CSCE 636 – Deep Learning

Human Action Recognition in Video Using R2plus1d DNN Model

Tsz Yi Yang
(nickname: CochiLocoYang)
Biological and Agricultural Engineering

May 1, 2021

Abstract

This report is documented for the practice of human action recognition in videos using deep neural network. The selected actions were “cleaning floor” (submission-8) and “sharpening knives” (submission-10). In the early stage of experimenting, different computational libraries were examined, and Microsoft-computervision-recipes library was the easiest to use with minor modification. The deep neural network (DNN) used throughout the semester was R(2+1)D-34 layers architecture. The model was pretrained with 65 million videos from Instagram (ig65m) and another pretrained weight option is using ig65m combined with DeepMind-Kinetic400 (kinetics). The baseline performance of the pretrained model has test accuracies at 92% on UFC101 and 73% on HMDB51 respectively. Due to the limitation of computational power, the video clips used to train for target action detection was approximately 350 clips for “cleaning floor” and for “sharpening knives” action the training videos were intentional reduced to 150 clips. Two models have similar performance compare to baseline with the test accuracies at 96% for “cleaning floor” and 90.5% for “sharpening knives” respectively. For the actual model deployment, there were 149 “cleaning floor” videos and 629 “sharpening knives” videos collected base on video title information from Youtube. The average video length is approximately 3 minutes each and that’s roughly 450 minutes and 1900 minutes of videos for detection. The runtime to complete action detection is 24 hours and 96 hours for cleaning floor and for sharpening knives.

1. **Topic**

   My first selected human action is cleaning floor. The model training was success with test accuracy at 96%, but the model deployment relied on manual input between different videos. I also trained the models with R(2+1)D-34 layers to detect happy and angry emotions. The test accuracies were low between 56% - 63% so the result was not submitted. Sharpening knives is the target action for submission-10 and the training videos were intentionally reduced to half for experimenting. Two pretrained weight models, ig65m and ig65m+kinetics, were used to experiment the model differences. The final test accuracies for "sharpening knives" model were 89% for ig65m and 92% for ig65m+kinetics. An automated action detection script was developed and I was able to submit K=7,985 clips (submission-8, cleaning floor) and K=24,536 clips (submission-10, sharpening knives).

2. **Motivation**

   The idea to select cleaning floor action is due to the total suspended particulate (TSP) will likely increase when using vacuum/cleaning regent/wiping surface...etc. This will be a good moment to increase the air exchange ratio in the room or increase fan speed for TSP removal. This action detection model can be used to trigger the HVAC system inside home when resident cleans the floor.

   Then I was interested to explore if an emotion recognition model can be helpful to differentiate the moods from my wife. My wife is more complicated than the capable of DNN model. The idea is that woman and man have different behavioral patten in different moods. My attempting was to identify the small motion differences when the emotion occurs. However, the existing emotion dataset were rather focus on face expressions than the body movement.

   Lastly, sharpening knives was picked because my wife broke the kitchen knife while chopping pork bone. She was mad at it, but the angry emotion model didn't help me to identify her bad mood. Afterward I went to buy the sharpener and sharpened the broken knife. The potential use of this model is building smart kitchen knives rack which reminds the user for the knives sharpening schedule or provide assistance to prevent under/overly sharpen the knives.

3. Related Work

Motion recognition:

R(2+1)D network was first discussed by Tran *et al.*, 2018, which were modified from R3D architecture. Unlike R3D network, R(2+1)D structured with a parallelized R2D + R1D layers where the 2D channel handles the spatial convolution followed by 1D convolution to decompose the temporal modeling. The advantage of using R(2+1)D is that it doubles the nonlinear nodes without changing its parameters compare to R3D by introducing one more ReLu function at each 1D layer. Initially the R(2+1)D were tested for 18 layers with test accuracies similar to R3D model at 50-70%; and for 34 layers the test accuracy bump up by nearly 10% at highest 92% accuracy [1]. Another advantage of R(2+1)D is the spatial and temporal components can be tuned separately.

In most recent development on R(2+1)D model, the deep network had further expanded to 101 layer and 152 layers. The parameter number has grown from 64 millions for 34 layers to 118 millions for 152 layers respectively. This implied the deeper model could decompose more complex structure and the results shows R(2+1)D-152 has improved its accuracy by ~10% compare to R(2+1)D-34 [2]. The other noticeable benefit of using deeper R(2+1)D is the learning curve is steeper and it handles the noise better than R(2+1)D-34 model. The training epoch is lesser than fewer layer models.

Emotion recognition:

Chang *et al.* describes using deep CNN to regress facial expressions on 3D morphable model (3DMM). The related 3DMM work usually involves uses of facial landmark detection base on image intensity to generate an artificial 3D face expression before human face emotion recognition. In the proposed method, Expression-Net (ExpNet), they skipped matching facial landmarks, pose fitting, and expression fitting, instead just using the ResNet-101 architecture to assign facial expression coefficients for generating the artificial 3DMM faces. In the facial landmark-free approach, the trained deep CNN has more robust results to generate such 3DMM faces disregarding the picture quality, face pose, viewpoint of the face, and the spatial location of human face images. ExpNet was tested on CK+ and EmotiW-17 datasets and the facial regressions were generally better or matching

performance compare to other methods. Few exceptions are the facial expression of Sad in CK+ and Happy in EmotiW-17 which ExpNet is slightly underperformed to other methods. The regressing runtime for facial expression in face landmark-free method is as fast as 0.088 second per image when other approaches have runtime averaging at 0.9 second, roughly 10x faster in general. The emotion recognition accuracy for ExpNet is also outperforming from other methods. The test accuracies of ExpNet on reduced scale image sets were above 65% for CK+ dataset and at approximately 30% on EmotiW-17 dataset [3].

4. **Propose Model**

   The main DNN architecture used is R(2+1)D-34 layers model (shown in appendix 1). This proposed model is a compromise between model performance, runtime, and cost of computation. Although R(2+1)D-101 and R(2+1)D-152 have better precision to detect motions, they are also very costly to train. The original tests were trained and validated on a parallelized 16-machines with total 128-GPUs rig, which is incredibly expensive for daily user. My first attempt was training the R(2+1)D model on a Nvidia RTX 3070 (8GB ram) and the CUDA memory often bottlenecks the training process. I had to minimize the batch size and reduce number of epochs and learning rate to prevent the script from crashing. Then I swap to a Nvidia RTX 2080Ti (11GB ram) with the same training setup and it was runnable for smaller training video dataset with higher epoch and finer learning rate. R(2+1)D-18 might be good alternative architecture if a decent GPU is not available. But the compromise is another 10%-20% accuracy drop compare with R(2+1)D-34 architecture.

5. **Dataset**

   The training and validation video dataset was portion of the DeepMind-Kinetic-400. This dataset was first released with 400 human action classes and were categorized into 38 different action groups with 306 thousands of total videos [4]. The revised version of Kinetic-600 and Kinetic-700 rolled out in the next two years following the original publication [5, 6].

6. **Model Training and Performance**

   This project was developed on a single GPU (RTX 2080Ti, 11GB ram) local machine and occasionally cross examined with my friend's computer rig with a single RTX 3070 GPU (8GB

ram). The project is published on my GitHub repository. The scripts were tuned to perform the deep learning tasks without issue on RTX 2080Ti but they may appear CUDA memory outage on lower GPU configuration. The training setup is listed in table 1 for the target actions.

The first trained action was cleaning floor. In this stage, many test configurations were experimented. The main tuning was focus on the number of training epochs. The test accuracies were 78% with 10 epochs and 96% with 20 epochs. The validation accuracy and loss curve are shown in figure 1. The video clips for "false" action were downloaded from YouTube and were totally random, which says the clips are diverse and may contain huge noise. My model extracts target action features very well as it recognizes "human", "tool", "flat surface". But some flaws in the model action detection were noticed. When any combination of "human", "tool", and "flat surface" appears, the model recognized as "cleaning_floor". Some false-positive examples like "a man standing with a tool in hand", "footage of people walking on a flat surface", or "a mop sitting on the floor" will likely be identified positively.

The next trained action was sharpening knives. For this experiment, two pretrained weight were tested and the test accuracies were at 89%-92% shown in figure 2. This experiment was intentionally reducing the training dataset to simulate real world scenario where harvesting massive dataset could be challenging. The video clips for “false” action were selective to those contained white noises such as any combinations of “human” + “knives” + “sharpener” and some false motion with those elements. For instance, a man chopping food with knife, using machinery tools to grind the metal, people holding items and talking…etc. This approach reduced the false-positive in real Youtube clip detection by eye browsing during model deployment. The main challenge remaining is the model can not detect the “Not in Motion” cases. Many Youtube videos were demonstrating the content and display a still graphic. There could be moments that the speaker is verbally introducing ideas with a demonstrative photo without motion. If the picture contains target features, my model will still recognize it as positive even without movement.

Table 1. Training setup for action recognition model for two independent actions.

Target action	Cleaning floor	Sharpening knives
Positive clips	300	150
Negative clips	300	150
Training epochs	10, 20	20
Learning rate	1e-4	1e-4
Batch size	16, 8	8
Pretrained weight	Ig65m	Ig65m, ig65m+kinetics

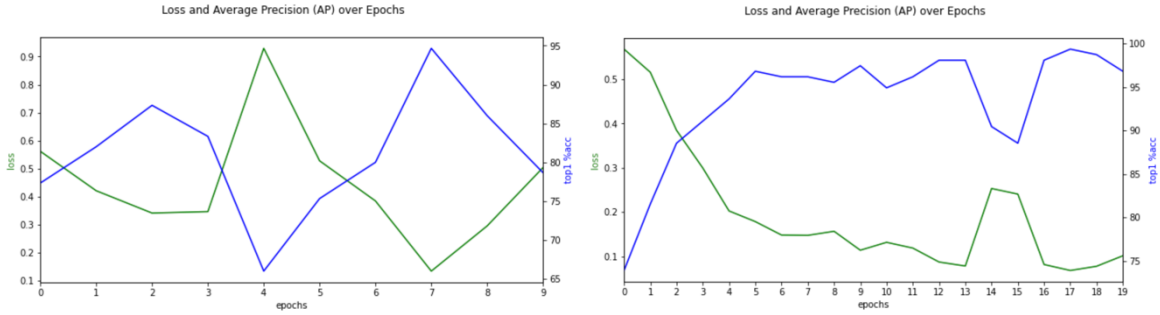

Figure 1. Validation accuracy (blue line) and loss (green line) diagrams for Left: 10 epcohs and for Right: 20 epochs model training. Other parameter is configurated the same as shown in table 1.

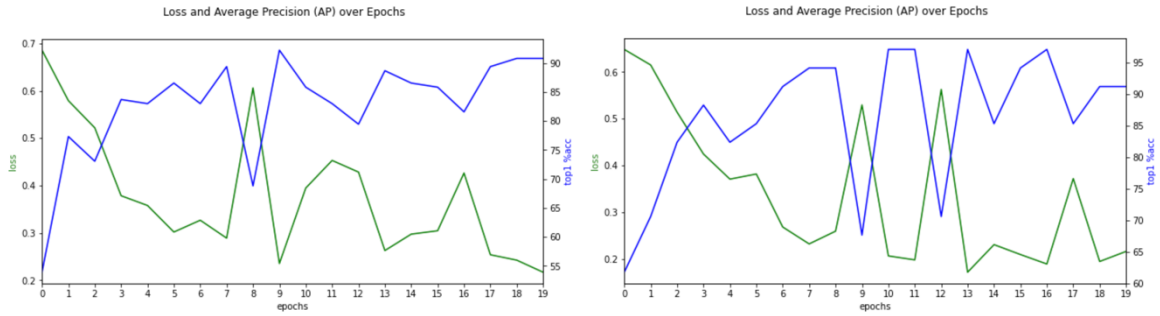

Figure 2. Validation accuracy (blue line) and loss (green line) diagrams with the same test configuration as shown in table 1. The diagram on the left is showing pretrained weight with ig65m and the diagram on the right showing pretrained weight with ig65m+kinetics.

Appendix 1. The architecture of R(2+1)D-34 layers DNN model contains nearly 64 millions of trainable parameter.

```
1.  ==========================================================================
2.  Layer (type:depth-idx)                   Param #
3.  ==========================================================================
4.  ├─R2Plus1dStem: 1-1                      --
5.  |    └─Conv3d: 2-1                       6,615
6.  |    └─BatchNorm3d: 2-2                  90
7.  |    └─ReLU: 2-3                        --
8.  |    └─Conv3d: 2-4                       8,640
9.  |    └─BatchNorm3d: 2-5                  128
10. |    └─ReLU: 2-6                        --
11. ├─Sequential: 1-2                       --
12. |    └─BasicBlock: 2-7                   --
13. |    |    └─Sequential: 3-1             111,008
14. |    |    └─Sequential: 3-2             111,008
15. |    |    └─ReLU: 3-3                   --
16. |    └─BasicBlock: 2-8                   --
17. |    |    └─Sequential: 3-4             111,008
18. |    |    └─Sequential: 3-5             111,008
19. |    |    └─ReLU: 3-6                   --
20. |    └─BasicBlock: 2-9                   --
21. |    |    └─Sequential: 3-7             111,008
22. |    |    └─Sequential: 3-8             111,008
23. |    |    └─ReLU: 3-9                   --
24. ├─Sequential: 1-3                       --
25. |    └─BasicBlock: 2-10                  --
26. |    |    └─Sequential: 3-10            221,516
27. |    |    └─Sequential: 3-11            443,200
28. |    |    └─ReLU: 3-12                  --
29. |    |    └─Sequential: 3-13            8,448
30. |    └─BasicBlock: 2-11                  --
31. |    |    └─Sequential: 3-14            443,200
32. |    |    └─Sequential: 3-15            443,200
33. |    |    └─ReLU: 3-16                  --
34. |    └─BasicBlock: 2-12                  --
35. |    |    └─Sequential: 3-17            443,200
36. |    |    └─Sequential: 3-18            443,200
37. |    |    └─ReLU: 3-19                  --
38. |    └─BasicBlock: 2-13                  --
39. |    |    └─Sequential: 3-20            443,200
40. |    |    └─Sequential: 3-21            443,200
41. |    |    └─ReLU: 3-22                  --
42. ├─Sequential: 1-4                       --
43. |    └─BasicBlock: 2-14                  --
44. |    |    └─Sequential: 3-23            884,632
45. |    |    └─Sequential: 3-24            1,771,136
46. |    |    └─ReLU: 3-25                  --
47. |    |    └─Sequential: 3-26            33,280
48. |    └─BasicBlock: 2-15                  --
49. |    |    └─Sequential: 3-27            1,771,136
50. |    |    └─Sequential: 3-28            1,771,136
51. |    |    └─ReLU: 3-29                  --
```

```
52. |    └─BasicBlock: 2-16                 --
53. |    |    └─Sequential: 3-30          1,771,136
54. |    |    └─Sequential: 3-31          1,771,136
55. |    |    └─ReLU: 3-32                --
56. |    └─BasicBlock: 2-17                 --
57. |    |    └─Sequential: 3-33          1,771,136
58. |    |    └─Sequential: 3-34          1,771,136
59. |    |    └─ReLU: 3-35                --
60. |    └─BasicBlock: 2-18                 --
61. |    |    └─Sequential: 3-36          1,771,136
62. |    |    └─Sequential: 3-37          1,771,136
63. |    |    └─ReLU: 3-38                --
64. |    └─BasicBlock: 2-19                 --
65. |    |    └─Sequential: 3-39          1,771,136
66. |    |    └─Sequential: 3-40          1,771,136
67. |    |    └─ReLU: 3-41                --
68. ├─Sequential: 1-5                      --
69. |    └─BasicBlock: 2-20                 --
70. |    |    └─Sequential: 3-42          3,539,506
71. |    |    └─Sequential: 3-43          7,081,216
72. |    |    └─ReLU: 3-44                --
73. |    |    └─Sequential: 3-45          132,096
74. |    └─BasicBlock: 2-21                 --
75. |    |    └─Sequential: 3-46          7,081,216
76. |    |    └─Sequential: 3-47          7,081,216
77. |    |    └─ReLU: 3-48                --
78. |    └─BasicBlock: 2-22                 --
79. |    |    └─Sequential: 3-49          7,081,216
80. |    |    └─Sequential: 3-50          7,081,216
81. |    |    └─ReLU: 3-51                --
82. ├─AdaptiveAvgPool3d: 1-6               --
83. ├─Linear: 1-7                      205,200
84. ==========================================================================
85. Total params: 63,697,175
86. Trainable params: 63,697,175
87. Non-trainable params: 0
88. ==========================================================================
```

Reference:

1. Tran, D., et al. *A closer look at spatiotemporal convolutions for action recognition*. in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2018.
2. Ghadiyaram, D., D. Tran, and D. Mahajan. *Large-scale weakly-supervised pre-training for video action recognition*. in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019.
3. Chang, F.-J., et al. *Expnet: Landmark-free, deep, 3d facial expressions*. in *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*. 2018. IEEE.
4. Kay, W., et al., *The Kinetics Human Action Video Dataset.* arXiv pre-print server, 2017.
5. Carreira, J., et al., *A Short Note about Kinetics-600.* arXiv pre-print server, 2018.
6. Carreira, J., et al., *A Short Note on the Kinetics-700 Human Action Dataset.* arXiv pre-print server, 2019.