Recognize Flu-like Symptoms with Deep Learning

Alan Lou
Stanford University
alanlou@stanford.edu

Yechao Zhang
Stanford University (SCPD)
yechaoz@stanford.edu

Abstract

Human action recognition has always been one of the most active research fields, especially with the recent developments of Deep Learning. In this paper, we present a variety of models to detect coughing and sneezing motions from video data. Previous work [17] exists in this subject area, but is limited and not based on deep learning approaches. Here we first explored two baseline models: CNN + LSTM model and 3D-Conv model. These two baseline models perform worse than the previous work, which attained a test accuracy of 44.4%. Then we improved the first baseline model to two more advanced models: VGG-16 Features + LSTM model and HRNet Features + LSTM network model. Finally, we investigated if adding an Attention layer after the LSTM layer helps improve the models. Our best-performing model outperforms the previous work significantly, achieving a test accuracy of 83.5%.

1. Introduction

The novel coronavirus disease 2019 (COVID-19) pandemic has reached almost every country in the world, infecting millions of people and plunging the global economy into recession as governments imposed tight restrictions to tackle the spread of the virus. The COVID-19 pandemic incurred significant financial costs to the economy. The total cost is estimated at more than $16 trillion [3].

Approaches for early detection of flu-like symptoms are expected to have a major impact in the direction of pandemics. One potential method of early detection is to detect typical flu-like symptoms, like coughing and sneezing, by automatically monitoring and analyzing video footage from densely populated areas. This will provide an invaluable source of information to detect respiratory viral infections such as COVID-19 early and therefore has great social and economical impact.

The previous work [17] by Thi et al. predicts flu-like symptoms from videos with traditional feature extraction methods such as histogram of oriented gradient (HOG) and histogram of optical flow (HOF). In this paper, we are going to improve the classification of symptoms with adoption of the recent deep learning models leveraged for computer vision.

Recognizing flu-like symptoms from videos is an action recognition problem. The inputs to our models are video frames with human performing different actions and the outputs are predicted action labels corresponding to the videos. Since the final goal of our model is to recognize flu-like symptoms like sneezing and coughing among a variety of actions, we will convert the predicted actions into a binary label of coughing and sneezing actions.

We will explore and compare 6 models to approach this problem. The first two are baseline models: CNN + LSTM model and 3D-Conv model. Next, we improve the first baseline model by replacing the CNN feature extractor with more advanced feature extractors like VGG-16 and HRNet. Finally, we investigate if adding an Attention layer after the LSTM layer helps improve the models.

2. Related Work

In this section, we describe the related work done for recognizing flu-like symptoms from video data. As mentioned earlier, there were limited attempts by others at the exact same problem, so we will also describe the works done in the broader field of action recognition.

The previous work to recognize flu-like symptoms [17], along with a number of other other works ([4], [13], [11]), assume that human actions can be sufficiently described by a set of local features in space-time and employ traditional feature extraction methods to extract such local features. These traditional methods are usually less computationally expensive than deep learning methods, but they do not take advantage of the increased performance of computer processors in recent years.

Around the same time, 3D convolutions [8] were used for the broader task of action recognition in 2013. The 3D-Conv model extracted features from both the spatial and the temporal dimensions by performing 3D convolutions and it achieved good performance in comparison to baseline methods.

Soon after this in 2014, two breakthrough research pa-

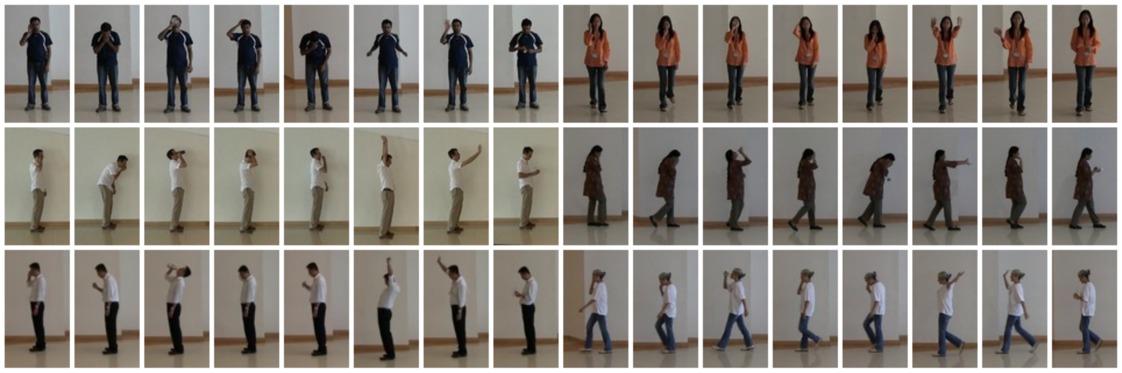

Figure 1. From top to down shows eight actions: answer phone call, cough, drink, scratch head, sneeze, stretch arm, wave hand and wipe glasses. From left to right shows six pose-and-view variations: stand-front, stand-left, stand-right, walk-front, walk-left, and walk-right.

pers - Single Stream Convolutional Network [9] by Karpathy et al. and Two Stream Convolutional Networks [14] by Simmoyan and Zisserman - were released. These deep Convolutional Networks based methods introduced many great ideas and paved way for future works.

There were significant progress made in field of video-based action recognition in the following years. The work [5] by Donahue et al. extended the previous works by using a recurrent convolutional (RNN) architecture instead of the stream based designs. Additionally, end-to-end trainable architecture was proposed for action recognition.

The paper [19] by Du Tran et al. repurposed 3D convolutional networks as feature extractors. Then the paper [23] by Yao et al. used an attention mechanism on top of 3D convolutional networks to learn spatiotemporal features.

There are also later works like [21], [6], [24] and [2] that evolved from the above-mentioned papers. All of these works employed deep learning based approaches and achieved better results compared with traditional methods.

3. Dataset and Features

In this section, we describe the dataset characteristics, the pre-processing process, and the training/validation/test split of the dataset.

3.1. Dataset characteristics

We use the BII Sneeze-Cough Human Action Video Dataset (BIISC) [18] created by Thi et al. This is a video action dataset dedicated towards the problem of flu-like symptoms detection, which is of central importance in early surveillance of respiratory disease outbreaks.

Below is a summary of the dataset statistics:

- Number of subjects: 20, M/F:12/8
- Number of action types: 8, answer phone call, cough, drink water, scratch head, sneeze, stretch arms, wave hand, wipe glasses
- Number of poses: 3, face to the camera, face to the left, face to the right
- Number of locomotion types: 2, standing, walking
- An extra horizontally flipped version has been synthesized for each of the videos.
- Number of videos in total: 20x8x3x2x2=1920

Each of the 1920 videos has a frame rate of 10 fps and a resolution of 480x290 pixels. The video ranges from 3 second to 10 second. We clipped videos to the first 6 second if the video length is longer.

3.2. Data Pre-processing

For CNN-based models, we extract 2 frames from each second of video, so each video converts to 12 or fewer images. For videos with fewer than 12 extracted images, we pre-pad them to length of 12. Then, we crop out the subject from the frame pictures to reduce background noise. To do so, we use Mask R-CNN [7] with Detectron2 and pre-trained MS COCO model. We also resize and pad the images to (224, 224) for faster computation and easier integration with VGG16 feature extraction later.

For HRNet-based models, we extract 10 frames from each second of video, so each video converts to 60 or fewer images. For videos with fewer than 60 extracted heatmaps, we pre-pad them to length of 60. The extracted heatmaps are of size 64x48x17.

3.3. Training/validation/test split

To evaluate model performance on the same basis, we split the data following Thi et al.'s previous work [17]. Specifically, the videos from subjects S002-S006 are used for testing and the remaining subjects are used for training.

For the videos used for training, we further split them to 70% training and 30% validation. The videos are shuffled first so this split is random.

4. Methods

In this section we describe the various deep learning based methods used to detect flu-like symptoms from videos.

4.1. CNN + LSTM (Baseline 1)

The whole network architecture is described in Figure 3. The model runs every image of the sequenced video frames through a Convolutional Neural Network (CNN). The CNN feature extractor (Figure 2) is composed of a series of Conv 3x3 layers, Batch Normalization layers and Max-Pooling 2x2 layers.

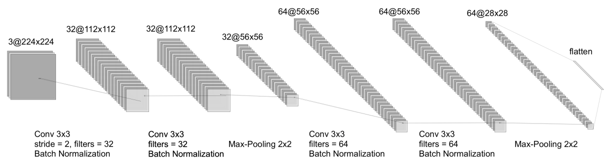

Figure 2. CNN Feature Extractor for CNN + LSTM model

Next, we feed the flattened outputs as a sequence into a Long Short Term Memory Recurrent Neural Network (LSTM).

LSTM is a special kind of RNN that is capable of learning long-term dependencies. The LSTM unit uses hidden state (h) and cell state (c) variables to encode its states, and uses 4 gates variables - input (i), forget (f), output (o) and gate (g) - to control whether or not to memorize certain information. The LSTM architecture is given by the following equations:

$$\begin{aligned}
i_t &= \sigma(W_x^{(i)} * x_t + W_h^{(i)} * h_{t-1}) \\
f_t &= \sigma(W_x^{(f)} * x_t + W_h^{(f)} * h_{t-1}) \\
o_t &= \sigma(W_x^{(o)} * x_t + W_h^{(o)} * h_{t-1}) \\
g_t &= tanh(W_x^{(g)} * x_t + W_h^{(g)} * h_{t-1}) \\
c_t &= f_t \circ c_{t-1} + i_t \circ g_t \\
h_t &= o_t \circ tanh(c_t)
\end{aligned} \tag{1}$$

For the baseline models, we take the output vector of the last LSTM unit and pass it to the final section, which consists of a dropout layer, a dense layer with Relu activation, another dropout layer, and a dense layer with Softmax activation function to produce the probabilities of every class.

Since we use Softmax as our final activation function, our loss function is categorical cross-entropy loss:

$$\begin{aligned}
L &= -\sum_i log(\frac{\exp s_i y_i}{\sum_j \exp s_j}) \\
s_i &= f(x_i; W)
\end{aligned} \tag{2}$$

Finally, we take the class with maximum probability as the predicted action.

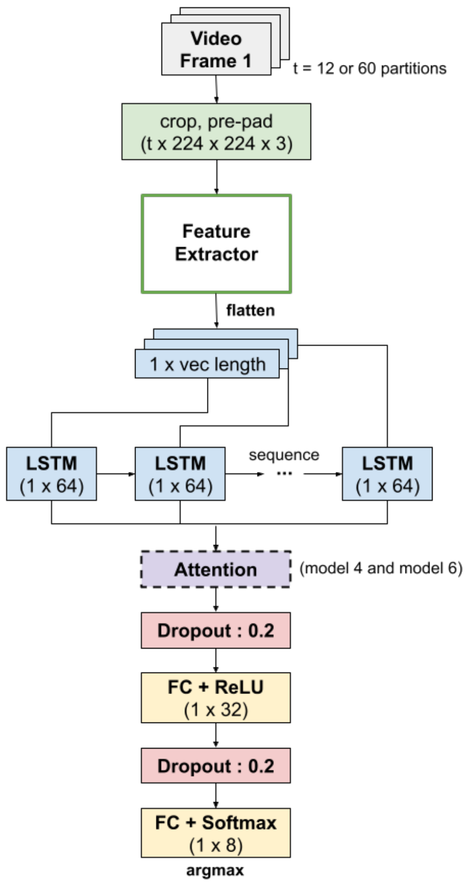

Figure 3. Network architecture

4.2. 3D-Conv (Baseline 2)

The second baseline model uses a series of 3D-Conv layer with kernel size 3 connected with a 3D max-pooling layer as the feature extractor. Then the flattened vector is fed into the Dense + Dropout layers described in Figure 3.

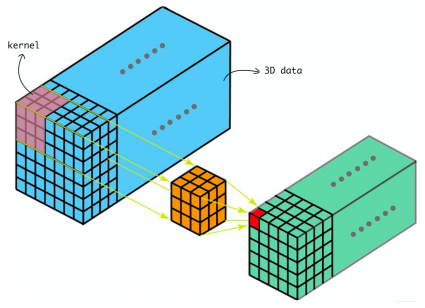

Figure 4. Example of a 3D convolution performed with 3D kernel and 3D data - https://towardsdatascience.com/9d8f76e29610

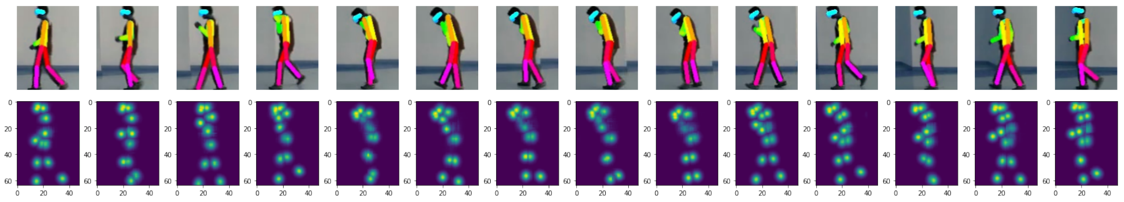

Figure 5. Top row is extracted video frames with HRNet visualization. Bottom row is the overlap of heatmaps from 17 channels.

The 3D convolution is performed with 3D kernel and 3D data that 2D video frame images merged to. An example of a 3D convolution is presented in Figure 4.

4.3. VGG-16 Features + LSTM

We modified the CNN + LSTM (Baseline 1) model by utilizing transfer learning from a pre-trained VGG-16 [15] network. VGG is a classical Convolutional Neural Network architecture that is characterized by its simplicity and efficiency. The only other components of VGG network are pooling layers and dense layers.

The VGG-16 feature extractor helps the network to learn more intricate features of the input video frames. The last dense layer of the VGG-16 model is fed into the LSTM + Dropout + Dense layers described in Figure 3.

4.4. VGG-16 Features + LSTM + Attention

This model is similar to the VGG-16 Features + LSTM model. The only difference is that we kept the output vectors of all LSTM units and used an Attention layer to let the model decide the importance of each time step.

The Attention mechanism is described by Bahdanau et al. [1] as follows:

$$\begin{aligned} e_t &= h_t w_a \\ a_t &= \frac{\exp e_t}{\sum_{i=1}^{T} \exp e_i} \\ v &= \sum_{i=1}^{T} a_i h_i \end{aligned} \tag{3}$$

Here h_t is the output of LSTM unit at time step t and w_a is the weight matrix for the attention layer. The attention importance scores for each time step, a_t, are obtained by multiplying the representations with the weight matrix and then normalizing to construct a probability distribution over time steps. Lastly, the representation vector for the video sequence, v, is calculated by a weighted summation over all the time steps using the attention importance scores as weights.

4.5. HRNet Features + LSTM

Human pose estimation, defined as the problem of localization of human joints and body parts in images, is often essential to the classification of human actions. High-resolution representations are needed for position sensitive tasks like human pose estimation, which makes High Resolution Net (HRNet) [20] a state of the art neural network for such task.

The HRNet maintains high-resolution representations by connecting high-to-low resolution convolutions in parallel and strengthens high-resolution representations by repeatedly performing multi-scale fusions across parallel convolutions. An illustration of the HRNet architecture is presented in Figure 6.

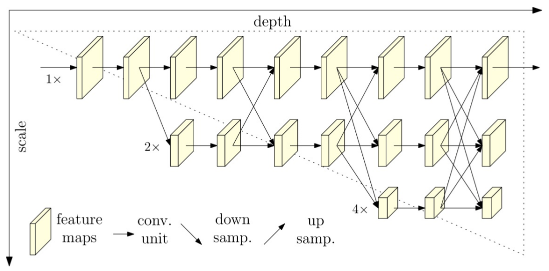

Figure 6. An illustration of the high-resolution network architecture - https://jingdongwang2017.github.io/Projects/HRNet/

This proposed model runs every frame of the input video through a pre-trained HRNet and outputs a heatmap of size 64 x 48 and 17 channels for each image. The 17 channels correspond to the 17 key-points of human body. Sample extracted HRNet features are presented in Figure 5.

Using pre-trained HRNet as the new feature extractor, we pass the flattened heatmap to the LSTM + Dense layers to obtain the action with maximum probability.

4.6. HRNet Features + LSTM + Attention

For this model, we take the structure of the HRNet Features + LSTM model and add an Attention layer after the LSTM layer.

The overall architecture of this model is again described in Figure 3. The description of the model's components is omitted here as they are already covered in the previous sections.

5. Results & Discussion

We convert the predicted action label into a binary label of coughing and sneezing actions. Since the binary label is biased, we evaluate the performance of our models using Precision (Prec.) and Recall (Rec.), which are calculated as $\frac{TP}{TP+FP}$ and $\frac{TP}{TP+FN}$, respectively. To compare with the previous work by Thi et al., we also adopt a different accuracy measure of $\frac{TP}{TP+FP+FN}$, which can be regarded as a lower-bounding summary of the (precision, recall) pair.

The result of binary classification on test dataset is reported in Table 1. The two baseline models perform worse than the handcrafted traditional feature extraction methods. The two models using VGG-16 Features have slightly better performance compared with the previous work. And finally the two models using HRNet Features perform significantly better than all other models. In our case, adding an Attention layer after LSTM layer did not help improve the accuracy of models.

Model	Prec.%	Rec.%	Acc.%
cuboid + AMK II	55.3	62.1	41.3
HOGHOF + AMK II	58.9	64.4	44.4
CNN + LSTM	30.3	45.0	22.1
3D-Conv	37.7	52.5	28.1
VGG-16 + LSTM	68.7	75.0	55.9
VGG-16 + LSTM + Attn	55.7	80.8	49.2
HRNet + LSTM	**85.5**	**96.7**	**83.5**
HRNet + LSTM + Attn	84.7	87.5	75.5

Table 1. Comparisons of recognition accuracies of Sneeze-Cough actions

Below we present the validation and test accuracy of multi-class action classification over training epochs. In the test dataset, the number of samples from each class is the same, so accuracy is a balanced performance measure. We also show the multi-class action recognition confusion matrix based on test dataset to better understand where the errors are occurring.

5.1. CNN + LSTM (Baseline 1)

Figure 7 shows training and validation accuracy over 60 epochs. Although training accuracy has not converged yet after training completes, we started to see validation accuracy flattens and the gap between training/validation curves gets larger. We decided to end training early so as to prevent overfitting.

We applied L2 regularization on Convolutional layers and added dropout layers to reduce overfitting. The batch normalization layers and max pooling layers in the model also help reduce overfitting, even thought this is not their sole purpose. The hyperparameters are tuned to increase validation accuracy while maintain a small gap between training and validation curves.

Based on Figure 7, overfitting still exists. This is probably because the training dataset is small and we are training the model from scratch. We can potentially stop training even earlier or collect more training data to further reduce overfitting.

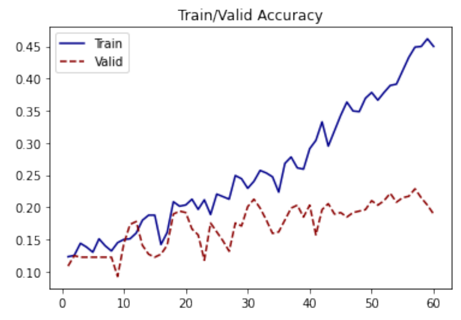

Figure 7. Train/Test accuracy for CNN + LSTM model

Figure 8 presents the confusion matrix on test dataset. The CNN + LSTM baseline model tends to predict certain classes over the other ones.

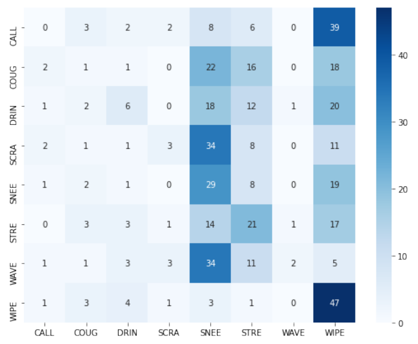

Figure 8. Confusion matrix on test dataset for CNN + LSTM model

For implementation, we used a mini-batch Adam optimizer [10] with a learning rate of 1×10^{-4}. Through a series of experiments, we compared and contrasted Adam with other stochastic optimization methods and found Adam to be the best overall choice in our case. We also saw that a batch size of 20 worked reasonably well for model's convergence, and a learning rate of 1×10^{-4} performed the best in terms of avoiding fluctuating curves or slow learning. We adopted this optimizer for all the models.

5.2. 3D-Conv (Baseline 2)

Figure 9 shows training and validation accuracy over 60 epochs. Similar to CNN + LSTM baseline model, we decided to end training early to prevent overfitting.

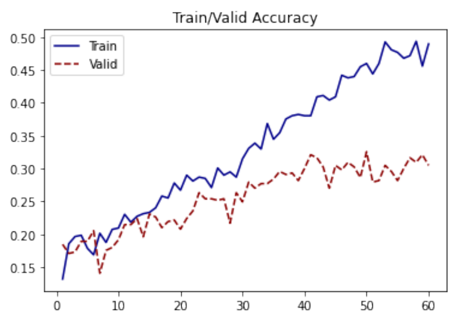

Figure 9. Train/Test accuracy for 3D-Conv model

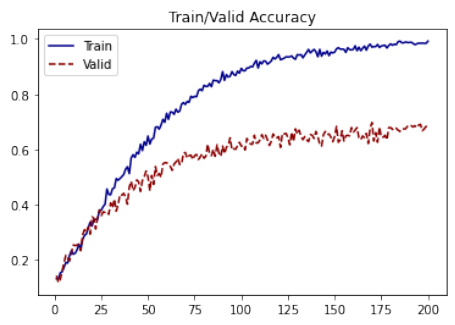

Figure 11. Train/Test accuracy for VGG-16 Feature + LSTM model

Figure 10 presents the confusion matrix on test dataset. The 3D-Conv baseline model has more balanced predictions across different classes compared with the CNN + LSTM baseline model. However, there are still certain classes (e.g. CALL) that have very few predictions.

The observations and handling about overfitting is similar to the CNN + LSTM baseline model, so the details are omitted here.

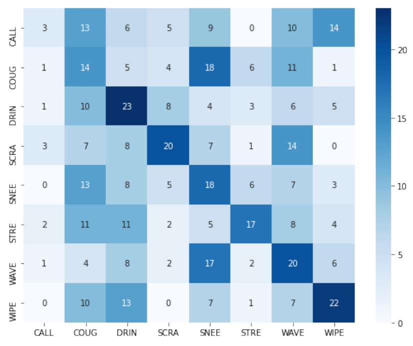

Figure 10. Confusion matrix on test dataset for 3D-Conv model

5.3. VGG-16 Features + LSTM

Figure 11 shows training and validation accuracy over 200 epochs. The training and validation accuracies are flat and stable after around 125 epochs.

We added dropout layers to reduce overfitting. Since we fixed the pretrained VGG-16 network when we do transfer learning, this helps reduce overfitting as the VGG-16 network was trained on a different and larger dataset.

The problem of overfitting appears to be better than the baseline models but it still exists. This is probably because the training dataset is small. We can potentially collect more training data to reduce overfitting.

Figure 12 presents the confusion matrix on test dataset. The confusion matrix looks reasonable. Certain class pairs are more likely to be predicted incorrectly. For example, SNEE (sneezing) and COUG (coughing) are often confused with each other. This is probably because these actions look similar or involve similar set of actions (e.g. covering mouth).

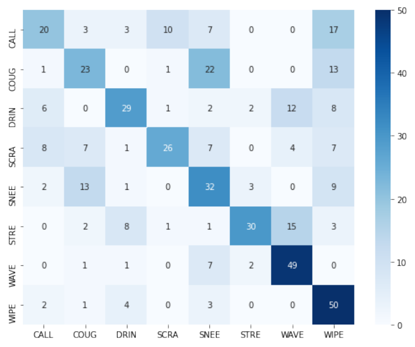

Figure 12. Confusion matrix on test dataset for VGG-16 Feature + LSTM model

Results and discussions about the VGG-16 Features + LSTM + Attention model are omitted here as they are very similar to the ones in this section.

5.4. HRNet Features + LSTM

Figure 13 shows training and validation accuracy over 30 epochs. The training and validation accuracies converged a lot faster compared with the previous models. Additionally, the converged accuracy is a lot higher than that of the other models. This is probably because the extracted human pose features are very indicative of human actions. We also extracted more frames in the pre-processing step for this model, so the additional frames may help capture perceptible actions more accurately.

We added dropout layers to reduce overfitting. Since we fixed the pretrained HRNet when we do transfer learning, this helps reduce overfitting as the HRNet was trained on a different and larger dataset.

The problem of overfitting appears to be better than the baseline models but it still exists. This is probably because the training dataset is small. We can potentially collect more training data to reduce overfitting.

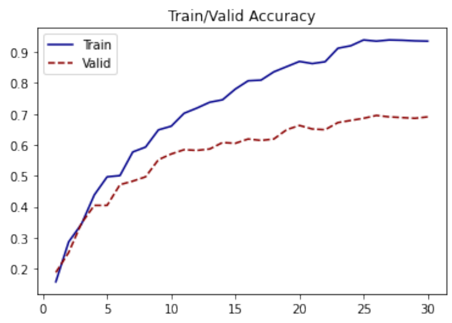

Figure 13. Train/Test accuracy for HRNet Feature + LSTM model

Figure 14 presents the confusion matrix on test dataset. The confusion matrix looks pretty good. Similar to the VGG-16 + LSTM model, certain class pairs are more likely to be predicted incorrectly.

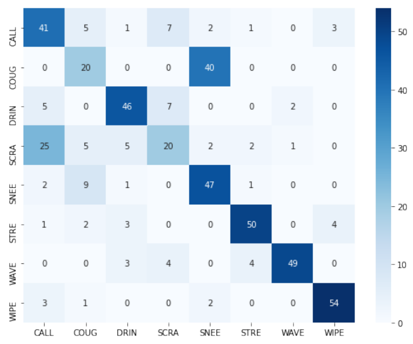

Figure 14. Confusion matrix on test dataset for HRNet Feature + LSTM + Attention model

Results and discussions about the HRNet Features + LSTM + Attention model are omitted here as they are very similar to the ones in this section.

Additionally, we look at examples of failure cases of our best-performing model, the HRNet Features + LSTM model, to understand why it is failing.

We present two cases where the model incorrectly categorized COUG as SNEE in Figure 15, and two cases where the model correctly categorized SNEE in Figure 16. This COUG-SNEE pair corresponds to the darkest non-diagonal cell in Figure 12, which is the place where the model makes most mistakes.

As we can see in the figures, the coughing and sneezing actions performed by the subjects are reasonably close. Both actions involve similar movements of body's key points. For example, the hand key points lift up and the head key point goes down. It is hard for the models to distinguish between these similar actions.

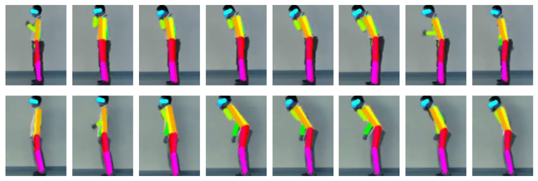

Figure 15. The two rows are extracted video frames of subject S005 and S006 performing coughing actions. The HRNet Features + LSTM model incorrectly categorized them as sneezing.

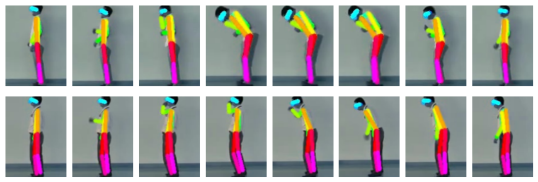

Figure 16. The two rows are extracted video frames of subject S005 and S006 performing sneezing actions. The HRNet Features + LSTM model categorized them correctly.

6. Conclusion

In this work, we explored and compared several deep learning based methods for recognizing flu-like symptoms from video data. The two baseline models are CNN + LSTM model and 3D-Conv model. Then we improved the CNN + LSTM model by replacing the CNN feature extractor with VGG-16 (model 3) and HRNet (model 5). Finally, we experimented with the Attention mechanism (model 4 and model 6).

The baseline models did not perform very well because their structures are relatively simple and they are trained from scratch. This makes the baseline models more subject to overfitting as well. The VGG-16 + LSTM model performs better than the benchmarks because transfer learning from VGG-16 helps the model learn more intricate features and makes it less likely to overfit. Our best performing model, the HRNet + LSTM model, surpasses the benchmarks significantly because of its utilization of human pose features. We also explored adding Attention mechanism to the models but it did not seem to improve the models' performance.

7. Future Work

Given more time and resources, we would continue to tune the architecture of our proposed models. Accuracy may be further improved with more hyper-parameter tuning and investigation of other optimizer types. We can also extract more frames (e.g. 60) for the baseline models and VGG-16-based models. Currently only 12 frames are extracted because of the memory constraint of our machines.

We would like to extend the 3D-Conv model (baseline model 2) by combining 3D Convolutional layers with Attention mechanism to see if it helps improve the model. Last but not least, we would like to collect more training data, which we believe will mitigates the overfitting issue.

We are also interested in investigating the impact of extra source of information. For example, if we consider both the extracted frames from video data as well as the sound signals from video data, we may be able to train better-performing models.

This project is extensible in terms of its application as well. For example, we may be able to use the same network architecture for other action-recognition tasks, like automatic detection of violence activities based on video data. The network for the new task can share similar low-level features as this task, and just have the last few layers to be different and trainable.

8. Contributions & Acknowledgements

All team members contributed equally to this project.

Alan pre-processed images with Mask R-CNN, trained HRNet + LSTM model, and wrote the final report. Yechao extracted HRNet features from images, trained CNN + LSTM model, trained 3D-Conv model, and trained VGG-16 + LSTM model.

We made use of the following public code while working on this project: Detectron2 by Wu et al. [22], HRNet by Sun et al. [16], and HRNet PoseTrack by lxy5513 [12]. Please see References section for links to the original repo.

All supporting files for this project are available at: https://github.com/alanlou/cs231n-project.

References

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2016. 4

[2] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset, 2018. 2

[3] David M. Cutler and Lawrence H. Summers. The COVID-19 Pandemic and the $16 Trillion Virus. *JAMA*, 324(15):1495–1496, 10 2020. 1

[4] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *2005 IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 65–72, 2005. 1

[5] Jeff Donahue, Lisa Anne Hendricks, Marcus Rohrbach, Subhashini Venugopalan, Sergio Guadarrama, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description, 2016. 2

[6] Rohit Girdhar, Deva Ramanan, Abhinav Gupta, Josef Sivic, and Bryan C. Russell. Actionvlad: Learning spatio-temporal aggregation for action classification. *CoRR*, abs/1704.02895, 2017. 2

[7] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. *CoRR*, abs/1703.06870, 2017. 2

[8] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):221–231, 2013. 1

[9] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014. 2

[10] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. 5

[11] Ivan Laptev, Marcin Marszalek, Cordelia Schmid, and Benjamin Rozenfeld. Learning realistic human actions from movies. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008. 1

[12] lxy5513. Posetrack by hrnet. https://github.com/lxy5513/hrnet, 2019. 8

[13] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local svm approach. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 3, pages 32–36 Vol.3, 2004. 1

[14] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. *CoRR*, abs/1406.2199, 2014. 2

[15] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 4

[16] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Hrnet-semantic-segmentation. https://github.com/HRNet/HRNet-Semantic-Segmentation, 2019. 8

[17] Tuan Hue Thi, Li Wang, Ning Ye, Jian Zhang, Sebastian Maurer-Stroh, and Li Cheng. Recognizing flu-like symptoms from videos. *BMC Bioinformatics*, 15(1):300, 2014. 1, 2

[18] Tuan Hue Thi, Li Wang, Ning Ye, Jian Zhang, Sebastian Maurer-Stroh, and Li Cheng. Recognizing flu-like symptoms from videos, 2014. 2

[19] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks, 2015. 2

[20] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu, and Bin Xiao. Deep high-resolution representation learning for visual recognition. *CoRR*, abs/1908.07919, 2019. 4

[21] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition, 2016. 2

[22] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 8

[23] Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher Pal, Hugo Larochelle, and Aaron Courville. Describing videos by exploiting temporal structure, 2015. 2

[24] Yi Zhu, Zhenzhong Lan, Shawn Newsam, and Alexander G. Hauptmann. Hidden two-stream convolutional networks for action recognition, 2018. 2