CS231n Project Milestone
Video Understanding: From Video Classification to Video Captioning

Jiajun Sun
Stanford University
jiajuns@stanford.edu

Jing Wang
Stanford University
jingw2@stanford.edu

Ting-Chun Yeh
Stanford University
chun618@stanford.edu

1. Introduction

Deep video understanding application has now been stimulated by the availability of large amount annotated videos. The ultimate goal of our project is to generate captions for each video. Immediately, Vision part of our network will be tested on video classification. Microsoft multimedia challenge datasets, as dataset used in this project, provides short-clip (around 10-15 seconds) video classification, rather than long videos. Therefore, our focuses will be on captioning and classifying short video-clips.

In the sense of video classification, there are two major state-of-the-art approach: temporal feature pooling or Long Short Term Memory. Both of them has been justified as valid options to integrate temporal information over video frames. In our project, both of them will be implemented. The winner of them will be selected to be part of our video captioning model.

2. Problem Statement

At this time, we are tackling big challenges: how to select frames and how to fuse frames of a video over time periods. It is difficult to decide how many frames needed and which frames are representative enough. In general, for each video, a frame per second was extracted for convenience, which, however, loses motion information. To fuse frames of videos, both spatial and temporal information need to maintain to make more accurate predictions.

2.1. Review

As image classification becomes a solved problem. People start to show interest on video classification. The simplest approach is image-based video classification, however, it performs poorly. Different from image-based classification, many people work on end-to-end CNN model and try to learn temporal-spatial structure. 3D CNN[4] is one of the straight forward ideas. It can capture the temporal-spatial structure, but it's time consuming and is hard to train. To capture temporal-spatial structure. Simonyan and Zisserman proposed a two-stream approach[5] which breaks down the learning of video into learning of spatial and learning of temporal. Further, many improvements on two stream model was proposed. Recently, the combination of CNN and LSTM[3] becomes popular to learn videos over long time periods.

2.2. Datasets

Video description dataset is provided by Microsoft Multimedia Challenge. This dataset is based on MSR-VTT(A Large Video Description Dataset for Bridging Video and Language) and it has been split into training, testing and validation set according to 65%:30%:5% split ratio[2].

This dataset in total has 10,000 video clips with a total duration of 41.2 hours. The video clips come from 20 different categories. Annotated description has in total of 200,000 sentences[2]. Labels for this dataset, including clip category and annotated description, are generated by AMT workers. Each clip comes with 20 natural sentences by 1327 AMT workers[2].

But not all videos are available at this time. Only available videos were downloaded and processed in our project. Additionally, as you can see the distribution of video categories, we have extremely unbalanced data. To address this, the video categories with low counts have been removed first in order to get more balanced data. In the end, 6100 videos are in processed data, of which 4270 are to train and the rest are to test. In terms of the number of frames, we first extracted 1 frame per second per video and totally 10 frames per video.

2.3. Expected Result and Evaluation

Video classification is expected to have over 60% F1 score. In the sense of video captioning, We will use METEOR as a metric to evaluate our video captioning result. METEOR is a machine translation scoring metric by align-

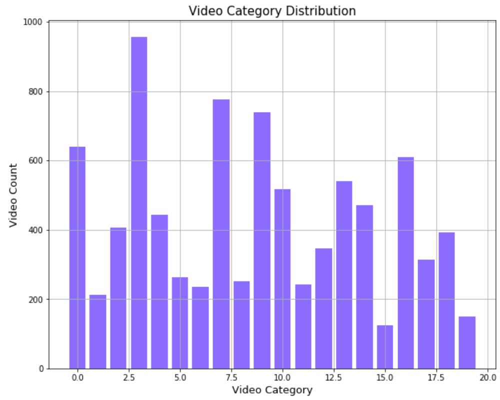


Figure 1: counts of video clips for each category

ing generated description with one or more reference description. We will compare our result with state of the art result. CMU & RUC has achieved a score of 0.282 on the same dataset in 2016. We will print out some of the results and manually compare them with natural description to identify where might the error come from. The ultimate high level result of this project is to have human readable video description; and this automatic generated description should generally convey similar to the ground truth annotated description.

3. Technical Approach

To A naive approach is to classify each frame and average the scores of classes. But actually it doesn't work very well. There are major two types of state-of-the-art approaches: temporal feature pooling[1] and LSTM[3]. In terms of video classification, both of two methods will be implemented and compare against each other.

Video captioning model, on the other hand, will have a encoder and decoder structures. The model, which is expected to outperforms in the video classification task, will be serve as the encoder for video captioning.

3.1. Single frame classification

The most direct way of video classification is to regard video as a single frame. This single frame represents the whole video. Although this method is easy to implement, it produces naive results. Not surprisingly, it is not enough to capture the context of the video by using a single frame.

3.2. 3D CNN

As described before, 3D CNN operates on stacked video frames. It extends the original 2D conv kernel and 2D pool kernel into 3D kernel to capture both spatial and temporal space which seems to make sense. C3D model proposed by Du Tran *et al* [4]. achieves state-of-the-art performance on video classification problem. However, training a 3D CNN is very time consuming and the spatial-temporal structure in videos may be too complex to capture.

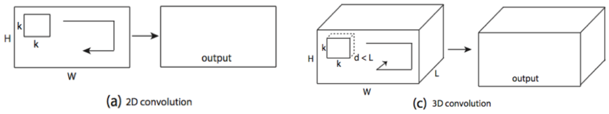


Figure 2: 3D convolution

3.3. Pretrained-based Temporal Feature Pooling

Temporal Features Pooling is introduced from bag-of-words representation application[3]. It is a layer acting on a concatenated array of video frames. Those frames are the outputs activations from CNN network. Yue-Hei Ng *et al* proposed five temporal feature pooling architecture: Conv Pooling, Late Pooling, Slow Pooling, Local Pooling and Time-Domain Pooling[3]. It turns out Conv Pooling with max pooling has the best performance, which is shown in figure 3. In details, for example, CNN architecture gives outputs for each frame for a video. Assume we have $x \in \mathbb{R}^{H \times W \times T \times D}$, which generated by concatenating outputs from pretrained CNN model. Then, apply max-pooling to x within a 3D pooling cube of size $H' \times W' \times T'$, which is a straightforward extension of 2D pooling to the temporal. It is noted that no pooling across different channels[1].

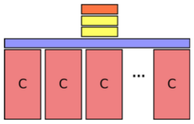


Figure 3: Temporal Pooling Layer[3]
Blue: Temporal max pooling, Yellow: Fully-connected, Orange: Softmax

Currently, we used VGG16 as our pretrained CNN model and apply $3 \times 2 \times 2$ temporal max pooling with stride 3, 2, and 2, respectively. The corresponding data dimension is (Time (T), Height (H), Width (W), Channels (D)). The entire architecture is shown in the table 1.

3.4. LSTM

Another typical approach for video classification is to use LSTM cell which connected to the output of a CNN. Like feature-pooling, LSTM networks operate on frame-level CNN activations, and can learn how to integrate information over time[3]. For each input activation frame LSTM

Table 1: CNN-based temporal max pooling architecture

Layer	Output
VGG16 without 3 fully-connected	$10 \times 7 \times 7 \times 512$
Temporal max pooling (padding = 'VALID')	$3 \times 3 \times 3 \times 512$
Fully-connected layer	1×4096
Fully-connected layer	$1 \times$ number of classes
Softmax	Scores

will output a hidden vector. In order to aggregate results from LSTM, we will linearly weighting the predictions over time sum them up.

4. Preliminary Results

First, we tried temporal max pooling architectures. Training hyperparameters are exhibited in the table 2. Finally, the accuracy and loss curves are shown in figure 4 and 5. We also output multi-class precision-recall matrix in figure 6.

During training process, the best validation accuracy is 55.0% and the test accuracy is 52.6%. From the training and loss curves, we could find both training and validation accuracies and losses finally converge. These exists a little big gap between training and validation, thus we encountered over-fitting problems. However, here we could not set too high regularizations, because it will lead to dominant regularization loss. Dropout and adjusted regularization are further considered.

In precision-recall report (Figure 6), it is easy to find the false positives and false negatives are almost balanced. However, current average F1 is 51%, which is lower than our expectation (60%). In addition, the model failed to predict the $news/events/politics$ categories. One of the reasons is that many videos in $news/events/politics$ can also be classified into other categories. It is still hard for human to differentiate the three categories.

Next step is to apply LSTM on outputs from pretrained model, which is expected to outperform the current accuracy. Since LSTM are not constraint by a fixed number of frames, we can try to increase the number of frame and see whether this help increase classification accuracy. We are also going to set up video captioning architecture based on video classification architecture.

Table 2: VGG16-based temporal max pooling architecture parameters

Parameters	Value
learning rate	1e-5
Optimizer	SGD + Momentum (nesterov)
Moment	0.9
Decay	1e-6
Batch size	32
Epochs	50
Regularization	1e-4
Validation split ratio	0.2

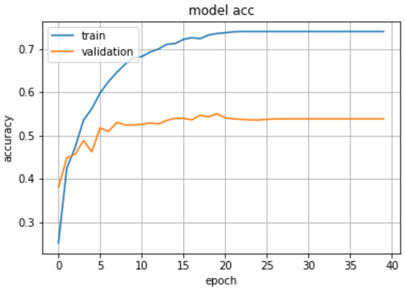


Figure 4: Model Accuracy vs Epochs

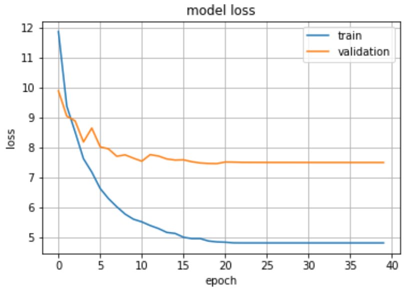
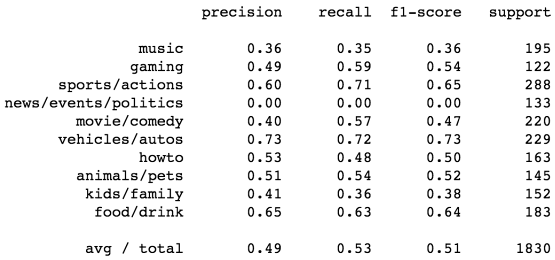


Figure 5: Model Loss vs Epochs



```
                     precision    recall  f1-score   support

              music       0.36      0.35      0.36       195
             gaming       0.49      0.59      0.54       122
     sports/actions       0.60      0.71      0.65       288
news/events/politics       0.00      0.00      0.00       133
       movie/comedy       0.40      0.57      0.47       220
    vehicles/autos       0.73      0.72      0.73       229
              howto       0.53      0.48      0.50       163
       animals/pets       0.51      0.54      0.52       145
        kids/family       0.41      0.36      0.38       152
         food/drink       0.65      0.63      0.64       183

        avg / total       0.49      0.53      0.51      1830
```

Figure 6: Precision-recall report

References

[1] Feichtenhofer, C., Pinz, A., & Zisserman, A. (2016) *Msr-vtt: A large video description dataset for bridging video and language.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 19331941).

[2] Xu, J., Mei, T., Yao, T., & Rui, Y. (2016) *Convolutional twostream network fusion for video action recognition.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 5288-5296).

[3] Yue-Hei Ng, J., Hausknecht, M., Vijayanarasimhan, S., Vinyals, O., Monga, R., & Toderici, G. (2015). *Beyond short snippets: Deep networks for video classification.* In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 46944702).

[4] Tran, D., Bourdev, L., Fergus, R., Torresani, L., & Paluri, M. (2015). *Learning spatiotemporal features with 3d convolutional networks.* In Proceedings of the IEEE International Conference on Computer Vision (pp. 4489-4497).

[5] Simonyan, K., & Zisserman, A. (2014). *Two-stream convolutional networks for action recognition in videos.* In Advances in neural information processing systems (pp. 568-576).