Table 6. On the validation set, we show the results in terms of Bleu_4, METEOR, ROUGH_L, and CIDEr. We only obtain the results in terms of METEOR on the test set. Our combining run (Run 1) performs the best on the validation set in all measurement metrics. This confirms the benefit of combining multimodal features for the description task. Our C3D feature does not perform well on the test set because of a bug in the

Table 6: Results of the video captioning task

Run	Model	Validation Results				Test Results
		Bleu_4	METEOR	ROUGH_L	CIDEr	METEOR
Run 1	ResNet+ C3D+ MFCC	0.111	0.158	0.393	0.412	0.185
Run 2	ResNet	0.092	0.153	0.377	0.369	0.185
Run 3	C3D	0.099	0.150	0.378	0.346	0.161
Run 4	MFCC	0.065	0.126	0.338	0.174	0.178

feature extraction step. Surprisingly, audio MFCC itself achieves a rather good performance on the test set, compared to its low performance on the validation set.

References

1. J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proceedings of the International Conference on Computer Vision*, Oct. 2003, vol. 2, pp. 1470–1477.
2. J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007.
3. R. Arandjelović and A. Zisserman, "All about VLAD," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.
4. O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *British Machine Vision Conference*, 2015.
5. Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Neural Information Processing Systems (NIPS)*, 2015.
6. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.
7. Lior Wolf, Tal Hassner, and Itay Maoz, "Face recognition in unconstrained videos with matched background similarity," in *in Proc. IEEE Conf. Comput. Vision Pattern Recognition*, 2011.
8. M. Mathias, R. Benenson, M. Pedersoli, and L. Van Gool, "Face detection without bells and whistles," in *ECCV*, 2014.
9. Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin, "Liblinear: A library for large linear classification," *J. Mach. Learn. Res.*, vol. 9, pp. 1871–1874, June 2008.
10. Kai Uwe Barthel, Nico Hezel, and Radek Mackowiak, *Navigating a Graph of Scenes for Exploring Large Video Collections*, pp. 418–423, Springer International Publishing, Cham, 2016.
11. 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)*, vol. 115, no. 3, pp. 211–252, 2015.
12. Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalanditis, Li-Jia Li, David A Shamma, Michael Bernstein, and Li Fei-Fei, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," 2016.
13. Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva, "Learning deep features for scene recognition using places database," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds., pp. 487–495. Curran Associates, Inc., 2014.
14. Genevieve Patterson and James Hays, "Sun attribute database: Discovering, annotating, and recognizing scene attributes," in *Proceeding of the 25th Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.
15. Justin Johnson, Andrej Karpathy, and Li Fei-Fei, "Densecap: Fully convolutional localization networks for dense captioning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.
16. Kai Kang, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang, "Object Detection from Video Tubelets with Convolutional Neural Networks," apr 2016.

17. Lijun Wang, Wanli Ouyang, Xiaogang Wang, and Huchuan Lu, "Visual tracking with fully convolutional networks," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 11-18-Dece, pp. 3119–3127, 2016.

18. Karen Simonyan and Andrew Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recoginition," *Intl. Conf. on Learning Representations (ICLR)*, pp. 1–14, 2015.

19. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, 2016.

20. Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," jun 2015.

21. Russell Stewart, Mykhaylo Andriluka, and Andrew Y Ng, "End-to-end people detection in crowded scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2325–2333.

22. João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista, "High-speed tracking with kernelized correlation filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583–596, 2015.

23. Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu, "Action recognition by dense trajectories," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on.* IEEE, 2011, pp. 3169–3176.

24. Florent Perronnin and Christopher Dance, "Fisher kernels on visual vocabularies for image categorization," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on.* IEEE, 2007, pp. 1–8.

25. Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri, "Learning Spatiotemporal Features with 3D Convolutional Networks," in *2015 IEEE International Conference on Computer Vision (ICCV).* 2015, pp. 4489–4497, IEEE.

26. Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

27. Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah, "Ucf101: A dataset of 101 human actions classes from videos in the wild," *arXiv preprint arXiv:1212.0402*, 2012.

28. George Awad, Jonathan Fiscus, Martial Michel, David Joy, Wessel Kraaij, Alan F. Smeaton, Georges Quenot, Maria Eskevich, Robin Aly, and Roeland Ordelman, "Trecvid 2016: Evaluating video search, video event detection, localization, and hyperlinking," in *Proceedings of TRECVID 2016.* NIST, USA, 2016.

29. Sergio Guadarrama, Niveda Krishnamoorthy, Girish Malkarnenkar, Subhashini Venugopalan, Raymond Mooney, Trevor Darrell, and Kate Saenko, "YouTube2Text: Recognizing and Describing Arbitrary Activities Using Semantic Hierarchies and Zero-Shot Recognition," in *2013 IEEE International Conference on Computer Vision (ICCV).* 2013, pp. 2712–2719, IEEE.

30. Vicente Ordonez, Girish Kulkarni, and Tamara L. Berg, "Im2text: Describing images using 1 million captioned photographs," in *Neural Information Processing Systems (NIPS)*, 2011.

31. Xiaoshan Yang, Tianzhu Zhang, and Changsheng Xu, "Semantic Feature Mining for Video Event Understanding," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 12, no. 4, pp. 1–22, Aug. 2016.

32. Quoc V Le and Tomas Mikolov, "Distributed Representations of Sentences and Documents.," *ICML*, 2014.

33. Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher Pal, Hugo Larochelle, and Aaron Courville, “Describing videos by exploiting temporal structure,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4507–4515.
34. Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond J. Mooney, and Kate Saenko, “Translating videos to natural language using deep recurrent neural networks,” in *NAACL HLT 2015, Colorado, USA, May 31 - June 5, 2015*, 2015, pp. 1494–1504.
35. Jun Xu, Tao Mei, Ting Yao, and Yong Rui, “Msr-vtt: A large video description dataset for bridging video and language,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.