UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

Computer Intelligenc

SARS-CoV-2: A Deep Learning Approach

Written by:

Ariosa Roberto
Belooussov, Nikita
Nuñez Picado, Andrey
Walzthöny, Eric

—

Sunday 16th January, 2022

Abstract

This project focuses on comparing Deep Learning methods such as Artificial Neural Networks (ANNs) as well as Convolutional Neural Networks (CNN), these are then compared to State of the Art models such as Category Embedding, TabNet, Models with Transfer Learning such as ImageNet50 and ResNet50. We use the Coswara Dataset which is a Dataset which contains about 2400 cough samples from COVID-19 patients. With this we want to compare the models and show that shallow CNN and ANN models can compete against SoTA models such models based with Transfer Learning from the aforementioned models. Albeit, we show that a CNN with Transfer Learning from ImageNet50 performs the best, our custom built ANN closely follows this result.

Keywords:

List of keywords SARS-CoV-2, COVID-19, Cough, ANN, CNN, Audio, Coswara-Cough,

1. Introduction

**A quick note on the introduction, we assume a prior knowledge on the workings of ANN and CNN in order to avoid going into too much repetition about their inner workings.*

We have been faced with numerous respiratory diseases which come from Human Coronaviruses (HCoV), these are a threat to our health[1]. These are usually mutated forms, which can happen when the virus ”jumps” or is transmitted to a host of a different species,also known as ”host switching”[4]. Since late 2019, we have been faced with SARS-CoV-2, which causes the respiratory diseases Coronavirus Disease-19 (COVID-19, which has affected millions of people.

What is also known is that the second symptom of COVID-19 is a cough (after fever) [3] since this disease also affects the respiratory system [2], we have taken advantage of this in order to produce models which are able to identify whether a person is positive with COVID-19, given a sample of their cough [5-7]. This means that one is able to predict, with a given accuracy and precision of the model, whether one has COVID-19 or not.

This type of tool is very useful at the time to perform an inexpensive evaluation of one’s health, simply uploading a sample of their cough [9].In various papers the methods used was to either extract features from the cough sample (in tabular data) or convert the image into a Mel-Spectrogram (image). For instance, it has been shown that using Machine Learning methods works well for classifying audio samples, or extracting features from them as well as using different Deep Learning techniques work well in order to accurately diagnose the individual based on a cough, voice or other type of sample related to voice.[9-11].

The objective of this was inspired from the aforementioned efforts of different institutions in order to create a tool that would work rapidly and inexpensively. We will therefore compare two different methods of Deep Neural Networks such as Artificial Neural Networks (ANNs) and Convolutional Neural Networks (CNNs). These models will be compared to SoTA models for image classification and tabular data classification. The models to which we will compare will be ImageNet50, ResNet50 for the CNNs and for the ANN we will use Category Embedding Networks, as well as TabNet, from pytorch_tabular, which have shown to be very effective [24,25], the comparison will be done by using Transfer Learning (TL) from the two aforementioned models (ImageNet50, ResNet50).

The experiments were done in the following environment with open source datasets, we compared these models using Google Colaboratory with a NVIDIA Tesla K80, and we used a openly available dataset called “Coswara Dataset" [31], the models were tracked using WandB, which is a great tool to manage multiple experiments. The Coswara dataset is a collection of about ~2400 cough, and sound samples, from patients that are positive or negative for COVID-19.

For the preprocessing of the data, we ensured that: 1) the audio sample was the same size 2) that there were no gaps in the middle of the audio sample, which means an incomplete Mel-Spectrogram. This was done using the librosa package for Python.

2. Previous work

There have been different implementations of these types of problems, but on a side note this happened to be a personal project (author #4), which I participated in with the Argentine Research Center (CONICET). I wanted to continue this work and improve the models that had been used as well as change frameworks from TensorFlow to PyTorch.

Apart from the personal side note, there have been successful stories about the implementation of the different papers using Deep Learning methods to evaluate whether their students were positive for COVID-19 [3-5]. Mainly, these sources come from universities which have developed this, and in some cases there have been governments which have put this at the public's disposal as well [3-5]. In these aforementioned works, the models that were mainly used were an ensemble of different models, which used different optimizers, configurations and loss functions [26]. Most of the models found in literature used a Convolutional Neural Network which is what we've decided to implement as well [3-5, 15,12, 26].

Different methods have been shown to work, especially when combined with Transfer Learning, to improve the efficacy of the original model. We wanted to initially test our own configurations of ANN and CNNs and then compare them with TL to compare how far off the models were.For the ANNs pytorch_tabular has been shown to be reliable and robust at the time of using tabular data, since it uses state of the art models mentioned before (CNE and TabNet) and in comparison the pre-trained models of ImageNet50 and ResNet50 are also leading in the are of image classification. For this reason we want to build from there and use them as our "golden standard" to compare [24].

The previous methods have established the following workflow, which we have used as well to facilitate the working with audio files and the chosen models:

1. Audio → Feature Extraction (WAV → CSV)
2. Audio → Mel-Spectrogram (WAV → JPG)

For the first path, we will use librosa, a python audio library, in which we can extract 26 different features from the audio sample. These will be then written to a CSV file and passed through our tabular data models. The second path, will take the audio and convert it to a Mel-Spectrogram (melody spectrogram which takes the log of the frequencies so that they have a higher linear proximity to be easier to discern) [29], and then saved as a image file (JPG), which will be passed to the CNN. These scripts have been adapted from my previous work with Mel-Spectrograms and audio samples [30], so that we focused most on these models.

3. The CI methods

For these techniques one can use a variety of tools, however what we saw is that two main ones were used which were:

1. Artificial Neural Networks (ANNs)
2. Convolutional Neural Networks (CNNs)

For the Artificial Neural Networks (ANN) which are known to work well with tabular data [16,17], which can also be used for a variety of applications that range from sound analysis, geographical data[18] or any type of features that can be represented in a tabular form. A classical example of an ANN is shown below in Figure 1, with two Hidden Layers and three outputs.

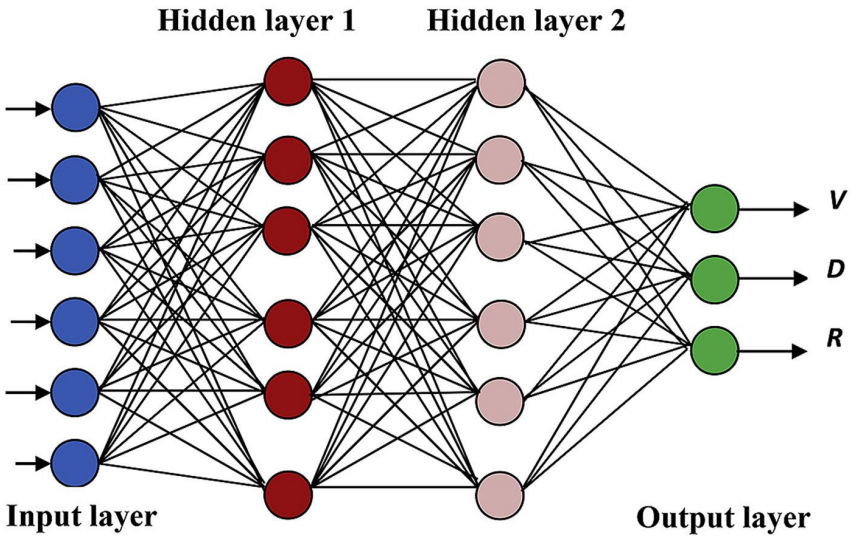

Figure 1. Sample Artificial Neural Network with 2 Hidden Layers and 3 outputs [19].

Additionally, there have also been promising developments from the area of Deep Learning where Convolutional Neural Networks (CNN) have been shown to be very flexible and accurate when working with images, whether it be for cancer classification, image classification, or cellular image analysis [12-14], with binary- or multi-label classification [15]. These models are great for image analysis as they can extract a great range of features from their convolutional layers [21]. A sample of a CNN is shown in Figure 2 below. Again this is not representative of the architecture that we have used.

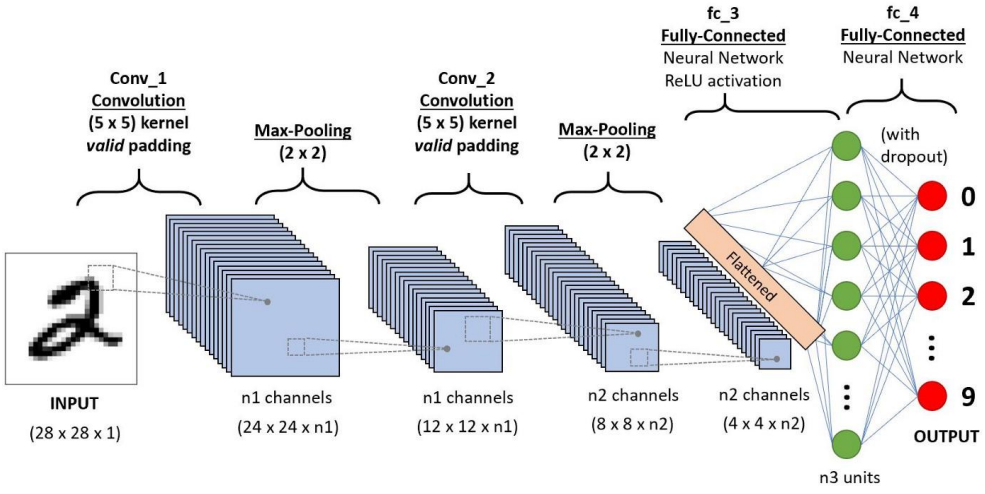

Figure 2. Sample Convolutional Neural Network (CNN) to identify handwritten digits[20]

Since now, there have been pretty dominant models in the area of image recognition such as ResNet50, famous for using a Residual Network [22] as well as ImageNet50 [23] which has achieved high scores for the majority of the tasks that it faced. While on the other hand for tabular data, there have been different models that have been useful such as TabNet, and Category Embedding Networks from pytorch tabular [25].

In order to be able to compare the TL and pytorch_tabular methods, we need to define our models' architectures. Since we are using two different types of inputs tabular and image, we need to make sure that our model accepts these formats. We define the two architectures for our ANN and CNN below, which one has been adapted from literature (CNN).

For our ANN we used a Fully Connected Neural Network (FCNN) with different number of connected layers, and found that using a similar structure to what can be seen below in Figure 3, worked best, which starts with 26 features and then boils them down to 16, 8, 4, and ultimately to two possible classes (positive or negative).

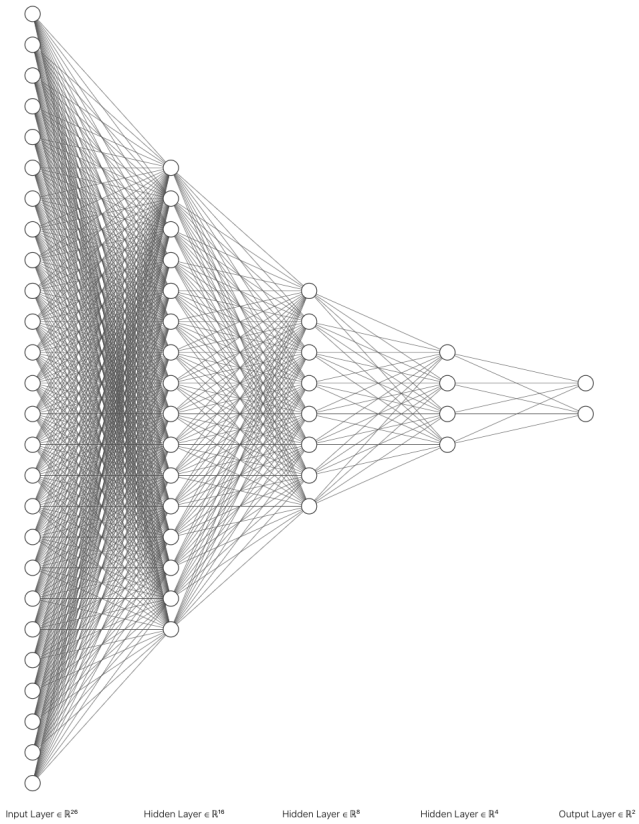

Figure 3. Architecture of the ANN used a Fully Connected Neural Network (FCNN). [27]

On the other hand for the CNN architecture, we used different models and it grew, and the most similar description of the one we are using is similar to the one in Figure 4 below. The CNN architecture uses common blocks of convolution layers with a max pool layer. This is then repeated with a change in the filters and then one can use a dropout or go directly to the Fully Connected (FC) layer.

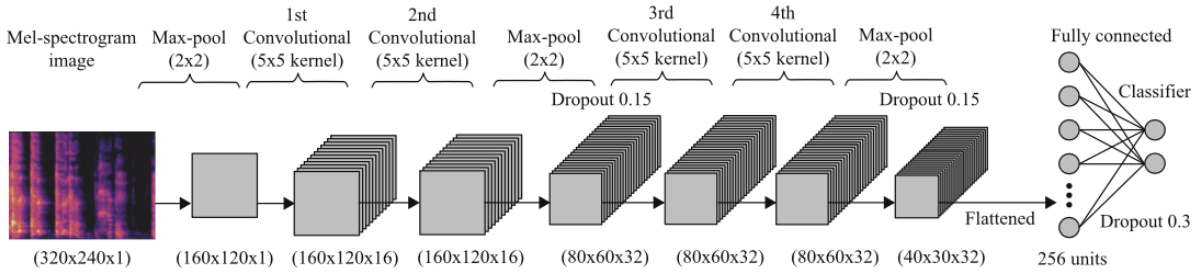

Figure 4. Architecture of the Convolutional Neural Network (CNN). The Mel-spectrogram comes into the network and there is a Max Pooling of (2,2) done, which reduces the image size by half. This is then passed through two convolution layers with 16 filters each and a kernel size of (5,5). Then passed through a Max Pooling layer (2,2) and a Dropout layer of 15%. This block is then repeated except that it will contain 32 filters in the convolution layers. This is then all flattened and passed on a Fully Connected layer with a binary output.

Once the architectures for the models were designed we used an appropriate framework, in this case, PyTorch, in order to build the models, in addition to that we also used a tracking tool to see the experiments [32], which is open to look at the experiments we ran. Once this was set up, we processed the audio samples into their respective forms, and used the models to predict. Since the tool for tracking uses the steps rather than epochs, we will be showing the steps of the models instead of the epochs.

4. Results and Discussion

During our experimentation we made about 4 runs for each of the models, and averaged them out in order to compare them. The data was split in a 70/30 for training and testing, and each of the models was trained for a maximum of 100 epochs. After every 20 epochs, we would perform a validation to check whether the model was performing well, this was placed in the same loop as the training.These can be seen below divided into their Training and Testing Accuracies. We saw that PyTorch CNE had the highest training accuracy with 90% which was followed by the Transfer Learning with ImageNet50, and in third place the Transfer Learning with ResNet50, with 89% and 88% respectively, this can be seen in Figure 5 below.

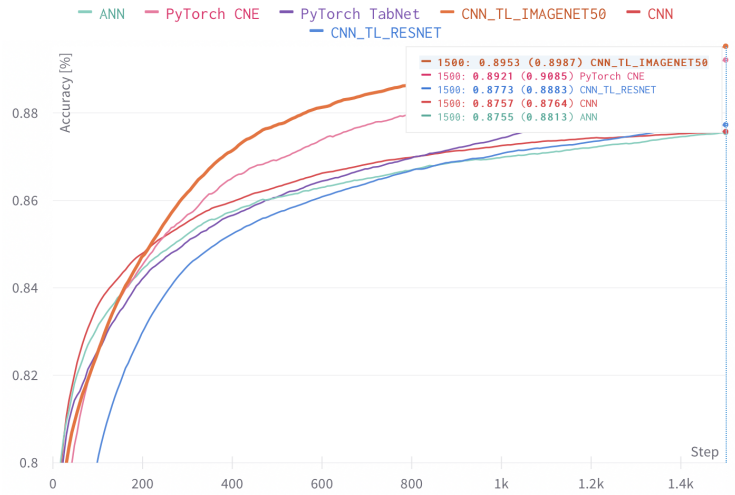

Figure 5. Training Accuracies of all the models compared. PyTorch CNE the highest of them all with ~ 90%.

However, during the testing we saw that the Transfer Learning with ImageNet50 performed the best with about 82% accuracy, and followed by the custom ANN (81%) and then followed by the PyTorch CNE with ~81%. This can be seen in Figure 6 below, again the TL showing strong capability to adapt to this custom dataset.

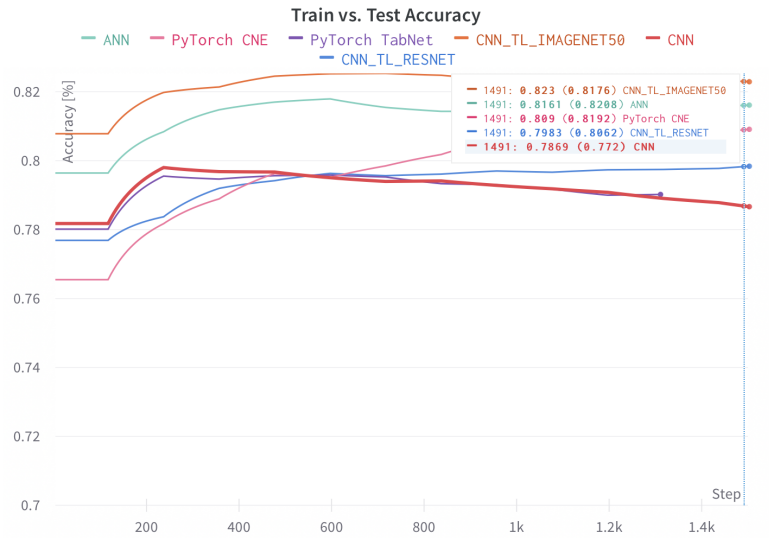

Figure 6. Testing Accuracies of all the models compared. CNN with TL from ImageNet50 the highest of them all with ~ 82%.

When we are looking at the loss of the models, we see that they are pretty similar, yet we saw that some are much faster at finding an optimum. This is done without hyperparameter optimization, albeit it could've been used with different tools. When looking at the training loss of the models, seen in Figure 7 below, we see that the PyTorch TabNet has the lowest loss, and outperforms the rest by finding a global optimum as efficiently as possible.

Figure 7. Training loss of all the models the lowest loss was seen by the PyTorch TabNet (~4.7) and then followed by the PyTorch CNE (~4.8) and closely by the CNN TL with ImageNet50 (~4.8).

When taking a look at the testing loss, we see that there is tendency that after a large amount of epochs (or steps in this case, which would be the number of times it goes through the dataloader) that it begins to increase. This is a note to keep in mind as we will discuss some of this in the next parts. We can see this behavior in Figure 8 below, as we can see that after the step number 600-800, we see an increase in the majority of the loss values for the different models.

We can also see that the tabular data models are using the same loss function which is a Mean Squared Error (MSE), with a SGD optimizer with a learning rate of 0.01, this has been working the best when coming to this type of data as they are the default value for most of the models [28]. On the other hand, the image datasets are using a different optimizer and loss function, for the CNN we are using a Binary Cross Entropy loss function, and for the optimizer we are using Adam, with a learning rate of 0.001, which worked best for us. The optimizer Adam has been shown to work efficiently with CNNs and image classification [28].

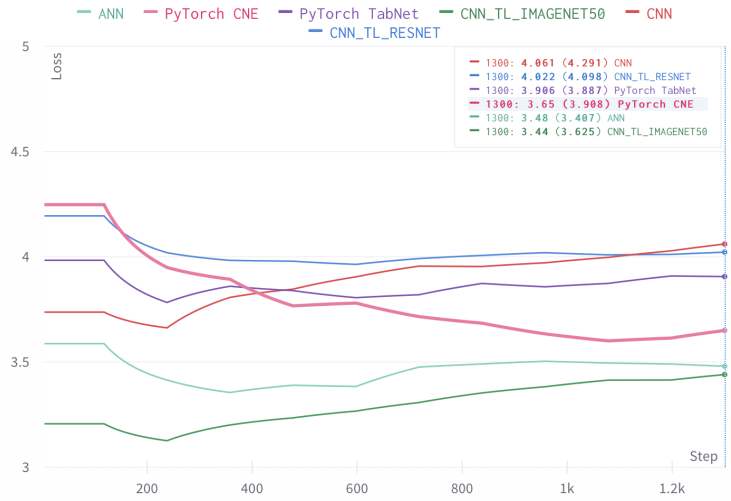

Figure 8. Testing loss of all the models, the lowest loss was seen by the CNN TL with ImageNet50 (~3.44) and then followed by the Custom ANN (~3.48) and lastly by the PyTorch CNE (~3.65).

What we saw is that the CNN with Transfer Learning with ImageNet50 had the highest accuracy with 82%, followed by our custom ANN and then pytorch_tabular's CNE with 81% and 81% respectively. We saw that using simple Deep Learning models we were able to predict with an average of 80% whether a given cough sample is positive or negative for COVID-19. However, this had many different difficulties when working with audio samples.

One of the issues that we encountered was the class-imbalance, out of the 2400 samples that were available from the Coswara-Dataset about 600 of them were positive and 1800 were negative samples. This is a major issue when working with Machine Learning or Deep Learning models. This issue could've been addressed by upsampling the minority class or downsampling the majority class. In our scenario, we trained the models with this class imbalance, which might be a reason why the models have a discrepancy in their training, testing accuracies and losses. **

On the other hand, for the features extracted from the audio samples using the librosa library, this could be a part where it would be necessary to go more in depth because there are features of audios which might be important when analyzing cough samples, which differ from speech. However, for our scenario, we extracted the main features that were used.

The weaknesses that this provides is that the models are trained with class-imbalance, but still manage to perform relatively well when it comes to the testing (and looking at their losses), as well as a possible "forgetting" of important features when doing the audio feature extraction. In addition to this, we also used 'vainilla' models, which means that they weren't optimized for the specific task, yet they were used with their best "raw" performance.

Albeit, this shows that we can have a strong model when comparing it to Transfer Learning styles with ImageNet50 or ResNet50 or SoTA tabular models such as CNE or TabNet, by making use of two

convolutional layers or the custom ANN that we detailed above. This is an insight to show that, when the components of the model are thought through then there is a possibility to outperform large networks [28]

5. Conclusions and future work

In conclusion, we want to state that using Artificial Neural Networks (ANN) and Convolutional Neural Networks (CNNs) work well when wanting to predict whether a person has COVID-19 based on a cough sample. Albeit, there needs to be more optimization in case of the hyperparameters, and the architecture that has been used. We saw that there were difficulties when working with the data and that might have been an issue at the time of analyzing the data, yet it was a good experience to work with such interesting models such as ANN's and CNNs. However, for future work we'd want to take a closer look at the functions of Convolutional Neural Networks (CNN) and how they can be best optimized to have a better performance for spectrogram tasks. The options that we might want to explore is to expand on the current CNN model that we built and optimize in order to make it more robust. In addition to this we'd also want to use more data to be able to make a model that would generalize better.

References

[1] Kirtipal N, Bharadwaj S, Kang SG. From SARS to SARS-CoV-2, insights on structure, pathogenicity and immunity aspects of pandemic human coronaviruses. Infect Genet Evol. 2020 Nov;85:104502. doi: 10.1016/j.meegid.2020.104502. Epub 2020 Aug 13. PMID: 32798769; PMCID: PMC7425554.

[2] Harrison AG, Lin T, Wang P. Mechanisms of SARS-CoV-2 Transmission and Pathogenesis. Trends Immunol. 2020 Dec;41(12):1100-1115. doi: 10.1016/j.it.2020.10.004. Epub 2020 Oct 14. PMID: 33132005; PMCID: PMC7556779.

[3]Lai CC, Shih TP, Ko WC, Tang HJ, Hsueh PR. Severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) and coronavirus disease-2019 (COVID-19): The epidemic and the challenges. Int J Antimicrob Agents. 2020 Mar;55(3):105924. doi: 10.1016/j.ijantimicag.2020.105924. Epub 2020 Feb 17. PMID: 32081636; PMCID: PMC7127800.

[4] Parrish, Colin R et al. "Cross-species virus transmission and the emergence of new epidemic diseases." *Microbiology and molecular biology reviews : MMBR* vol. 72,3 (2008): 457-70. doi:10.1128/MMBR.00004-08

[5] Rodriguez, Laura, "Entrevista a Carles Ventura". 2022. https://www.uoc.edu/portal/en/news/entrevistes/2022/001-carles-ventura.html

[6] https://www.covid-19-sounds.org/en/

[7]https://news.mit.edu/2020/covid-19-cough-cellphone-detection-1029

[8]Ali, Charles N. John, MD Iftikhar Hussain, Muhammad Nabeel, AI4COVID-19: AI enabled preliminary diagnosis for COVID-19 from cough samples via an app,Informatics in Medicine Unlocked,

[9] Zoabi Y, Deri-Rozov S, Shomron N. Machine learning-based prediction of COVID-19 diagnosis based on symptoms. NPJ Digit Med. 2021 Jan 4;4(1):3. doi: 10.1038/s41746-020-00372-6. PMID: 33398013; PMCID: PMC7782717.

[10] https://arxiv.org/pdf/2107.10716.pdf

[11] Alafif T, Tehame AM, Bajaba S, Barnawi A, Zia S. Machine and Deep Learning towards COVID-19 Diagnosis and Treatment: Survey, Challenges, and Future Directions. Int J Environ Res Public Health. 2021 Jan 27;18(3):1117. doi: 10.3390/ijerph18031117. PMID: 33513984; PMCID: PMC7908539.

[12Williams, Travis & Li, Robert. (2018). An Ensemble of Convolutional Neural Networks Using Wavelets for Image Classification. Journal of Software Engineering and Applications. 11. 69-88. 10.4236/jsea.2018.112004.

[13] Moen, E., Bannon, D., Kudo, T. *et al.* Deep learning for cellular image analysis. *Nat Methods* 16, 1233–1246 (2019). https://doi.org/10.1038/s41592-019-0403-1

[14] Hedyehzadeh M, Maghooli K, MomenGharibvand M, Pistorius S. A Comparison of the Efficiency of Using a Deep CNN Approach with Other Common Regression Methods for the Prediction of EGFR Expression in Glioblastoma Patients. J Digit Imaging. 2020 Apr;33(2):391-398. doi: 10.1007/s10278-019-00290-4. PMID: 31797142; PMCID: PMC7165204.

[15] https://arxiv.org/abs/1604.04573

[16]https://en.wikipedia.org/wiki/Artificial_neural_network

[17]https://arxiv.org/abs/2110.01889

[18] https://developers.arcgis.com/python/guide/ml-and-dl-on-tabular-data/

[19] M.A. Mojid, A.B.M.Z. Hossain, M.A. Ashraf, Artificial neural network model to predict transport parameters of reactive solutes from basic soil properties, Environmental Pollution, Volume 255, Part 2, 2019,

[20] https://miro.medium.com/max/1400/1*uAeANQIOQPqWZnnuH-VEyw.jpeg

[21]LeCun, Yann et al. “Object Recognition with Gradient-Based Learning.” *Shape, Contour and Grouping in Computer Vision* (1999).

[22] https://arxiv.org/abs/1512.03385

[23] https://ieeexplore.ieee.org/document/5206848

[24] https://arxiv.org/abs/1908.07442

[25] https://arxiv.org/pdf/2104.13638.pdf

[26]Imran A, Posokhova I, Qureshi HN, Masood U, Riaz MS, Ali K, John CN, Hussain MI, Nabeel M. AI4COVID-19: AI enabled preliminary diagnosis for COVID-19 from cough samples via an app. Inform Med Unlocked. 2020;20:100378. doi: 10.1016/j.imu.2020.100378. Epub 2020 Jun 26. PMID: 32839734; PMCID: PMC7318970.

[27] https://alexlenail.me/NN-SVG/

[28]https://www.fast.ai

[29]https://developer.apple.com/documentation/accelerate/computing_the_mel_spectrum_using_linear_algebra

[30] https://github.com/walzter/COVID_Cough

[31] https://github.com/iiscleap/Coswara-Data

[32] https://wandb.ai/walzter/ci_lab_covid/overview