Large Scale Multimedia Analysis
Homework 2 Report

Ji In Kwak
Institute of Software Research
Carneige Mellon University
jiink@andrew.cmu.edu

1 Introduction

The goal of the homework 2 is to perform multimedia event detection(MED) pipeline with features from videos. Each video data consists of frame images, so that we can extract the features from the image. Otherwise, in the final step, we can extract the features of videos with the 3-dimensional CNN models. We will explain the implementations of SIFT features, CNN features, and 3D-CNN features, respectively, in the following sections.

2 Task 1: SIFT features

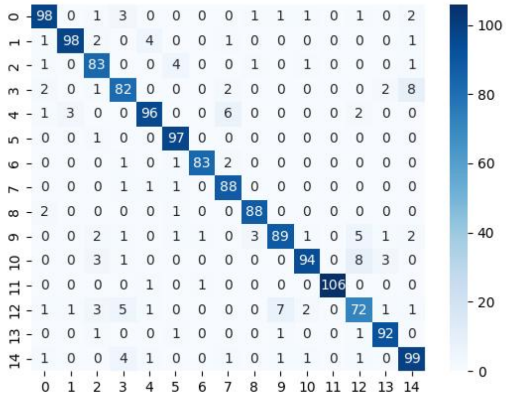
The first task for processing video frames is to extract the SIFT(Scale-invariant feature transform) features from each frame images. After extracting SIFT features from each video, we can construct the bag of words features which can be used in MLP classifier.

First step is resampling the video frames and extracting the SIFT features. Since the video is the sequence of frame images, successive frames are likely to have similar images. Hence, we downsampled the frames into target frame rate 1.5Hz. In order to do this, we calculated the downsampling ratio as $\frac{Target\ frame\ number}{Current\ frame\ number}$ and selected video frames at intervals by the inverse of this ratio. For example, if the target frame rate is 1.5Hz and current frame rate is 30Hz, we can choose the selected frames at 20 intervals. Then, for each frame image, we can extract 32 SIFT features with feature dimension 128.

Second step is to train the K-means clustering with the extracted SIFT features. Since there are too many number of extracted SIFT features, we chosen the subset of features. I chose about half of the features with is 15 for each frame image. The number of clusters are set as 150 to distinguish with the feature dimension 128. If the number of clusters become too small, it cannot verify the differences between every SIFT features. Otherwise, if it is too large, then it might not capture the correlations between similar features. Hence, it is important to set the appropriate number of clusters and train the K-means clustering.

The last step is to compute a bag-of-words feature vectors. After training K-means clustering, 150 cluster centroids are saved in 'kmeans/sift_150.pkl' file. From each video frame image, we find the nearest cluster for every SIFT features. Then, count the number of clusters that is selected as nearest clusters. The 150 dimension counts vector becomes the bag-of-words feature of the frame. Since we are dealing with the video data, use mean pooling methods to average the information from all frames. The extracted features are trained using MLP classifiers which the skeleton code is given. I used MLP with one hidden layer and put batch-normalization layer with larger batch size 1024. The best public score in Kaggle was 0.3235 which is no less than 0.02 below the sift baseline.

Preprint. Under review.



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	98	0	1	3	0	0	0	0	1	1	1	0	1	0	2
1	1	98	2	0	4	0	0	1	0	0	0	0	0	0	1
2	1	0	83	0	0	4	0	0	1	0	1	0	0	0	1
3	2	0	1	82	0	0	0	2	0	0	0	0	0	2	8
4	1	3	0	0	96	0	0	6	0	0	0	0	2	0	0
5	0	0	1	0	0	97	0	0	0	0	0	0	0	0	0
6	0	0	0	1	0	1	83	2	0	0	0	0	0	0	0
7	0	0	0	1	1	1	0	88	0	0	0	0	0	0	0
8	2	0	0	0	0	1	0	0	88	0	0	0	0	0	0
9	0	0	2	1	0	1	1	0	3	89	1	0	5	1	2
10	0	0	3	1	0	0	0	0	0	0	94	0	8	3	0
11	0	0	0	0	1	0	1	0	0	0	0	106	0	0	0
12	1	1	3	5	1	0	0	0	0	7	2	0	72	1	1
13	0	0	1	0	0	1	0	0	0	1	0	0	1	92	0
14	1	0	0	4	1	0	0	1	0	1	1	0	1	0	99

Figure 1: The confusion matrix of MLP model trained with CNN features

3 Task 2: CNN features

In second task, we extracted the video features from pre-trained CNN model. Resnet18 [1] is one of the most popular image training model trained with ImageNet dataset. Resnet18 consists of 18 layers and for each frame, the feature is generated from the last average pooling layer.

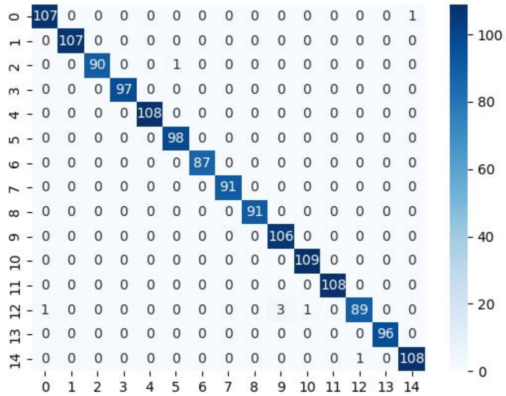
After extracting the video features from Resnet18 with dimension 512, we can train the MLP classifier. There are 1 hidden layer with dimension 512 and the validation accuracy was 0.91. The prediction accuracy for public test set is 0.8877. For 15 classes, we can construct the confusion matrix to represent the classification results. The image of confusion matrix is shown in Figure 1.

4 Task 3: 3D-CNN features

The last task is to use the 3D CNN model for the video classification. It is different with normal CNN model. CNN model extract the features from every frames but 3D CNN model deal with every video as an 3 dimensional input $(Time \times Height \times Width)$.

We can use the pretrained 3D CNN model from torchvision which includes R3D_18, R2PLUS1D_18, and MC3_18 [2]. I compared the results of R3D_18 and R2PLUS1D_18. The average pooling layer is used for the feature extractor and the dimension of features are both 512. In task 3, I trained the MLP classifier by monitoring the minimum validation loss, not the maximum validation accuracy. That is because when we train the model, the validation accuracy is not updated while the loss value is still decreasing, so that we cannot save the model with the lowest loss value. By monitoring the validation loss, the saved best model might perform better in the public and private test set.

For R(2+1)D network, we used 1 hidden layer with dimension 1024 and the validation accuracy was 0.9947. The prediction accuracy for public test set is 0.9652. Otherwise, for Resnet 3D model, the validation accuracy was 0.982 and the public test set accuracy was 0.95187 which is slightly lower than the previous. I constructed the confusion matrix for the R(2+1)D network which is shown in Figure 2.



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	107	0	0	0	0	0	0	0	0	0	0	0	0	0	1
1	0	107	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	90	0	0	1	0	0	0	0	0	0	0	0	0
3	0	0	0	97	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	108	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	98	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	87	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	91	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	91	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	106	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	109	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	108	0	0	0
12	1	0	0	0	0	0	0	0	0	3	1	0	89	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	96	0
14	0	0	0	0	0	0	0	0	0	0	0	0	1	0	108

Figure 2: The confusion matrix of MLP model trained with 3D CNN features from R(2+1)D

References

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[2] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018.