Deep Learning for Edge-AI Respiratory Disorder Monitoring

Simon Aquarone, Zhengru (James) Fang, Sid Lamichhane *

December 20, 2024

***Abstract*— The frequency of coughing is commonly used as the primary measure of effectiveness in clinical trials of treatments for chronic coughing. Noninvasive, real time cough counting directly on a wearable, embedded device is a fitting solution. This paper explores Edge-AI to implement deep learning architectures based on CNNs for cough classification while considering limited memory for inference. It additionally analyses the importance of input signals and preprocessing methods on the classification performance of cough detection. The CNN models conceived are able to classify coughs with a similar performance as large models but with less than 16,000 parameters when the input signals, preprocessing techniques and window length are correctly chosen.**

I. Introduction

In this report we present lightweight deep neural network (NN) architectures for cough detection. Their small number of parameters (<16,000) allow them to fit on the small, embedded devices created by the Embedded Systems Lab (ESL) at EPFL. These devices are worn on the chest and must perform real-time and low-power monitoring.

Cough detection is a critical step to measuring cough frequency, which can be used as a marker to monitor the severity of respiratory conditions. [1] A cost-effective, non-invasive method for continuously monitoring the cough patterns of patients is needed to provide individualized care. Coughs should be detected in real-time, energy-efficient, and privacy-preserving manner.

With this in mind, the Embedded Systems Laboratory (ESL) of EPFL has been developing a device that can be worn on the chest and records audio and movement data. The data is processed "at the edge" which means on the device itself. Coughing can be detected through the sound and chest movement it produces, but it can be difficult to distinguish a cough from other similar sounds or movements, such as laughter.

In order to adapt to the RAM constraints, ESL requires that the neural network contains less than 16,000 parameters. Parameter-heavy networks consume greater power during inference unlike smaller networks. It is crucial to determine the best architecture and preprocessing approach so that the neural network is efficient in extracting features from the input signals. Our tasks were to first determine which data preprocessing to use, then to determine which signals were the most impactful, and finally perform window length optimization. We also explored the impact of downsampling the audio data. We present architectures, all based on convolutional neural networks (CNNs), for three types of inputs: 2-microphone audio, inertial measurement unit (IMU), and combined audio-IMU.

*This work was performed in collaboration with the Embedded Systems Laboratory of EPFL which also provided the used data.

II. Methods

A. Data set description

The cough detection data set provided by the ESL consists of 16 subjects. Each subject produced forced cough sounds and 4 “parasitic sounds”: speech, deep breathing, laugh, and throat clearing. These four parasitic sounds have similar audio and chest movements to coughing. The data is recorded under four different background scenarios and two kinematic scenarios. The background noise scenarios are no noise, traffic, another person coughing in the background, and loud drum music. The kinematic scenarios are sitting and walking. The 16 subjects produced 21,590 total samples, of which around 20% are coughs. All subjects except one produced between 150 and 450 cough samples. For each cough there are two input devices: audio input and IMU input. For audio input two microphones were placed, one facing outward and the other facing inward (towards chest), both sampled at 16 kHz. The IMU input consists of six channels: X, Y, Z accelerometer and Y, P, R gyroscope measurements sampled much slower at 100 Hz due to the different nature of IMU signals.

B. Data augmentation

Deep neural networks require a large training dataset to achieve satisfying performance. As there are only 16 subjects, ESL applied data augmentation of the raw biosignals. The data augmentation technique used is to time shift the raw signals to the left or right by a random amount. The number of times this is done is determined by an augmentation factor. We chose 2 as the augmentation factor meaning the time shift was done twice for each biosignal sample.

C. Data prepossessing

Deep learning models have to be effective for classifying audio samples, even with a high noise, making interesting to explore for a cough detection device. [2] The microphone measures the amplitude of sound at fixed time intervals. This is the raw audio data that can be fed into a neural network and trained. Transforming audio signals into images and then using the same networks that are effective in image classification yield excellent results in the audio domain. [3]

Fourier transforms only give you a spectrum of frequencies, to data on time. This is why Short-Time Fourier Transforms (STFT) and Mel-frequency cepstrum (MFCC) are popular for audio classification. These two transformations work by taking smaller windows and calculating the Fourier transform on each of these small windows to obtain their frequency spectrum. When combining all these Fourier transforms, a spectrogram is obtained as shown in Figure 2. The difference between STFT and MFCC is that MFCC uses the Mel Scale instead of the frequency scale and uses the decibel Scale instead of amplitude. The Mel scale is a scale of pitches that mimics the way humans perceive sound using a logarithmic scale and therefore contrast is greater than with STFT.

The choice for the window length of the STFT and MFCC is important as the sampling frequency is fixed by the microphone (16kHz) [4]. Due to the uncertainty principle [5], the choice of the window size is a trade-off between frequency resolution and time resolution. Increasing the window size increases the resolution in the frequency domain, but decreases it in the time domain. After trying multiple values and seeing which gave the most distinct spectrograms, N=256 was chosen as the window size. The uncertainty principle can be partly counteracted by taking windows that overlap each other, separated by H samples. The hop length H=N/2 is chosen as a good trade off between the increase in time resolution it provides and the increase in data volume it causes which is especially important to consider for Edge applications. Applying STFT or MFCC on our data with these N and H as parameters gives an image of size 129x88 (or 128x88 for MFCC) for each audio channel. If both the inner and outer microphones are used, the spectrograms at the input of the CNN are 2x129x88. These sizes are considering the 0.7 second overall window. For the IMU data, the global training set mean and standard deviations in all six channels were computed, then used to normalize

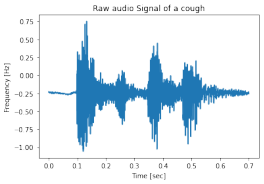

Fig. 1. Audio data from outer microphone of a cough sampled at 16kHz with a window of 0.7 seconds

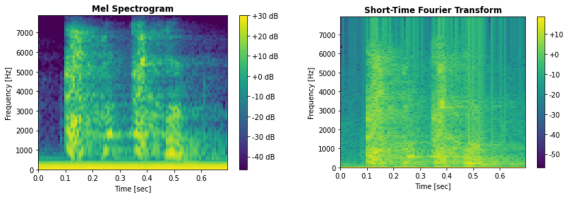

Fig. 2. MFCC and STFT of the cough audio sample shown in Figure 8 with N = 256 and H = N/2

the input for train and validation set. For the audio data we chose not to normalize it as audio data is much more dependant on exterior environment factors such as the room, the pressure, temperature and background sound levels. As the dataset we are given was collected in a set environment, for generalization purposes it was better to have the model learn on non-normalized data.

D. Training and validation

A criteria was that all samples from one subject must be used exclusively in the training or validation dataset but must not be in both. To classify correctly on novel subjects, a model must be able to accurately predict the output for new data based on the patterns and relationships it has learned from the training data. If a model overfits the training data, it may perform poorly on novel subjects due to it learning patterns that are specific to the training data set which may not generalize to new subjects. On the other hand, if a model underfits to the training data, it may also perform poorly on novel subjects because it has not learned enough about the underlying patterns and relationships in the data. Therefore, it is important to strike a balance between overfitting and underfitting in order to build a model that can classify correctly on novel subjects.

The subjects are split a 80-20 training-validation split. Two different training-validation splits are done for each test of a model architecture to compute the average performance reducing variances that arise from

the distribution of training/validation subjects. All tested models/architectures were trained with cross-entropy loss, using the Adam optimizer with a learning rate of 0.001 for 20 epochs, apart from IMU models which would underfit at 20 epochs, so we used 30 epochs. The chosen batch size was 32. Since coughs only composed 20 percent of the dataset, area-under-curve (AUC) score was preferred to raw accuracy during evaluation.

For the IMU data and raw audio data, 1D convolutional kernels of size 5 was used. For both MFCC and STFT, 2D convolutions with 5x5 kernels were used to capture patterns not only across time but also between frequency bands. Stride of 2 was used to decrease nummber of parameters.

III. Results

Unless specified, a window length of 0.7 seconds and a sampling frequency of 16kHz was used.

A. Audio preprocessing

TABLE I
AUDIO TRANSFORMATION IMPACT ANALYSIS

Transformation	Parameter Size	AUC
Raw	15254	90.84%
STFT	15926	96.31%
MFCC	15926	97.32%

For models with only audio data as input, performance decreased in this order: MFCC, STFT and then raw audio data as can be seen in table I. The slightly better MFCC results are due to Mel spectrograms having more contrast and a cleaner, more distinct spectrogram than with STFT due to the logarithmic nature of the decibel scale. Meanwhile, raw audio data models using 1D convolutions performed on average around 6% worse than STFT and MFCC model which is expected due to the larger difficulty a CNN model has to extract features from time-series data.

B. Determining impactful signals

We can see from table II that the inner microphone is not as impactful as the outer, but still provides useful information as the model with both mics still outperforms the model with just the outer mic.

The resulting AUC when any IMU signal was turned off was higher than with all IMU signals active (see Figure 3), therefore the model was overfitting.

TABLE II
MICROPHONE IMPACT ANALYSIS

Used Microphone(s)	Parameter Size	AUC
Inner mic	15726	95.615%
Outer mic	15726	96.77%
Both mics	15926	97.32%

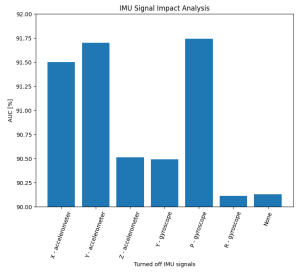

Fig. 3. Impact of turning off IMU signals on AUC

Many combinations of turned off signals were possible with the combined model. The inner microphone was less impactful, and x-accelerometer and p-gyroscope were the least impactful signals from the IMU. Leaving these 3 signals out of the input data for the combined model therefore gave us the best AUC score, better than without excluding these signals and nearly matching the audio model scores (see table III).

TABLE III
COMBINED SIGNAL IMPACT ANALYSIS

Audio Signals	IMU Signals	Parameters	AUC
Inner mic	accel and gyro	15747	93.87%
Outer mic	accel and gyro	15757	96.73%
Outer mic	**accel y,z and gyro r,y**	**15667**	**96.90%**
Outer mic	accel z and gyro r	15587	96.18%
All	All	15947	96.88%

C. Window length optimization

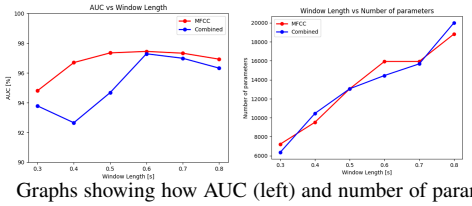

Fig. 4. Graphs showing how AUC (left) and number of parameters (right) change with window lengths

The main segment of a cough lasts on average 0.32 seconds. There are distinct sounds and chest movements that occur before and after a cough that could be

useful features for the models. Window length must be long but not too long such that the detection becomes delayed and impacts the real-time detection. A short window length means fewer parameters as smaller input size. The best performing models previously found (MFCC with both audio channels and the combined model with outer microphone, accelerometer y,z and gyroscope r,y as input signals) were tested. For window lengths of 0.6 and 0.8, only the number of input channels of the first fully connected layer had to be changed. But as window length decreased below that, the kernel sizes had to be changed from 5 to 3 due to the input data not being big enough.

As shown in Figure 4, the performance of the MFCC model decreases slowly as the window length becomes shorter than 0.6s. The performance of the combined model drops more sharply. The IMU CNN may not be able to effectively learn the difference between a cough and a laugh for example due to the samples being cut. The best AUC was achieved with a window length of 0.6 seconds for both models. Due to maxpooling, stride and the kernel sizes chosen for the convolutional layers of the MFCC model, the number of parameters for a window length of 0.6s is the same as for a 0.7s window. Looking at the graph on the right of Figure 4, we observe a linear relationship between the number of parameters as a result of the different window lengths.

D. Down sampling

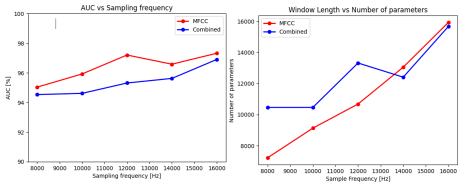

Fig. 5. Graphs showing how AUC (left) and number of parameters (right) change with different sampling frequencies for audio data

A cough's audio frequency spectrum is generally lower than the audible spectrum of 20kHz. This is why the ESL chose 16kHz for the sampling frequency as high pitched sounds are not necessary for cough classification. When the sampling frequency is decreased the size of input data decreases, and therefore the number of parameters decreases as well. When we downsample to lower than 14kHz, the kernel sizes of the convolutions for the audio CNN needs to be decreased to 3. Figure 5 shows that decreasing the sampling frequency causes an almost linear decrease in AUC up to 8kHz. Below 8kHz was not tested due to generability concerns. Indeed, even if AUC would continue to decrease linearly, models built with a sampling frequency below 8kHz would not generalize well to subjects with higher pitched coughs.

IV. Discussion and Conclusion

The goal of this project was to implement a deep learning model capable of effectively detecting coughs against similar perturbations while maintaining the number of parameters underneath 16,000 for implementation on low-power embedded devices. An analysis on raw audio data against MFCC and STFT was done, which gave us the overall best model for a 0.7s window: using MFCC on both audio signals (AUC of 97.32% achieved). We then studied models using only IMU data and another combining audio data with the IMU signals. IMU signals on their own achieved AUC scores of 5% less than audio models, but the combined model with STFT only had a 1% difference to the best MFCC model.

Signal importance showed having both microphones for audio models was best. For the combined model it was best if only the most impactful signals were kept as the AUC increased and the number of parameters decreased slightly. The analysis of window length and downsampling was done to explore ways to reduce the parameters. We found that a 0.6s window increased the performance of the MFCC model with both microphones to 97.44% AUC with 15926 parameters. The best combined model was also improved with a 0.6s window and achieved 97.28% AUC with 14443 parameters. These results were obtained from 16kHz sampling. Our downsampling study showed a trade-off that can be made between number of parameters and AUC using downsampling.

RNNs based on LSTMs were considered but disfavored due to high latency and parameter heaviness rendering them unsuitable for real-time cough detection, and also CNN were providing excellent results. Directed, forced coughs were used in the dataset as natural coughs are more difficult to source, but this raises the concern for generability of the models trained on such a dataset, as natural coughs may be less predictable. A future improvement would be to pre-train the models using the COUGHVID dataset [6]. Validation will be done on unseen data by ESL.

REFERENCES

[1] J. Hall, M. Lozano, L. Estrada-Petrocelli, S. Birring, and R. Turner, "The present and future of cough counting tools," Tech. Rep.

[2] "Audio Deep Learning Made Simple: Sound Classification, Step-by-Step, Doshi, Ketan." [Online]. Available: https://towardsdatascience.com/audio-deep-learning-made-simple-sound-classification-step-by-step-cebc936bbe5

[3] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. Weiss, and K. Wilson, "CNN Architectures for Large-Scale Audio Classification *," Tech. Rep.

[4] "Discrete Short-Time Fourier Transform (STFT), Muller, Meinard and Zalkow, Frank." [Online]. Available: https://www.audiolabs-erlangen.de/resources/MIR/FMP/C2/C2$_STFT-Basic.html$

[5] "Human hearing beats the Fourier uncertainty principle, Zyga, Lisa." [Online]. Available: https://phys.org/news/2013-02-human-fourier-uncertainty-principle.html: :text=The

[6] L. Orlandic, T. Teijeiro, and D. Atienza, "The COUGHVID crowdsourcing dataset: A corpus for the study of large-scale cough analysis algorithms," Feb. 2021, For more information about the data collection, pre-processing, validation, and data structure, please refer to the following publication: https://www.nature.com/articles/s41597-021-00937-4 The cough pre-processing and feature extraction code is available from the following c4science repository: https://c4science.ch/diffusion/10770/. [Online]. Available: https://doi.org/10.5281/zenodo.4498364

V. APPENDIX

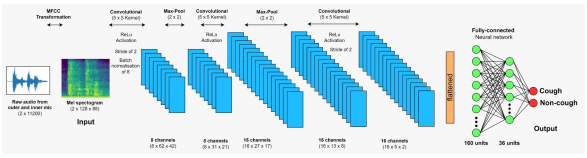

Fig. 6. Architecture of best classifier (MFCC) for audio

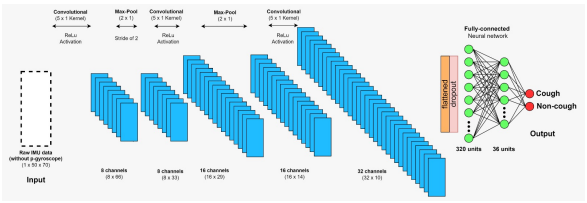

Fig. 7. Architecture of best classifier only considering IMU

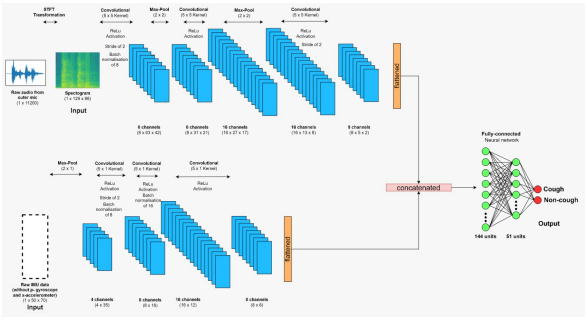

Fig. 8. Architecture of best classifier considering audio and IMU