BBM 418 - Computer Vision Laboratory

HACETTEPE UNIVERSITY

Department of Computer Engineering

Programming Assignment-4
Spring 2019-2020
Dr. Nazli Ikizler Cinbis

Due Date: 23:59 pm on Friday, June 12th, 2020

Two-stream CNNs to Recognize Actions in Videos

In this assignment, you will implement and analyze a two-stream video action recognition model. The goals of this assignment are follows;

- Understand two stream models
- Understand video action recognition models.
- Understand optical flow features and their effects.
- Gain experience with a major deep learning framework, PyTorch.

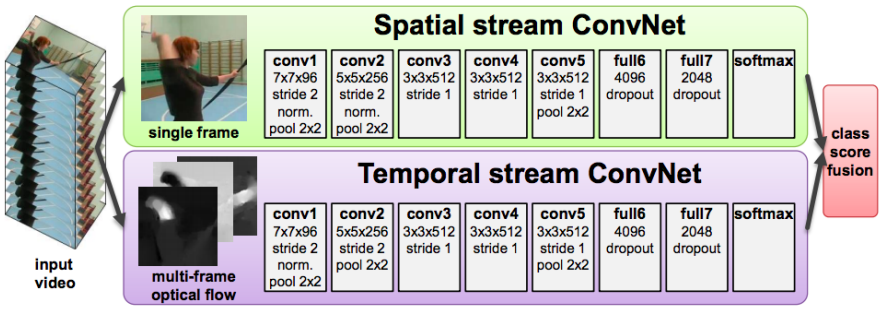

Figure 1: A basic two-stream action recognition model, taken from [1].

Introduction

Recognition of human gestures or actions in videos is a challenging task due to the temporal content of videos which includes an additional information. Therefore we aim to recognize actions from a given video. Compared to still image classification, video action classification is more challenging therefore the field attracts more attention.

In this assignment, we are going to recognize American Sign Language signs present videos. We are going to extend CNNs to classify sign language videos. In order to do this, we are going to both utilize spatial and temporal streams. Note that, it is crucial to learn effective spatio-temporal features on video based applications.

Figure 2: An example from MS-ASL dataset.

Dataset

You will utilize MS-ASL dataset[2]. The dataset includes 4 different subsets; ASL100, ASL200, ASL500, and ASL1000. Each subset includes their own train, validation, and test sets. You will use a very small subset with only 10 classes which has label between 0 to 10. The subset includes the following class names and labels;

1. label = 0, clean_text = hello
2. label = 1, clean_text= nice
3. label = 2, clean_text = teacher
4. label = 3, clean_text = eat
5. label = 4, clean_text = no
6. label = 5, clean_text = happy
7. label = 6, clean_text = like
8. label = 7, clean_text = orange
9. label = 8, clean_text = want
10. label = 9, clean_text = deaf

Label ids between 0 and 10 represents your assignment subset. You will download class samples from the given json files. You may reach the json files from; https://www.microsoft.com/en-us/download/details.aspx?id=100121.

The subset includes at least 45 samples per class. Therefore for each class in your subset you will only download the samples that is available. Please do not forget to align videos with respect to start and end time.

Recognition model

You are expected to implement a two-stream Convolutional Networks for action recognition using videos. (See Figure 1). Our model will be a slight variation of the model proposed from [1] (See Figure 1). You will train the model by both using optical flow frames and video frames.

You will implement each stream with a CNN and softmax scores of which are combined by fusion. The architecture has two networks; spatial net and motion net. SpatialNet operates on RGB frames and TemporalNet operates on optical flow input. Each stream (individual network) will be implemented by ConvNets. The input to the SpatialNet will be a single frame of the video. The input to the temporal stream will be a stacking optical flow displacement fields between 10 consecutive frames. Here are some example codebases that you can use : In OpenCV: DIS optical flow algorithm: `https://docs.opencv.org/3.4/da/d06/classcv_1_1optflow_1_1DISOpticalFlow.html#details`, NVIDIA's PWCNet: `https://github.com/NVlabs/PWC-Net`. You can also use other optical flow extraction libraries.

You are expected to implement three experiments:

1. **Train separate CNNs.** SpatialNet and TemporalNet will be trained and evaluated separately. They will have separate loss functions to optimize. [40 pts]
2. **Late fusion.** Two networks are fused together at the output, by combining the softmax scores of the networks. Loss is defined as the combined loss of the two networks, which is by adding the two losses, and backpropagating over the combined losss. [30 pts]
3. **Early fusion.** Two networks are fused together before the last FC layer. The feature vectors of the two networks are concatenated and a FC layer is learned on top of this concatenated vector. There will be only one loss function here. [30 pts]

Note that the base network architectures should be the same across these experiments.
For each of the above experiments:

1. Model your network architecture and describe it in detail.
2. Tune your parameters accordingly and give accuracy on validation set for all setups. Report and examine each experiment that you conduct in detail.
3. Evaluate your best trained model in the test set and plot the confusion matrix. What do the results tell us? Explain and discuss in detail.

The Implementation Details

1. You should pay attention to code readability such as comments, function/variable names and your code quality: 1) no hard-coding 2) no repeated code 3) cleanly separate and organize your code 4) use consistent style, indentation
2. Implement your code with Python 2 or 3 and use the necessary Pytorch, OpenCV, NumPy libraries.
3. You should use the PyTorch as the deep learning framework. You can use Google Colab to run your experiments.

What should you write in the report?

- Give explanations for each step.
- Explain the experimental results, used parameters and comment on the results in detail.
- Give your model's loss plot and accuracy plot both for training and validation set during training.
- Put the results of different hyper-parameters (learning rate, batch size), the effect of them, with the loss/accuracy plots.
- A basic structure might be: 1) Introduction (what is the problem, how do you approach to this problem, what is the content of your report) 2) Implementation Details (the method you followed and details of your solution) 3) Experimental Results (all results for separate parts with different parameters and your comments on the results) 4) Conclusion (what are the results and what are the weaknesses of your implementation, in which parts you have failed and why, possible future solutions)
- You should write your report in LaTeX
- You should give visual results by using a table structure.

What to Hand In

The upload link is; `https://classroom.github.com/a/itVaA1iD`

Your submission format will be:

- README.txt *(give a text file containing the details about your implementation, how to run your code, the organization of your code, functions etc.)*
- code/ *(directory containing all your code)*
- report.pdf

Academic Integrity

All work on assignments must be done individually unless stated otherwise. You are encouraged to discuss with your classmates about the given assignments, but these discussions should be carried out in an abstract way. That is, discussions related to a particular solution to a specific problem (either in actual code or in the pseudocode) will not be tolerated. In short, turning in someone else's work, in whole or in part, as your own will be considered as a violation of academic integrity. Please note that the former condition also holds for the material found on the web as everything on the web has been written by someone else.

References

1. Simonyan, Karen, and Andrew Zisserman. "Two-stream convolutional networks for action recognition in videos." Advances in neural information processing systems. 2014.
2. Joze, Hamid Reza Vaezi, and Oscar Koller. "Ms-asl: A large-scale data set and benchmark for understanding american sign language." arXiv preprint arXiv:1812.01053 (2018).
3. http://blog.qure.ai/notes/deep-learning-for-videos-action-recognition-review