DeepCough: Classifying Cough and Speech from FluSense Dataset with Residual Neural Network

Khoa Le
University of Massachusetts, Amherst
Amherst, MA 01003

1. Idea

Amid the coronavirus crisis, there is a rising need for smartphone applications that assist users with performing social distancing. The applications can come in many forms, ranging from location-based apps using GPS, distance-based apps using Bluetooth to audio-based apps. One important application, which is the focus of this project, is cough detection. Advances in this type application will greatly reduce the risk of users being infected by prompting people to increase distance between each other, as well as alerting users of their current health conditions.

Audio signals can be transformed into spectrograms, which represent the spectrum of frequencies as time varies. Classifying different types of audio signals therefore can be rephrased as classifying between different spectrogram images. For the purpose of identifying cough, this application can significantly benefit from state-of-the-art convolutional neural network (CNN) models to classify spectrograms of cough from that of speech. Therefore, the approach of this project is to learn a model, based on a well-known CNN named Residual Neural Network or resnet [1] that can automatically classify cough from speech.

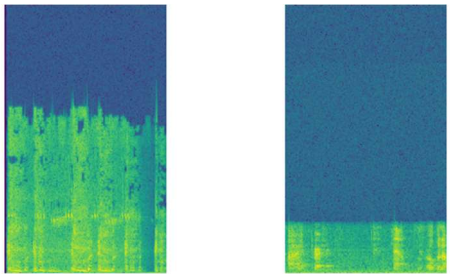
The data used for this project are from the FluSense [2] dataset, which provided the raw audio data of cough and other activities like speech and breathe as well as their detailed annotations. This project will only focus on classifying two main audio signals that are the most prevalent in the dataset, as well as in real situations, namely cough and speech. The resnet model is obtained from using Keras and TensorFlow libraries in Python, and the use of the model in training and inference in the codes is heavily inspired by the DeepWeeds project [3] and its GitHub repository.

2. Technical Approach

The technical aspect of this project involves three important phases: spectrogram images generation, training a residual neural network for classification of cough and speech and evaluating this network's performance in terms of accuracy.

2.1. Spectrogram Image Generation

The raw audio .wav files obtained from FluSense [2] were passed through a Python script that performed the following actions:



1a 1b

Figure 1. Spectrograms of Cough (a) and Speech (b)

- The sample data and sample rate were extracted from .wav files using scipy library and used to compute spectrograms with consecutive Fourier Transforms, where the applied window size was 20 and step size was 10.
- Using the annotation files provided by FluSense [2], the intervals where sounds of interest like coughing and speech appeared were obtained. The parts of the spectrogram that corresponded to these intervals were retained, while other parts were discarded. The script kept these parts to be of equal sizes (265 time points by 442 frequency bands) in order to be input into the neural networks.
- Spectrograms where there were no signal intensities at any frequency for half of the time frame were discarded as well to avoid noise.
- The spectrograms were saved as images of equal sizes (256 x 256 pixels) in the **processed-data/** directory. Figure 1. Shows examples of these images.

Olsen et al. detailed several image augmentation strategies that would be useful to implement in this project [3]. ImageDataGenerator in Keras was used to perform the following enhancements:

- For rotation and scale, images were resized to 256×256 pixels, arbitrarily zoomed in from 50% to 100%, and rotated within the range of $\pm 360^{\circ}$ [3].
- Images for training and validating were also randomly flipped horizontally before rescaled to 224×224 pixels to input into the learning models [3].

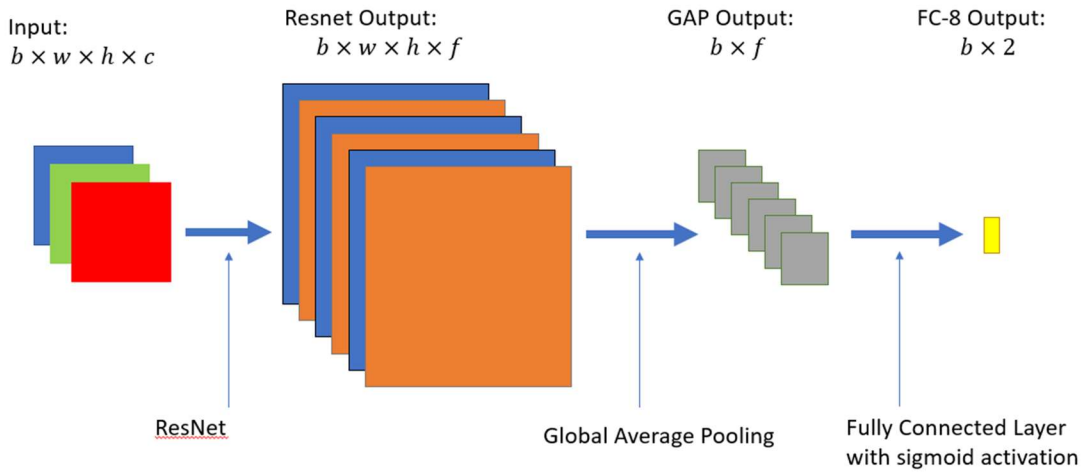


Figure 2. Architecture of training pipeline

2.2. Training a Residual Neural Network

The data were segregated into training dataset, validation dataset and test dataset. The training set is for the ResNet to learn of the features within the spectrogram images, the validation set is for cross validation and improving the generalization of the model, and the test set is to evaluate the network accuracy for this classification task. The ratio of train-val-test is 60-20-20. There were more speech samples than cough samples, so some speech samples were discarded randomly in order to the two classes to have balanced labels.

This project performed training and validating separately on the Residual Neural Network pre-trained with ImageNet weights using Keras's base implementation [1]. The input into architecture was a matrix of size $b \times w \times h \times c$, where b denotes the batch size (32 was chosen), $w \times h$ is the size of the images (256 x 256), and c is the number of the color channels (3). The outputs of the network were of shape $b \times w \times h \times f$, where f denotes "the number of extracted features for each $w \times h$ spatial location" [3]. The outputs were then passed through a global average pooling layer to minimize overfitting in the resulting model. The learning process was optimized by Keras implementation of Adam Optimizer with a starting learning rate of $1e^{-14}$. The learning rate was set to be dynamic, as it reduced by 50% for every 16 epochs that the classification error on the validation set did not decrease [3].

Finally, in order to classify the Cough and Speech spectrograms, the base implementations of ResNet were slightly modified to replace the ImageNet output layer with a 2-neurons fully connected layer, followed by sigmoid activation, corresponding to the 2 different types of audio signals the project focused on. The weights and biases are initialized using uniform random distribution. The use of sigmoid is preferable in this experiment since it correspond to the probability that a spectrogram image is categorized as a certain class [3]. The details of this pipeline are illustrated in Figure 2.

2.3. Evaluation Method

The primary evaluation metric was F1 score, in which it seeks a balance in our case where there is a little imbalance within the dataset. Remember that F1 score, ranging between 0 and 1, is calculated based on other two metrics, precision and recall as follow:

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

3. Evaluation

Class	Pred_Cough	Pred_Speech
True_Cough	100	49
Tru_Speech	38	111

Table 1. ResNet Evaluation Confusion Matrix

The preliminary result is listed in Table 1. in the form of a confusion matrix. The outcome indicates that the network

was able to classify more correct samples than incorrect ones. A detailed record of the model's performance on individual class with metrics like F-1 score, Recall and Precision is seen in Table 2. The weighted average F-1 score of the two classes were 0.707, which is impressive given the fact that the size of the training data set is relatively small and pre-processing transformations applied on the images were rudimentary.

	Precision	Recall	F-1
Cough	0.725	0.671	0.697
Speech	0.694	0.745	0.718
Weighted Average	0.709	0.708	0.707

Table 2. ResNet Evaluation Metrics

4. Limitation

There are limitations in this project which could be resolved if more time and resources were provided. First and foremost, the project was not able to convert the generated ResNet model into a full-fledged mobile application due to its sheer size (about 275 Mbs at the moment). This has been an emerging problem in the field, as experts called edge computing, which is heavily researched. However, there are resources online that detail the instructions to transfer the model to an Android application, which can be the future improvement for this project. At the moment the model is stored in a separate GoogleDrive as its size exceeds the limit allowed on GitHub.

In addition, since a smartphone application was not developed, not a lot of though has been put into the user interfaces. The author imagined that this app would be running in the background rather than being used actively by mobile users, so the UI/UX could be very simple. Lastly, the inference time might be an issue, since it took roughly 4.7 seconds to pass a wav file through the whole data pipeline and the model for the inference to be generated. Because of this, the real-time detection capability is compromised.

Since the deliverable of this project was not a smartphone app, a demo video was also not included. However, detailed instructions were provided on the GitHub to reproduce the results obtained during evaluation.

5. Conclusion

This project served the purpose of a proof of concept for applying deep neural network to cough detection, specifically in this case was using Residual Neural Network to classify cough and speech sounds. The model was able to achieve a weighted average F-1 score of 0.707 with only simple signal transformations being applied. There are limitations which if resolved, would significantly contribute to the ongoing researches in social distancing applications in the battle against COVID-19.

6. Miscellaneous

- Scripts and instructions to run the code: https://github.com/Khoale1096/resnet_cough_classifier
- ResNet model Repository: https://drive.google.com/file/d/1XW05IqaFKksJnCq_19DP_RonQG_Wixb-/view
- FluSense annotations: https://github.com/Forsad/FluSense-data
- For raw audio signals please contact FluSense authors to obtain the link.

References

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[2] F.A. Hossain, A. A. Lover, G. A. Corey, N. G. Reich, and T. Rahman, "FluSense: A Contactless Syndromic Surveillance Platform for Influenza-Like Illness in Hospital Waiting Areas," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies,* vol. 4, no. 1, pp. 1-28, 2020.

[3] A. Olsen, D. A. Konovalov, B. Philippa, P. Ridd, J. C. Wood, J. Johns, W. Banks, B. Girgenti, O. Kenny, J. Whinney, B. Calvert, M. R. Azghadi, and R. D. White, "DeepWeeds: A Multiclass Weed Species Image Dataset for Deep Learning," *Scientific Reports*, vol. 9, no. 1, 2019.