Universitat Politècnica de Catalunya
Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona
Grup de Processament de la imatge

Online Action Detection

Junting Pan

Advisors: Xavier Giró-i-Nieto and Shih-Fu Chang

A thesis submitted in fulfillment of the requirements for the degree of the
Master in Telecommunications Engineering

Barcelona, October 2017

Abstract

In online detection, the objective is to detect the start of an action in a video stream as soon as it happens. It is an important yet challenging problem. In many realistic scenarios, we need to detect the action before the action is completed. For example, in the autonomous driving system, it is crucial to detect whether the pedestrian is crossing the street well in time in order to make a decision to stop or to reduce the velocity.

Online action detection is a very challenging task in many aspects. It is very hard to predict the start of action for three reasons: First, the background is very diverse. Moreover, there is only a few action instance in a very long video. Last but not least, the model only observes part of the action to predict.

To address those challenges, we propose a framework for online action detection and simulate experiments on a large-scale untrimmed video dataset. With the proposed method we have obtained very competitive performance. We also proposed a new evaluation metric for online detection models: Point mean Average Precision (Point mAP), a more appropriate metric than the existing evaluation metrics that have been designed for action detection in an offline setting. We have conducted experiments on THUMOS'14 dataset of video analysis where our proposed model achieved the state-of-the-art performance on the online action detection task.

Keywords: Video analysis, Online Action Detection, Deep Learning, Convolutional Neural Networks, Generative Adversarial Networks.

Acknowledgements

First of all, I want to express sincere gratitude to my advisors, Xavier Giro-i-Nieto and Prof. Shih-Fu Chang. I would thank Xavi for introducing me to the world of Machine learning and Computer Vision; for giving me the opportunities to participate in many research projects; for encouraging me and supporting me. Not only I have learned research abilities but also his optimistic attitude which really helps to overcome the difficult moments during the development of this project. I would also thank Prof. Chang who gives the chance to become a member of the DVMM lab, one of the best research labs in the world in the field of Multimedia. Thanks for his patience and guidance, for teaching me how to become a researcher and how to think about the problems as well as how to solve them.

I would also like to thank all colleagues in the DVMM lab for their insightful discussions. Thanks to them for their companion during those nights that we work together until the sun rises. Thanks to them for make my stay at New York become one of the best experiences in my life. I really enjoyed the chat at lunchtime and the songs that we sang together when we walk back to home.

Finally, I would like to thank my family always supporting me in an unconditional way. Thanks for being there whenever I need help. Thanks for everything.

Thank you all!

Revision history and approval record

Revision	Date	Purpose
0	15/09/2017	Document creation
1	7/10/2017	Document revision
2	15/10/2017	Document revision
3	19/10/2017	Document approbation

DOCUMENT DISTRIBUTION LIST

Name	e-mail
Junting Pan	junting.pa@gmail.com
Xavier Giró i Nieto	xavier.giro@upc.edu
Shih-Fu Chang	sc250@columbia.com

Written by:		Reviewed and approved by:		Reviewed and approved by:	
Date	15/09/2017	**Date**	19/10/2017	**Date**	19/10/2017
Name	Junting Pan	**Name**	Xavier Giró i Nieto	**Name**	Shih-Fu Chang
Position	Project Author	**Position**	Project Supervisor	**Position**	Project Supervisor

Contents

1 Introduction **8**

2 Related work **11**

2.1 Online Action Detection 11

2.2 Early Action Detection 11

2.3 Offline Action Detection 12

2.4 Generative Adversarial Networks 12

3 Convolutional Neural Networks **14**

4 Generative Adversarial Networks **16**

4.1 Generative Adversarial Networks (GAN) 16

4.2 Semi-Supervised Generative Adversarial Networks 16

5 Methodology **18**

5.1 Network Architecture 18

5.1.1 Prediction Network 18

5.1.2 Generative Adversarial Network Module 18

5.2 Training 19

5.2.1 Cross-entropy loss 19

5.2.2 L2 similarity loss 19

5.2.3 GAN loss 20

5.3 Prediction and post-processing 21

6 Experiments **23**

6.1 Experiment setup 23

6.2 Evaluation Metric 23

6.3 Results 24

7 Conclusions **26**

List of Figures

1.1 Illustration of an online action detection prediction.[2] 8
1.2 Online detection Pipeline 9

2.1 Visual feature anticipation. 11
2.2 CDC for temporal action localization. 12
2.3 DCGAN for image generation. 13

3.1 Mathematical model of a neuron inside the Neural Network.[11] 14
3.2 Neural networks 14
3.3 Convolutional Neural Networks 15
3.4 2D and 3D convolution operations 15

4.1 Generative Adversarial Networks. 16
4.2 Comparison between different types of GANs. 17

5.1 Hard sampling strategy. 19
5.2 Similarity between temporal windows. 20
5.3 Model architecture for supervised learning. 20
5.4 GAN learning. 21

6.1 Per class Average Precision (AP) when time error tolerance is 10s. 25
6.2 mAP of action start detection from 1s to 10s 25

List of Tables

6.1 Average $PmAP$ from time error tolerance=1s to time error tolerance=10s. . . . 24

Chapter 1

Introduction

The goal of online action detection is to detect the start of action as soon as it happens in an input video. This task involves two parts: First, detecting the change from background to action. Second, recognizing the action class category.

Online action detection is important for several real-world applications such as video surveillance, self-driving car, and robotic applications to name just a few. In those scenarios, it is crucial to detect the action with minimal latency. As an example, video surveillance systems must detect the abnormal situations well in time in order to activate emergency alerts.

Figure 1.1: Illustration of an online action detection prediction.[2]

In online action detection, a good model should be able to detect the action right after it starts, which means only part of the video is observed at the time of prediction, unlike the offline action detection, where the prediction can be made based on the observation of the whole action instance. Moreover, background samples are very diverse in the real world, therefore it is very hard to detect the action start based on the knowledge of the historical data. Additionally, at the very beginning of the action, some actions may look like very similar at the very beginning. For example, long jump and high jump, shot put andthrow discus.

Most of the existing video analysis models focus on offline event detection or temporal action localization. The major difference between online and offline action detection is that in the offline setting, the whole video is available to detect when the event starts. Whereas for online action detection, the input is a video stream, and ideally the prediction is made only based on part of the video and before the action is completed. This property has made the online action detecting task very challenging.

Convolutional Neural Networks (ConvNets) are one of the most widely used deep learning architecture, it has shown impressive performance in tasks such as object recognition, object

detection and many other tasks in the field of computer vision [3, 29, 23, 30, 14, 6, 18, 26]. A ConvNet is composed by multiple convolutional layers with learnable filters that perform spatial convolution over the input and they are usually followed by a non-linear activation function e.g. sigmoid or Rectified linear Unit (ReLU). While ConvNets are the best choice for image level task, 3D ConvNets [25] with kernels that convolve simultaneously in time and in space, was introduced to model both appearance and motion, in video understanding tasks.

In this thesis, we propose a new method to tackle the online action detection problem using deep learning architectures based on 3D ConvNets [25]. During training, the network observes a set of short video clips of 16 frames, i.e half a second of video, and its parameters are learned by minimizing an objective function defined over the model output and ground truth label. During testing, the input is a video stream, given the current frame and 15 previous frames, the model is capable to predict the action class label in the current frame. An action start is detected when the change of the predicted class category is of the form: (1) background to action or (2) action A to action B.

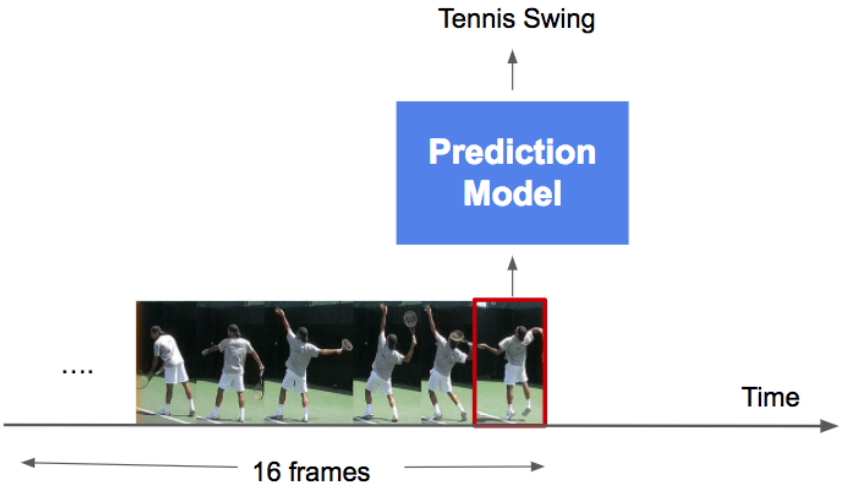


Figure 1.2: Pipeline for our online action detection model. The frame marked in red is the current frame. The prediction model predicts the class label of the last frame in a 16 frames window.

In addition to the detection framework, we also proposed a new evaluation metric Point Average Precision. Unlike the frame level Average Precision(AP), the newly proposed metric only takes into account the action start frame since the objective of online action detection is only to detect the action start and not per frame label.

In our experiments, we have trained and tested on a large scale video datasets THUMOS'14 [9], the results show that the proposed model significantly outperforms all baseline method, achieving state-of-the-art performance.

In particular, the main contributions of this project are as follows:

- We proposed a novel framework for online action detection.
- Our proposed model has shown a consistent improvement over our baselines.
- We introduced a new evaluation metric for online action detection.

The source code used for this project can be found in Github:

https://github.com/junting/online-action-detection

Chapter 2

Related work

2.1 Online Action Detection

Different from traditional offline action detection, this problem has not attracted much attention since recently. De Geest *et al.* [2] introduced three baselines, the first one is based on Fisher Vector and improved trajectories; the second one is a ConvNet that operate on a single frame; the last one is an LSTM network [8], which has the capability to model temporal pattern of the video.

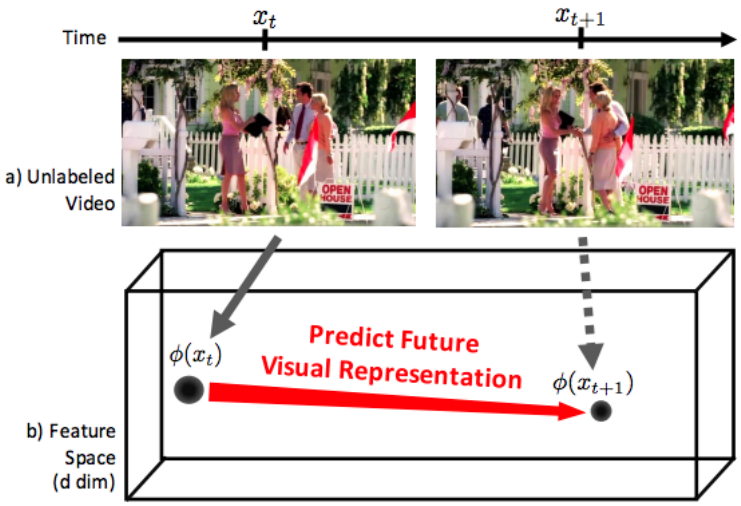


Figure 2.1: Predicting Representations: The model aims to anticipate the visual representation of frames in the future. [27]

2.2 Early Action Detection

Our problem is closely related to early action detection problem. Where the machine needs to predict the future action class category [13], or future action representation [27] before the action starts. Most of these works focus on recognizing action class that is going to happen next rather than precisely detect when the action has started.

2.3 Offline Action Detection

This problem is first formulated by Gaidon *et al.* [5], where a long untrimmed video is given, and the goal is to temporally localize the action. Early research only focused on limited action categories. Later, many large-scale datasets with more action diversity were introduced, such as THUMOS [9], MEXAction2 [1] and ActivityNet [4]. This has enabled the use of deep learning techniques to be applied in such task. 3D ConvNets and Recurrent Neural Networks (RNNs) are very popular choice to model the temporal connections over the video frame sequence. Yeung *et al.* [28] has incorporated reinforcement learning together with RNN to detect action boundary by only observing a few frames, but it assumes that the whole video is given in order to select which frame to use. Shou *et al.* [22] introduced an end to end segment based 3D ConvNets (S-CNN); Shou *et al.* [21] has recently proposed Convolutional-De-Convolutional (CDC) networks, where CDC filters are used with 3D convolutional filters to extract frame level semantic information, it has been proved to be very effective to precisely localize boundaries of action instances. Our work differs from all approach mentioned above in the sense that, for offline action detection the whole video is given to find the temporal boundaries, in the contrast, our approach only requires partial data to detect the action.

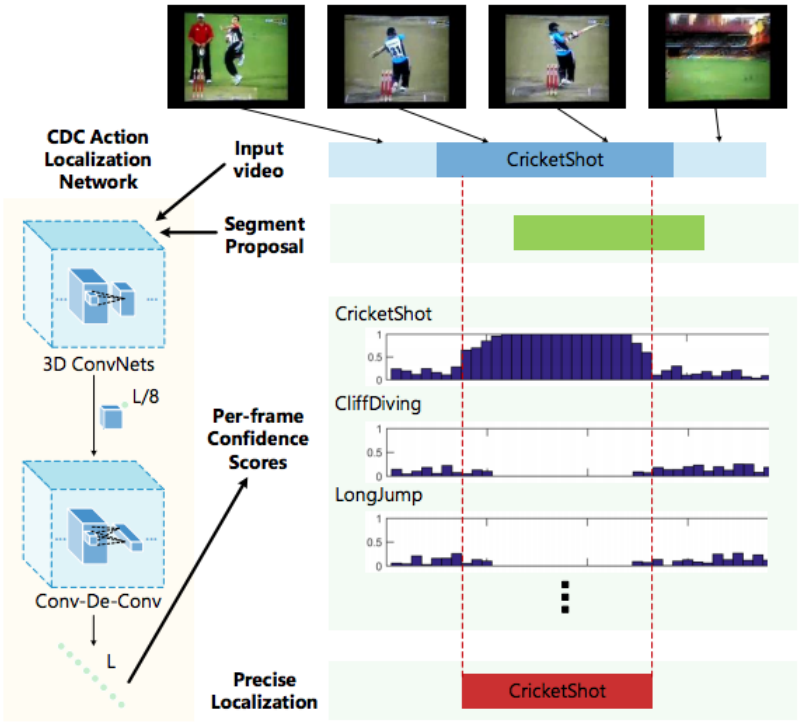


Figure 2.2: In CDC [21], given an input raw video, it predicts per frame class label.

2.4 Generative Adversarial Networks

Generative Neural Networks was first presented by Goodfellow *et al.* [7]. The idea is to alternatively train two networks that have opposite objective functions. First, the generator network aims to generate images that are indistinguishable from real images, thus it can fool

the discriminator networks. The objective of the discriminator networks is to correctly classify whether the samples are real or is generated by the generator network. Recently, there is a burst in the research of Generative Adversarial Networks. Radford *et al.* [19] has proposed DCGAN, an deep ConvNets architecture that can successfully generate realistic scene images. Moreover, they have shown the filters learned at the discriminator useful for image classification task. In [10], the use of GAN has been explored for image to image translation task, where the generator network maps the image from the source domain X to a target domain Y, and the role of the discriminator is to detect the difference between the generator's output $G(X)$ and real samples Y. Concurrently, Pan *et al.* [17] and Luc *et al.* [15] have applied adversarial networks to enhance performance in supervised learning tasks. In our work, we used GAN during training to generate hard negative samples in the feature space to improve our classification accuracy.

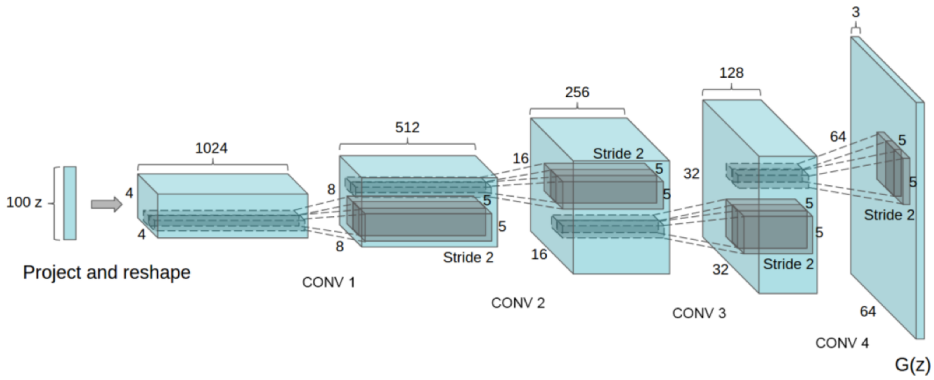


Figure 2.3: A 100 dimensional uniform distribution Z is first projected to a small spatial extent convolutional representation with many feature maps, and then this high level representation is converted into a 64×64 pixel image [19]

Chapter 3

Convolutional Neural Networks

Neural Networks are machine learning training algorithms which exploit multiple layers of non-linear information processing for pattern analysis and classification, being inspired by how the human brain works. Neuron is the basic computational unit, each neuron performs a dot product with the input and its weights, then it adds the bias and applies the activation function (non-linearity).

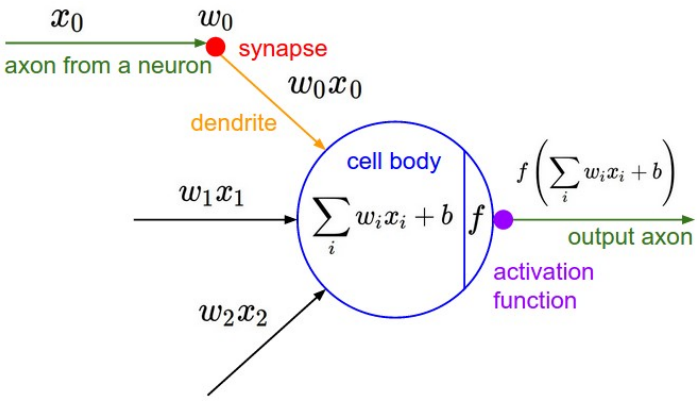


Figure 3.1: Mathematical model of a neuron inside the Neural Network.[11]

Neural Networks are modeled as layers (grouped neurons) that are connected in a cyclic graph. This layered architecture enables very efficient computation based on matrix multiplications interwoven with the application of the non-linearity.

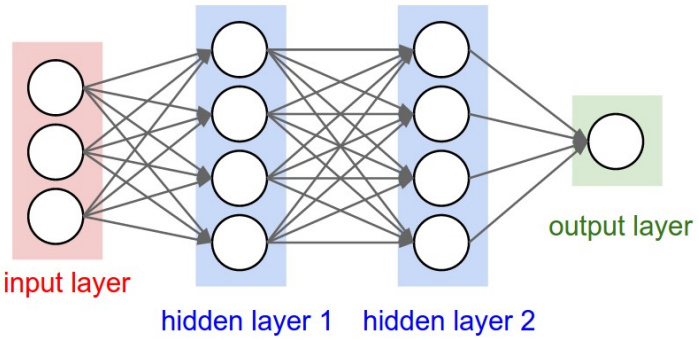


Figure 3.2: A 3-layer Neural Network with 3 inputs and 1 output. [11]

Once the architecture is chosen, in order to train Neural Networks, backpropagation is applied to compute the gradients on the connections of the networks, with respect to a loss function.

Convolutional Neural Networks are a class of Neural Networks which are made to take image inputs. The layers of a ConvNet have neuron organized in 3 dimensions: width, height, depth. So that each layer accepts and 3D input and transforms it to a 3D output. Due to the overfitting problem caused by the millions of parameters of the ConvNet, it usually uses the same weight vector for each single depth slice, then the forward computation of the layers in each depth slice is computed as a convolution of neuron's weights and the input.

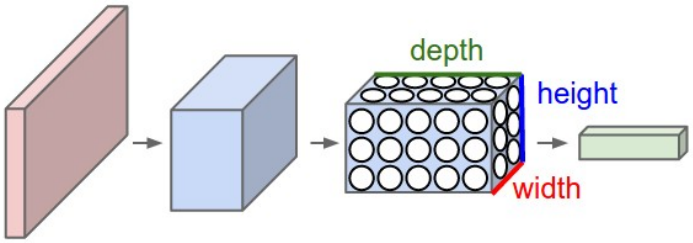


Figure 3.3: A ConvNet arranges its neurons in three dimensions (width, height, depth), as visualized in one of the layers. Every layer of a ConvNet transforms the 3D input volume to a 3D output volume of neuron activations. [11]

There are three main types of layers to build ConvNet architectures:

- **Convolutional Layer:** Is the core building block of the network. Its parameters consist of a set of learnable fileters. A dot product is compute between the filters and the input, the ConvNet will learn filters that activate when some specific type of feature in the input is detected.
- **Pooling Layer:** It reduces the spatial size of the input in order to diminish the number of parameters and computation in the ConvNet.
- **Fully Connected Layer:** Neurons between two adjacent layers are fully pair wise connected, while in neurons from the same layer are not.

While 2D ConvNets are suitable to extract image features by performing spatial convolution, 3D ConvNets can also learn video features performing spatiotemporal convolution that can model appearance and motion simultaneously. In figure 1 illustrates the difference between 2D convolution and 3D convolution.

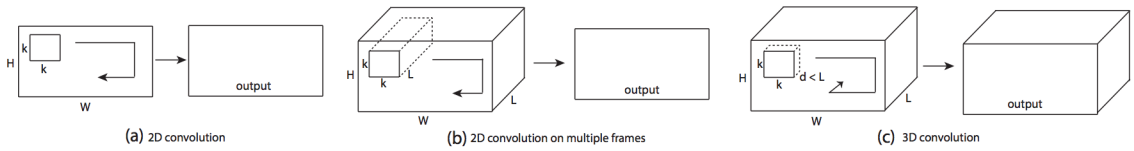


Figure 3.4: 2D and 3D convolution operations. a) Applying 2D convolution on an image results in an image. b) Applying 2D convolution on a video volume (multiple frames as multiple channels) also results in an image. c) Applying 3D convolution on a video volume results in another volume, preserving temporal information of the input signal. [25]

In [25], a 3D ConveNets (C3D) were trained on large scale video dataset for video classification tasks. The network takes 16 frames as input and after several 3D convolutional and pooling layers, the network predicts the class score of the input clip. It has been demonstrated to work remarkably well for many video analysis task. Later on, pretrained C3D was widely used to extract video feature for many transfer learning tasks [22, 21, 16].

Chapter 4

Generative Adversarial Networks

4.1 Generative Adversarial Networks (GAN)

Generative Neural Networks (GAN) is first proposed by Goodfellow *et al.* [7]. In GAN, training is driven by two competing agents: first the generator synthesizing samples from random noise that match with the training data: second, the discriminator distinguishing between a real sample drawn directly from the training data and a fake one synthesized by the generator.

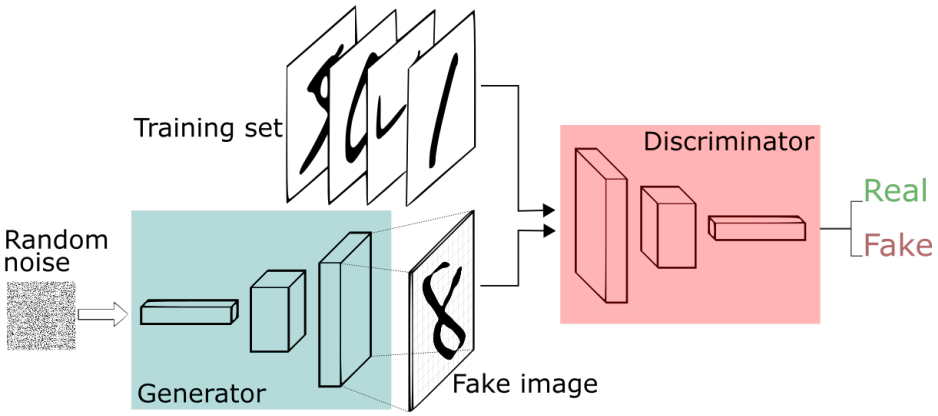


Figure 4.1: Generative Adversarial Networks. The discriminator tries to distinguish real images from fake images. The generator turns random noise into imitations of the image attempting to fool the discriminator.

The two networks are alternatively trained, the generator is trained to maximize the uncertainty of the discriminator network; the discriminator is optimized to discriminate samples from the real data distribution and synthesized sample distribution. As the result, the generator learns the real sample distribution, the discriminator can no longer distinguish between samples that come from the two different sources.

4.2 Semi-Supervised Generative Adversarial Networks

GAN can also be applied in semi-supervised learning [20]. GAN model can be added to any classifier with a certain number of classes e.t. K classes. In this case, all the samples generated by G can be treated as class $y = K + 1$, that correspond to the fake class in the original GAN.

The loss function of semi-supervised GAN can be split into two parts, the first part is the standard supervised classification loss, and the second part is the unsupervised GAN loss.

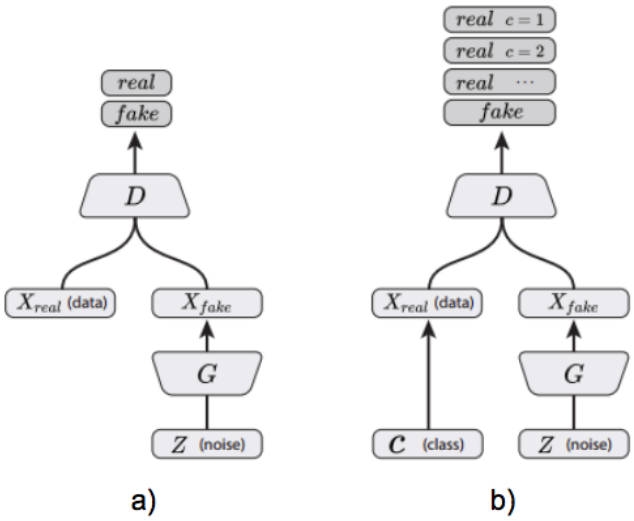


Figure 4.2: Comparison between a) original GAN and b) Semi-supervised GAN

Chapter 5

Methodology

This section provides a detailed description of our proposed framework for online action detection, including Network Architecture, details of training and post Processing.

5.1 Network Architecture

The proposed framework is base on two components: prediction network to predict the action class category and an auxiliary generative adversarial network for improving the classification accuracy. This section provides details on the structure of two modules and the considered objective functions.

5.1.1 Prediction Network

Our deep network is based on the C3D [25] network, because of it has shown promising results in many video analysis tasks, such as action detection, action recognition, and temporal localization. C3D is a 3D ConvNets which includes 3D convolution layers and 3D max-pooling layers, and it was demonstrated that it can simultaneously extract both spatial and temporal feature. The network is composed of eight 3D Convolutional layers interspersed with five 3D pooling layers and followed by three fully connected layers. All 3D Convolutional layers have filter size 3 and stride 1 in all three dimensions, and all pooling perform a spatial downsampling by a factor of two while having some temporal downsampling variation. The network takes a window of 16 frames of size 112×112 input and outputs the probability distribution over all $(K + 1)$ action classes. The network is initialized with the weights of the C3D trained on the Sports-1M [12] for the video classification task.

5.1.2 Generative Adversarial Network Module

The GAN module is composed of two networks, the generator network intents to generate hard negative samples. The discriminator D takes a synthetic sample or real sample as input and outputs the probability distribution over the two sample sources. In our work, instead of generating samples in the image space like [20], we generate samples at the feature space. The generator network consists of three fully connected layers. Each of the three fully connected layer is followed by Leaky ReLU activation, with the exception of the final layer, which uses a ReLU

activation. The discriminator network shares the weights of the last three fully connected layer of the prediction network. Thus, we use the last three layers of the prediction network as the discriminator network.

5.2 Training

The large variety of negative data have made the online action detection task very challenging, the positive samples occupy very small part of the video. To combat with the unbalanced data problem, we use the following strategy: we densely sampled around the start boundary of the action in each training batch. As shown in Figure 5.1, half of the sample of a training batch are windows containing the start of an action and the other half are the rest part of the video.

The whole network was trained in two stages. At the first stage, the prediction network is trained with cross-entropy loss and an L2 similarity loss. At the second stage, the adversarial module is added, and all three loss terms are jointly optimized (supervised and unsupervised).

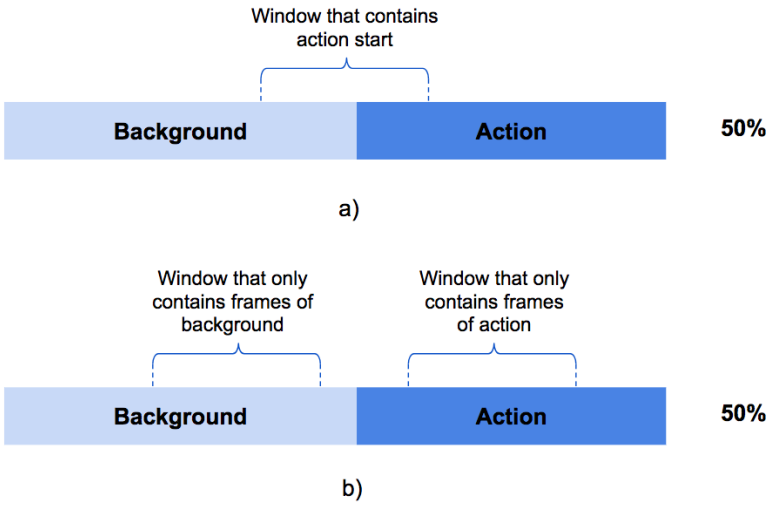


Figure 5.1: This figure shows how we sample data for the training mini batch. 50% of data contains action start while the other 50% does not contain action start.

5.2.1 Cross-entropy loss

The loss function of classification is the categorical cross entropy loss 5.5. Given 16 frames input window as input, the output is a vector of $K+1$ dimension, that represents the probability distribution over all K action classes and background.

5.2.2 L2 similarity loss

Background contents prior to action start could be diverse and thus disturb training good start detector. But contents after action start are more consistent and easier to classify, in this way we propose to minimize the feature distance 5.2 between current window around start boundary and

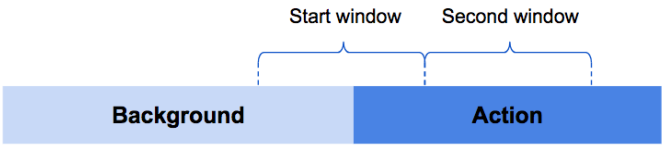


Figure 5.2: The start window contains the action start and the second window is the adjacent window to the first one. Since the start window contains both frames from the background and frames from the action, its feature representation is more diverse than the window that only contains action frames.

its next window (second window) to make it easier to classify the window around action start. During the first training stage, the two losses are jointly optimized.

$$L_{crossentropy} = -\sum_{i=1}^{K+1} L_i \cdot log(P_i) \tag{5.1}$$

$$L_{2\ similarity} = \left\| \psi(\phi(x_{start})) - \psi(\phi(x_{second})) \right\|^2 \tag{5.2}$$

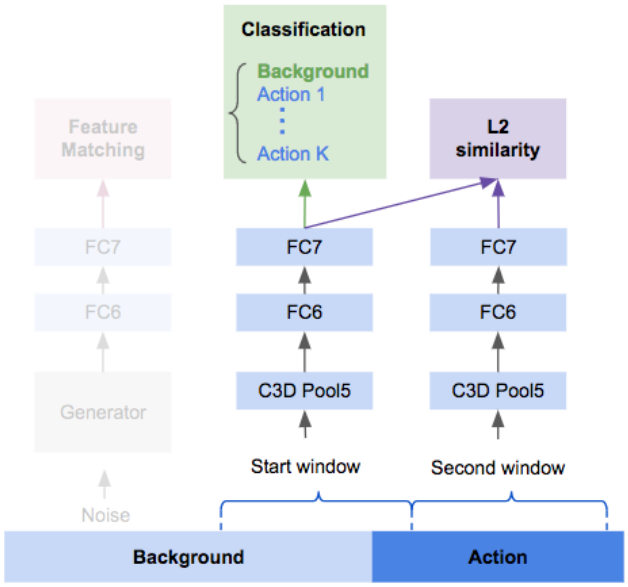


Figure 5.3: The model is first trained with classification loss and similarity loss.

5.2.3 GAN loss

Although L2 similarity loss makes windows around start boundaries more compact in the feature space, some negative samples become even more challenging to be separated. In order to solve this problem, we can use hard negative mining approach, however, the dataset is too large to identify hard samples during each training batch. Inspired by [20], we modify the C3D

model and investigate GAN models to automatically generated hard samples so that can obtain better classification boundaries. The goal of the adversarial module is to generate hard negative samples which are similar to the current window but are still separable from the current window, so that can serve as hard negative samples.

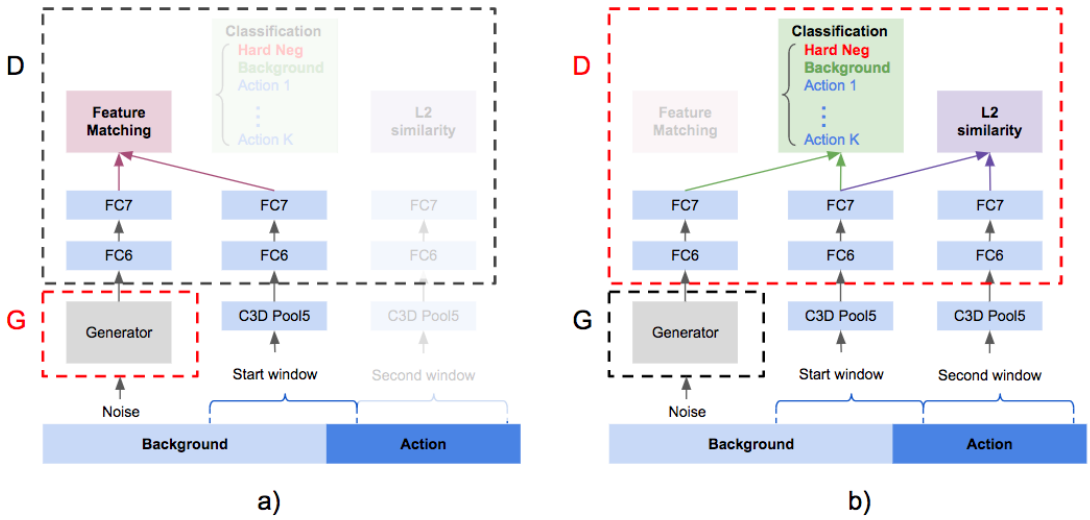


Figure 5.4: We trained GAN by optimizing G and D alternatively. a) The generator G weights are updated while the discriminator D is fixed. b) The weights of discriminator are updated the while the generator is fixed.

Training proceeds alternating between training the generator minimizing the distance between generated sample's feature and real sample's feature while keeping the discriminator fixed and training discriminator to classify the input data into one of $K+2$ possible class (the $K+2$th class corresponds to the fake class) by fixing the parameters of the generator.

$$L_G = \left\| \mathbb{E}_{x_{start} \sim p_{data}}[\psi(\phi(x_{start}))] - \mathbb{E}_{z \sim p_{noise}}[\psi(\phi(x_{second}))] \right\|^2 \tag{5.3}$$

$$\begin{aligned} L_D = -\{ & \mathbb{E}_{x_{start} \sim p_{data}}[\log\ p_D(y_c = \hat{y_c} | \phi(x_{start}))] \\ & + \mathbb{E}_{z \sim noise}[\log\ p_D(y_z = HardNeg | G(z))] \} \end{aligned} \tag{5.4}$$

$$L_{GAN} = L_D + L_G \tag{5.5}$$

5.3 Prediction and post-processing

Given 16 frames sequence, the model predicts the class category of the last frame of the sequence (current frame). The predicted class category corresponds to the class that has the maximum confidence score beyond the established threshold. If the maximum class confidence score is below the threshold the frame will be assigned as background. During testing, we slide

the input window with stride size 1. The model predicts a class label per every frame. The action start is detected, if there is a change between two adjacent windows: (1) from background to action A; (2) from action A to action B. In the first case, the model detects action A start and in the second case, the model detects action B start.

Chapter 6

Experiments

In this section, we provide details about implementation, evaluation, and validation of our proposed model.

6.1 Experiment setup

We have conducted our experiments on the THUMOS'14 [9] dataset. 2,755 trimmed videos of the training set and 1,010 untrimmed videos of the validation set (3,007 action instances) were used for training. 213 videos (3,358 action instances) of the testing set were used for testing.

The weights in all layer of the prediction network are initialized using the weights of C3D model trained on Sports-1M dataset [12], except the last fully connected layer that is initialized from a normal Gaussian distribution. The network was trained with mini batch gradient descent with Adam optimizer with a two different learning rate, $10e-5$ for the pretrained weights and $10e-4$ for the last layer. The generator network of the GAN module was initialized from normal Gaussian distribution, the learning rate used to update generator network's parameter is $3e-3$. First, we have trained the prediction network for 10.000 iterations with a mini batch of size 12, and then we add the GAN module to train 10.000 iterations more.

6.2 Evaluation Metric

In this work, we proposed a new evaluation metric to evaluates the model performance in the online action detection task. There are previous works used per frame label mean Average Precision (mAP), which is widely used in offline action detection. Nevertheless, in the online setting model only needs to detect the start of the event, the objective is not to predict labels for every frame. Therefore, we propose Point mean Average Precision ($PmAP$), which only takes into account action start frame. For each action class, we take all detected start time in the test set by their confidence scores for the specific class and compute Average Precision (AP), Then we average over all classes to obtain mAP. The correct detection must fulfill two requirements: (1) the prediction error is smaller than error tolerance threshold and (2) action category is correct.

The mean average precision of the system is obtained by computing the mean of the average precisions (AP) of the $K+1$ classes, as presented in Equation 6.1.

$$mAP = \frac{1}{|K+1|} AP(k) \tag{6.1}$$

At the same time, the average precision for a class is computed by averaging the precision at position i ($P(i)$) of all detections in the ranked list (sorted by their class confidence score, high to low). $P(i)$ is calculated as $P(i) = TP(i)/(TP(i) + FP(i))$, where $TP(i)$ and $FP(i)$ are defined as the number of true positive and false positive between positions 1 and i in the ranked list. Equation 6.2 formulates this metric.

$$AP = \frac{1}{|P|} \sum_{i=1}^{N} P(i) \cdot I(i) \tag{6.2}$$

$I(i)$ an indicator function that is equal to 1 if frame i is a true positive, and equal to 0 otherwise. P is the total number of positives.

6.3 Results

This chapter presents the results obtained with the framework presented in Chapter 5. We have evaluated our model using our newly proposed metrics, from 1-second error tolerance to 10-second error tolerance. 1-second error tolerance means that detections whose time error is lower than 1 second would be treated as correct detection. We considered a confidence score threshold of 0.6 for all our experiments. .

	Random Guess	C3D-Online	C3D-Online-DS	C3D-Online-DS-L2	**our model**
mAP (%)	0.3	3	4.7	5.3	**5.8**

Table 6.1: Average $PmAP$ from time error tolerance=1s to time error tolerance=10s. DS means dense sampling.

In Table 6.1, our model clearly outperform all baseline methods. However, even the relative gain is high, the absolute $PmAP$ value is still low. To further analyze this problem, we checked the per class AP as shown in Figure 6.1. We observed those action class with lower $PmAP$, such as Pole Vault, Javelin Throw, High Jump and Long Jump, they share very similar scene and motion information at the very beginning of the action. All videos from those classes are recorded in the athletic stadium and all actions start with running. This similarity in terms of context and motion between those action classes makes them very challenging to distinguish just from the beginning of the action. In order to address this problem, we will conduct some ablation study on using different depth/structure/model to extract more discriminative class features in future works.

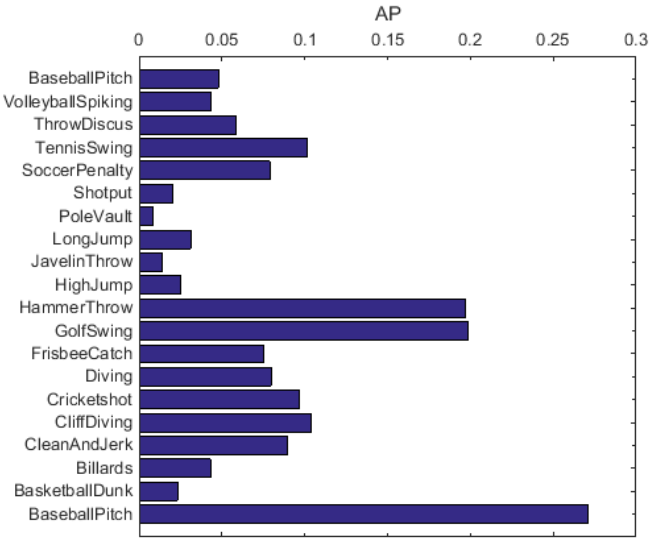
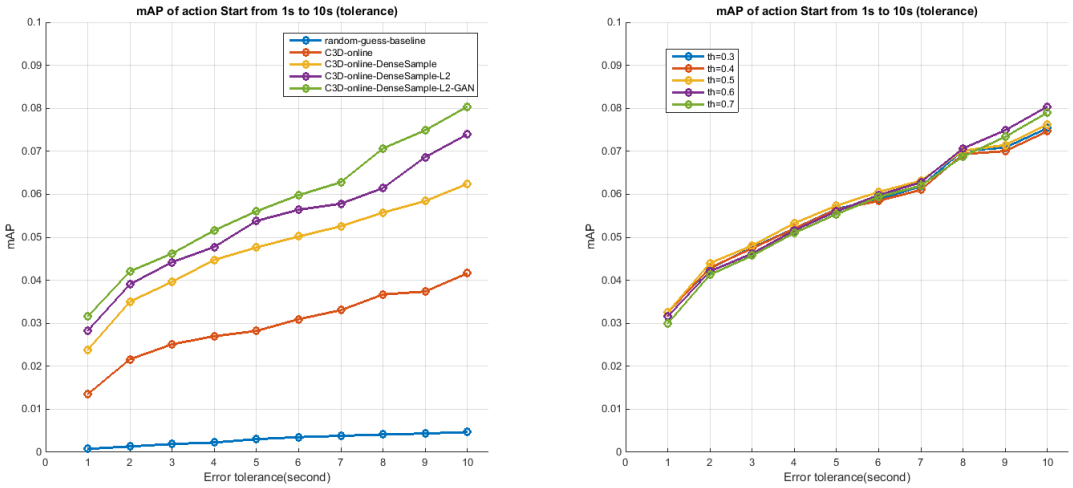


Figure 6.1: Per class Average Precision (AP) when time error tolerance is 10s.



(a) (b)

Figure 6.2: $PmAP$ of action start detection from 1s to 10s (time error tolerance). a)Presents a comparison of different models; b) Show how the confidence score threshold affects the model performance.

Chapter 7

Conclusions

The goal of this thesis is to develop an effective action detection model as well as a suitable evaluation tool in the online setting, where the input is a video stream.

Online action detection is a very difficult task, and there is no work specifically addressed on this problem, except [2] has proposed some very basic baseline method. In order to overcome those challenges in online action detection, we have conducted several experiments on a large scale dataset. We have also proved that densely sample training data around action start help train better online detection model. Reducing the distance in feature space between the start window and the its future adjacent window helps the model to learn a more compact representation. Last but not least, we have proposed GAN-based model to generate hard negative samples and improve classification accuracy.

We have introduced a novel evaluation metric, different from the frame level mAP, this newly proposed metric only focus on the starting frame, which makes it more appropriate for online action detection. We have evaluated our methods on a large scale video dataset, and our proposed models have shown very competitive performance over the baseline methods, we could obtain a 50% of the relative gain.

Finally, as the future work, it will be interesting to use other state-of-art video feature extraction network [24, 31]. Another possible modification is to generate class specific hard negative samples by providing class information as input to the generator network. We will also conduct experiments on other video dataset and work on additional benchmarks.

Bibliography

[1] Mexaction2. `http://mexculture.cnam.fr/xwiki/bin/view/Datasets/Mex+action+dataset`.

[2] Roeland De Geest, Efstratios Gavves, Amir Ghodrati, Zhenyang Li, Cees Snoek, and Tinne Tuytelaars. Online action detection. In *European Conference on Computer Vision*, pages 269–284. Springer, 2016.

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[4] Bernard Ghanem Fabian Caba Heilbron, Victor Escorcia and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–970, 2015.

[5] Adrien Gaidon, Zaid Harchaoui, and Cordelia Schmid. Actom sequence models for efficient action detection. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3201–3208. IEEE, 2011.

[6] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[7] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[8] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[9] Haroon Idrees, Amir R Zamir, Yu-Gang Jiang, Alex Gorban, Ivan Laptev, Rahul Sukthankar, and Mubarak Shah. The thumos challenge on action recognition for videos "in the wild". *Computer Vision and Image Understanding*, 155:1–23, 2017.

[10] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint arXiv:1611.07004*, 2016.

[11] Andrej Karpathy. Convolutional neural networks for visual recognition. In *Stanford CS class CS231n*.

[12] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014.

[13] Qiuhong Ke, Mohammed Bennamoun, Senjian An, Farid Boussaid, and Ferdous Sohel. Human interaction prediction using deep temporal features. In *European Conference on Computer Vision*, pages 403–414. Springer, 2016.

[14] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[15] Pauline Luc, Camille Couprie, Soumith Chintala, and Jakob Verbeek. Semantic segmentation using adversarial networks. *arXiv preprint arXiv:1611.08408*, 2016.

[16] Alberto Montes, Amaia Salvador, Santiago Pascual, and Xavier Giro-i Nieto. Temporal activity detection in untrimmed videos with recurrent neural networks. In *1st NIPS Workshop on Large Scale Computer Vision Systems*, December 2016.

[17] Junting Pan, Cristian Canton, Kevin McGuinness, Noel E. O'Connor, Jordi Torres, Elisa Sayrol, and Xavier and Giro-i Nieto. Salgan: Visual saliency prediction with generative adversarial networks. In *arXiv*, January 2017.

[18] Junting Pan, Elisa Sayrol, Xavier Giro-i Nieto, Kevin McGuinness, and Noel E O'Connor. Shallow and deep convolutional networks for saliency prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 598–606, 2016.

[19] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[20] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pages 2234–2242, 2016.

[21] Zheng Shou, Jonathan Chan, Alireza Zareian, Kazuyuki Miyazawa, and Shih-Fu Chang. Cdc: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. *arXiv preprint arXiv:1703.01515*, 2017.

[22] Zheng Shou, Dongang Wang, and Shih-Fu Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[23] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *arXiv preprint arXiv:1409.1556*, 2014.

[24] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014.

[25] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015.

[26] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2015.

[27] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Anticipating visual representations from unlabeled video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 98–106, 2016.

[28] Serena Yeung, Olga Russakovsky, Greg Mori, and Li Fei-Fei. End-to-end learning of action detection from frame glimpses in videos. *arXiv preprint arXiv:1511.06984*, 2015.

[29] Hanwang Zhang, Zawlin Kyaw, Shih-Fu Chang, and Tat-Seng Chua. Visual translation embedding network for visual relation detection. *arXiv preprint arXiv:1702.08319*, 2017.

[30] Hanwang Zhang, Zheng-Jun Zha, Yang Yang, Shuicheng Yan, Yue Gao, and Tat-Seng Chua. Attribute-augmented semantic hierarchy: towards bridging semantic gap and intention gap in image retrieval. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 33–42. ACM, 2013.

[31] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Dahua Lin, and Xiaoou Tang. Temporal action detection with structured segment networks. *arXiv preprint arXiv:1704.06228*, 2017.