TeliNet: Classifying CT scan images for COVID-19 diagnosis

Mohammad Nayeem Teli
Department of Computer Science
University of Maryland
College Park, MD 20742
nayeem@umd.edu

Abstract

COVID-19 has led to hundreds of millions of cases and millions of deaths worldwide since its onset. The fight against this pandemic is on-going on multiple fronts. While vaccinations are picking up speed, there are still billions of unvaccinated people. In this fight against the virus, diagnosis of the disease and isolation of the patients to prevent any spread play a huge role. Machine Learning approaches have assisted in the diagnosis of COVID-19 cases by analyzing chest X-rays and CT-scan images of patients. To push algorithm development and research in this direction of radiological diagnosis, a challenge to classify CT-scan series was organized in conjunction with ICCV, 2021. In this research we present a simple and shallow Convolutional Neural Network based approach, TeliNet, to classify these CT-scan images of COVID-19 patients presented as part of this competition. Our results outperform the F1 'macro' score of the competition benchmark and VGGNet approaches. Our proposed solution is also more lightweight in comparison to the other methods.

1. Introduction

On March 11, 2020, the World Health Organization(WHO) declared COVID-19, caused by a novel coronavirus, a global pandemic [5]. Soon after that the global lock downs and restrictions on travel brought life to a stand still as we know it. Since then there have been over 208 Million cases with over 4.3 Million deaths caused by this virus worldwide, as of the middle of August, 2021. In an effort to combat this virus our main defenses were isolation, quarantining, face masks and frequent washing of hands. Some of these approaches continue till date as we transition into in-person work and other activities.

In addition to these simple measures, frequent testing and tracing back helps identify and isolate spreaders and super spreaders of the virus. The real game changer in preventing hospitalizations and deaths in this yet to be over fight against this virus have been the vaccines. However, with so many cases, and slower rate of vaccinations the researchers have relied on many types of diagnostic techniques and tests to identify a COVID-19 positive patient.

There are three common types of tests that are carried out to identify a COVID positive patient. First type of testing, Reverse-Transcriptase Polymerase Chain Reaction (RT-PCR) testing, identifies the presence of actual viral genetic material by taking a nasal pharyngeal swab. The second type of testing is antigen testing. It detects the presence of one of the outer proteins of the viral shell or envelope. The third type of testing is to detect the presence of antibodies inside a person that identify the presence of the virus. The RT-PCR testing is the most popular and more accurate than antigen testing. However, it is more sensitive than antigen testing and needs to be done in a laboratory setting. As a result RT-PCR testing has a turnaround time of several days.

In addition to these tests it is a common practice to supplement diagnostic tests for detection and accurate prognosis of different diseases by applying machine learning approaches [24, 1, 6, 8, 20, 4, 19, 21]. The most commonly used imaging diagnostic tools include, X-ray imaging, Computed Tomography (CT), ultrasound imaging and Magnetic Resonance Imaging (MRI).

Since the beginning of the Coronavirus pandemic, many machine learning approaches have been applied to help clinicians detect and diagnose COVID-19. Many of these approaches specifically use images obtained from Chest Radiographs (CXR) and chest Computed Tomography(CT) [26]. In order to further push Machine Learning research for better analysis and classification of COVID-19 diagnostic image data, a workshop is organized in conjunction with International Conference on Computer Vision (ICCV) '21 conference. In this workshop a competition is organized to classify CT-scan image series into covid and non-covid classes [18]. This paper presents our work on this data set in response to this COVID-19 CT scan series classification challenge.

We introduce TeliNet, a very simple and lightweight but effective Convolutional Neural Network (CNN) based approach to classify chest CT-scan series images of patients as COVID-19 positive or not. We compare our results with VGGNet-16 [29] and the benchmark classification approach presented by Dimitrios et al. [18] on the same data set. The results indicate that our TeliNet method performs significantly better than both these approaches. Our F1 'macro' score on the validation data set is 0.81 in comparison to the benchmark score of 0.70 and a VGGNet-16 score of 0.72. Our method is also significantly faster than VGGNet-16 when run on the same hardware. The main contributions of this paper include, 1) TeliNet, a lightweight and simple but very effective CNN based approach to classify CT-scan series, and 2) low computational requirements. Our approach was run on a MacBook Pro with 2.3 Ghz dual core i5 processor with 8 GB of RAM. This makes it easily accessible to users who may not have access to high-end computational resources.

In the next few sections we present the related work, followed by the data set used in this research. Next, we discuss the methodology, including the architectural details of TeliNet and VGGNet. Finally, we present results followed by the conclusions.

2. Related work

Computer vision has been a very useful technique in assisting clinicians with diagnosis. Due to the advancement of pattern recognition and classification algorithms, computer aided diagnostic tools have greatly supplemented the ability to detect different diseases [24]. In order to classify the images obtained using X-rays, MRI and Computer Tomography, Convolution Neural Networks (CNN) have been the most popular tool [33]. Different types of CNN architectures have been used for image diagnosis since 2012, such as, AlexNet [22] [16], VGGNet [29], ResNet [11], InceptionNet [30], etc.

Bressem et al. [3] studied and compared 16 different CNN architectures on two datasets, CheXpert and COVID-19 image data collection. Bressem et al. concluded that it is not necessary to use complex architectures and that simpler CNN's perform great in classification on both data sets. Sahlol [27] combined Inception CNN as a feature extractor with Fractional-order Calculus - Marine Predators Algorithm (FO-MPA) to detect COVID-19 in images using the COVID-19 dataset available on Kaggle. Jia et al. [15] used a modified MobileNet and modified ResNet architectures to classify COVID-19 and non-COVID-19 CT scan images data set [31]. The results indicate almost perfect accuracy on that data set. Ibrahim et al. [12] compared four different architectures to classify chest X-ray and CT images to detect COVID-19, pneumonia and lung cancer using a single model. Many other Convolutional Neural Network based architectures [25, 2, 9, 14, 23, 10] have been similarly proposed since the onset of COVID-19 to help diagnose this disease and assist clinicians.

3. Data set

The data set used for the this research is part of "AI-enabled Medical Images Workshop and COVID-19 Diagnosis Competition (MIA-COV19D)", organized in conjunction with the International Conference on Computer Vision (ICCV) 2021 in Montreal, Canada, October 11- 17, 2021. The data is a part of a chest CT scan database manually annotated for COVID-19 / nonCOVID-19 diagnosis. The data set consists of 5000 CT scan series' with each series consisting of 50-700 2D-CT slices. Some of the representative slices of covid and non-covid CT scan slices are presented in Figure 1. Each of these slices is a 512×512 pixels 2D image. There are three different subsets that are provided, training, validation and test. The training set has 335672 images belonging to 2 classes and the validation data set has 75534 images belonging to 2 classes. The test set consists of 450 folders each representing a series containing a total of 609057 2D slices. For a detailed description of this data a curious reader is encouraged to read the benchmark paper by Dimitrios et al. [18]. This paper describes the competition data set and the benchmark approach applied on it in great detail.

4. Methods

In this section we present our CNN architecture, TeliNet, and compare its results with VGGNet-16 and the benchmark results. Both, TeliNet and VGGNet-16 approaches were run on the same hardware. The following sections describe our experiments using the two architectures and their details. However, we begin with data pre-processing, followed by TeliNet and VGGNet-16 architectures' description.

4.1. Preprocessing

In order to better organize the data for our experiments we use data generators. These data generator inputs are applied to both TeliNet as well as VGGNet architectures. For the benchmark approach results we rely on the experiments and the results provided in the benchmark paper by Dimitrios et al.

Data generators require that the data is organized in folders representing each class. Since we have two main classes, COVID-19 and non-COVID-19 the data is stored in two folders named as such. At the top level we have two main folders train and validation. Each of these two folders contain covid and non-covid folders representing the two classes for binary classification. All images belonging to their specific class were stored in their respective folders.

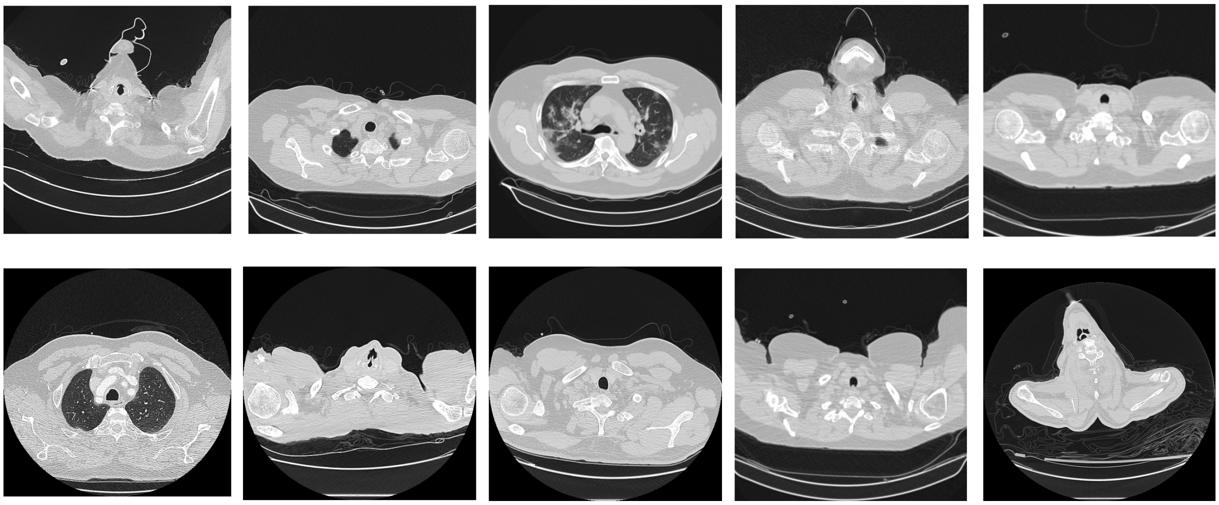


Figure 1. Samples of 2D CT scan slices. The top row represents covid slices and the bottom row, non-covid slices.

A separate test folder contains eight different subsets. Each of those subsets contains images for predictions with 450 folders, each representing a CT-scan series of test slices for which labels were not provided.

4.2. TeliNet

It is a four-layer shallow Convolutional Neural Network. It consists of 2D convolutions, batch normalizations, LeakyReLU activation functions and max pooling layers. This architecture is represented in Figure 2 with its corresponding summary in Figure 3.

As seen, it is a very shallow network with less than 8.5 Million trainable parameters. That is much smaller than over 134 Million trainable parameters in VGGNet-16. The input images are resized from the original 512×512 pixels to 256×256 images and re-scaled to values between 0 and 1. 2D convolution is carried out on these images using 16 filters of size, 3×3 followed by LeakyReLU activations. After the initial convolutions, there are 3, 2D convolutions layers, each followed by batch normalization, LeakyReLU activation functions and maxpooling. The padding is same in all layers, with a unit stride for each convolution. The max pooling for each layers consists of 2×2 pixels with a stride of 2.

After the convolution layers, two dense layers, each of 256 units is used, followed by batch normalization and LeakyReLU activation functions. Unlike the convolutions layers, we use a 10% dropout for this dense layer. Finally, we have a Sigmoid unit for binary classification as covid or non-covid.

Batch normalization smoothes the optimization problem by normalizing the inputs to each layer. This helps fight the covariant shift problem [13] and also enables a quicker and more stable training [28] of the data set using TeliNet. It leads to faster convergence and takes advantage of a wider range of learning rates. To further benefit from these features we use a learning rate scheduler that reduces the learning rate by a factor of 70% with each epoch. This values was chosen after a series of pilot experiments. The advantages of using LeakyReLU as an activation function over other activation functions are well known [32]. It also facilitates a faster convergence, complimenting the speed-up due to fewer trainable parameters and batch normalizations.

4.3. VGGNet

Simonyan and Zisserman [29] presented the VGGNet architecture for a large scale image recognition. This work presented the use of the small 3×3 convolution filters that helped extend the depth of the deep convolution network to 16-19 network layers. Using this architecture Simonyan et al. were able to secure both the first and the second positions in the ImageNet [7] challenge of 2014.

For this research we used VGGNet-16 that consists of 16 layers and over 134 Million trainable parameters. The architectural summary is presented in Figure 4. Each layer of convolutions involves filters of sizes 3×3, ReLU activation function,and padding to retain the size with a stride of 1. The maxpooling in this architecture is a 2×2 filter with a stride of 2. The simplicity of this approach is the uniformity of the filter sizes, padding and maxpooling across all layers.

The input to this architecture is a 224×224 pixel image convolved with 64 filters. Same convolutions are repeated in the next layer. This is followed by maxpooling and two layers of convolutions using 128 filters and maxpooling. Next, there are three convolutions using 256 filters and maxpooling. It is followed by 3 convolutions with 512 filters and maxpooling. The next two layers are fully connected layers with 4096 units each. The final layer is a

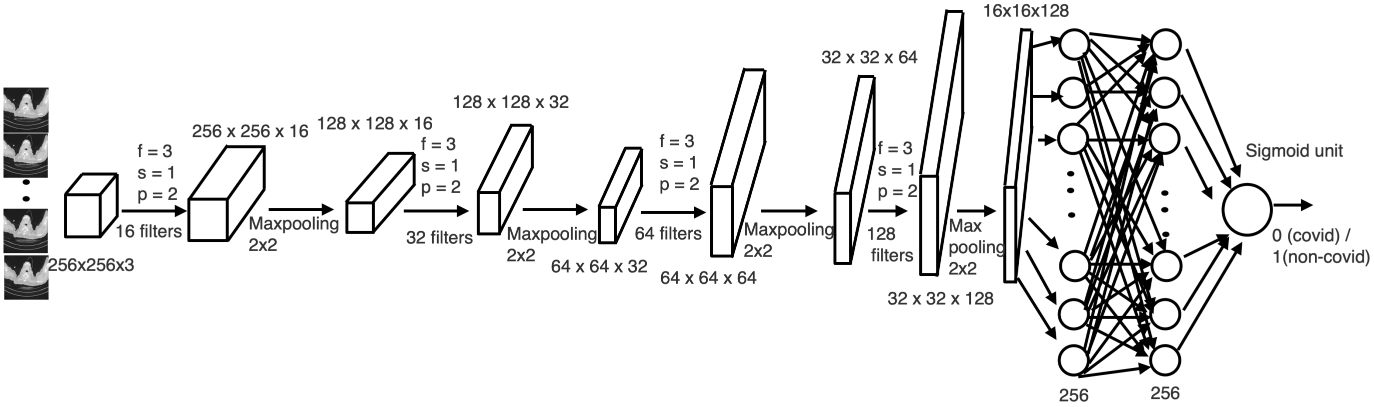


Figure 2. TeliNet architecture.

Method	F1 'macro' score
benchmark [18]	0.7
VGGNet-16	0.72
TeliNet(ours)	**0.81**

Table 1. Validation set F1 score results using the three approaches.

Sigmoid unit for the binary classifcation into covid and non-covid images. For a detailed description of the VGGNet-16 architecture, please refer to the original work by Simonyan and Zisserman [29].

5. Results

We present a comparison of the results of TeliNet, VGGNet-16 and the benchmark approaches. In the training set there are 153681 covid slices and 181991 non-covid slices. The validation set contains 35016 covid slices and 40517 non-covid slices. Since the data sets are not balanced we use an F1 'macro' score metric to measure the efficacy of our approach just like the one used in the Benchmark paper [18]. These results are shown in Table 1.

The results of TeliNet and VGGNet-16 were obtained on the same hardware, comprising of a MacBook Pro, with a 2.3 GHz Dual-Core Intel Core i5 processor and 8 Gb of RAM. Both these approaches were trained using the RMSprop optimizer, an adaptive learning method, proposed by Geoffrey Hinton, after empircally comparing with the Adam optimizer [17]. We also tried experiments using different sized batches ranging from 4 to 128 and finally settled on 32 for the final results due to better performance.

Since it is a binary classification problem, we used binary cross-entropy loss function to train these networks. Accuracy, precision and recall metrics were computed. Precision and recall were used to determine the F1 score by taking their harmonic mean.

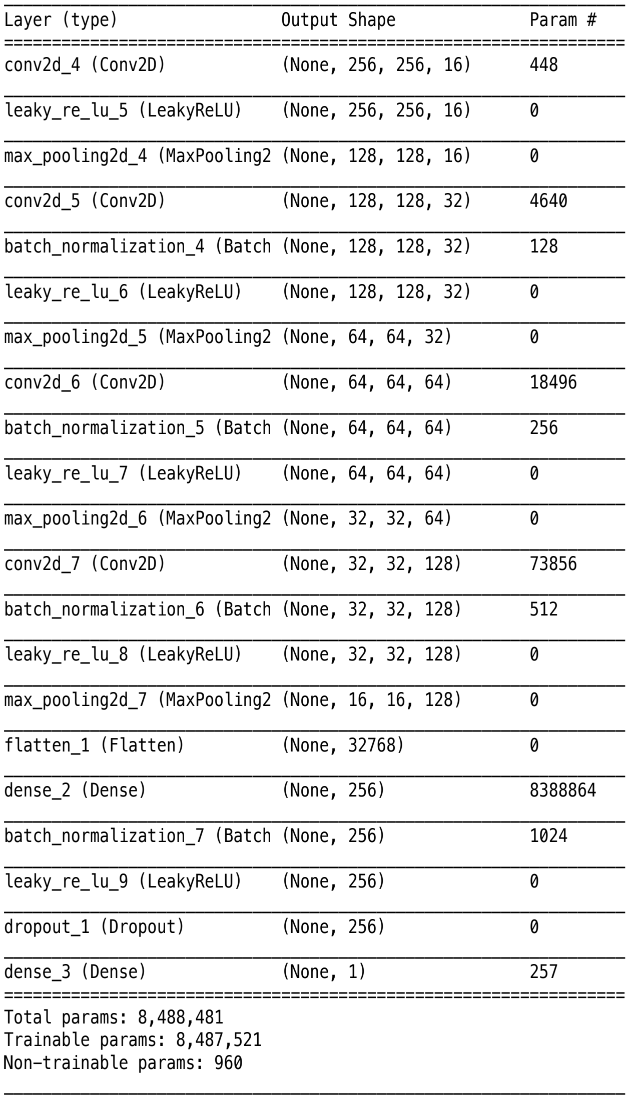
Even though our solution has done better than the benchmark method and the VGGNet-16 architecture, there is a plenty of room for improvement. Our F1 'macro' score on validation data set is 81% in comparison to 72% and 71% on the same data set by VGGNet-16 and the benchmark approach. On the test set TeliNet gives an F1 'macro' score of 70.86% in comparison to 67% for the benchmark score. The labels for the test set were not given to us and therefore these results are based on the outcomes of the experiments performed on the test sets by the competition organizing team.

It is important to note that these results are for 3D CT-scan series'. While it may seem that the reported results on the validation and the test sets have been reported on 2D slices, it is in fact a 3D series of these slices. Each of these series' provides a sequences of 2D slices. Our results were obtained on these series' with the label assigned based on the classification outcomes on these slices. Although our goal was to use a majority voting to determine the label for a series in these slices, it was seen that majority of these slices were classified to be belonging to a single class for each series anyway and so it was a very straightforward decision.

6. Conclusion

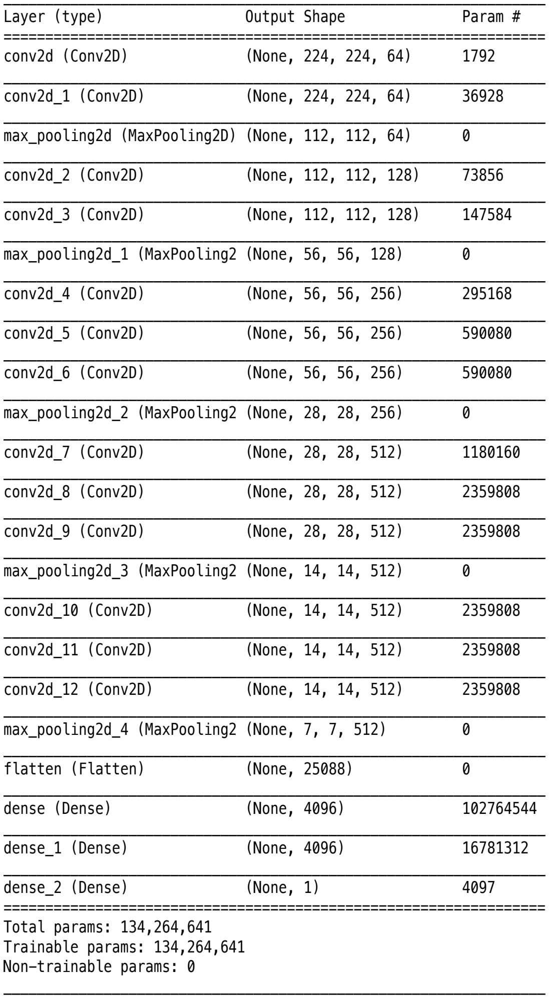
In this research we introduce a shallow and lightweight but very effective CNN architecture, TeliNet, for binary classification of CT-scan series' of COVID-19 and non-COVID19 slices. We compare our results with VGGNet-16 and the benchmark results on this difficult data set. The results indicate that our approach does significantly better than the other two methods on the validation data set and better than the benchmark approach on the test set. The results on the test set using the VGGNet have not been evaluated since the test set labels were not provided to us.

Our contributions include, 1) a very simple CNN architecture trained to do better than more complex architectures,



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_4 (Conv2D)            (None, 256, 256, 16)      448
_________________________________________________________________
leaky_re_lu_5 (LeakyReLU)    (None, 256, 256, 16)      0
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 128, 128, 16)      0
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 128, 128, 32)      4640
_________________________________________________________________
batch_normalization_4 (Batch (None, 128, 128, 32)      128
_________________________________________________________________
leaky_re_lu_6 (LeakyReLU)    (None, 128, 128, 32)      0
_________________________________________________________________
max_pooling2d_5 (MaxPooling2 (None, 64, 64, 32)        0
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 64, 64, 64)        18496
_________________________________________________________________
batch_normalization_5 (Batch (None, 64, 64, 64)        256
_________________________________________________________________
leaky_re_lu_7 (LeakyReLU)    (None, 64, 64, 64)        0
_________________________________________________________________
max_pooling2d_6 (MaxPooling2 (None, 32, 32, 64)        0
_________________________________________________________________
conv2d_7 (Conv2D)            (None, 32, 32, 128)       73856
_________________________________________________________________
batch_normalization_6 (Batch (None, 32, 32, 128)       512
_________________________________________________________________
leaky_re_lu_8 (LeakyReLU)    (None, 32, 32, 128)       0
_________________________________________________________________
max_pooling2d_7 (MaxPooling2 (None, 16, 16, 128)       0
_________________________________________________________________
flatten_1 (Flatten)          (None, 32768)             0
_________________________________________________________________
dense_2 (Dense)              (None, 256)               8388864
_________________________________________________________________
batch_normalization_7 (Batch (None, 256)               1024
_________________________________________________________________
leaky_re_lu_9 (LeakyReLU)    (None, 256)               0
_________________________________________________________________
dropout_1 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 257
=================================================================
Total params: 8,488,481
Trainable params: 8,487,521
Non-trainable params: 960
_________________________________________________________________
```

Figure 3. Description of the TeliNet architecture with Convolution layers and the tunable parameters.



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 224, 224, 64)      1792
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 224, 224, 64)      36928
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 112, 112, 64)      0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 112, 112, 128)     73856
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 112, 112, 128)     147584
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 56, 56, 128)       0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 56, 56, 256)       295168
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 56, 56, 256)       590080
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 56, 56, 256)       590080
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 28, 28, 256)       0
_________________________________________________________________
conv2d_7 (Conv2D)            (None, 28, 28, 512)       1180160
_________________________________________________________________
conv2d_8 (Conv2D)            (None, 28, 28, 512)       2359808
_________________________________________________________________
conv2d_9 (Conv2D)            (None, 28, 28, 512)       2359808
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 14, 14, 512)       0
_________________________________________________________________
conv2d_10 (Conv2D)           (None, 14, 14, 512)       2359808
_________________________________________________________________
conv2d_11 (Conv2D)           (None, 14, 14, 512)       2359808
_________________________________________________________________
conv2d_12 (Conv2D)           (None, 14, 14, 512)       2359808
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 7, 7, 512)         0
_________________________________________________________________
flatten (Flatten)            (None, 25088)             0
_________________________________________________________________
dense (Dense)                (None, 4096)              102764544
_________________________________________________________________
dense_1 (Dense)              (None, 4096)              16781312
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 4097
=================================================================
Total params: 134,264,641
Trainable params: 134,264,641
Non-trainable params: 0
_________________________________________________________________
```

Figure 4. Description of the VGGNet-16 architecture with Convolution layers and the tunable parameters.

and 2) a lightweight architecture that can be run on a personal computer. These advantages make this architecture accessible to clinicians and researchers who may not always have a high end hardware.

However, this paper would not be complete without discussing the pitfalls of the Machine Learning research conducted in the diagnosis using chest X-rays and CT scan images. Roberts et al. [26] conducted a review of 62 published papers after a thorough screening of 2,212 studies conducted between Jan. 1, 2020 to October 3, 2020 that presented new Machine Learning models for the diagnosis or prognosis of COVID-19 from chest X-ray and CT images. In each and every paper they found either methodological flaws and/or underlying biases and lack of reproducibility which render these proposed approaches useless of any potential clinical use. Keepng this in mind it is worthwhile to know that we have presented not only detailed architectures and the methodologies applied, we also make our code available on Github[1] for easy reproducibility. The dataset has been validated and the results of the competition have been uploaded to a Leaderboard for easy evaluation of the approaches and the validation of the results.

[1] https://github.com/nayeemmz/TeliNet

References

[1] An overview of deep learning in medical imaging focusing on mri. *Zeitschrift für Medizinische Physik*, 29(2):102–127, 2019. Special Issue: Deep Learning in Medical Physics. 1

[2] Nur-A Alam, Mominul Ahsan, Md. Abdul Based, Julfikar Haider, and Marcin Kowalski. Covid-19 detection from chest x-ray images using feature fusion and deep learning. *Sensors*, 21(4), 2021. 2

[3] Keno K. Bressem, Lisa C.Adams, ZChristoph Erxleben, Bernd Hamm, Stefan M. Niehues, and Janis L.Vahldiek. Comparing diferent deep learning architectures for classifcation of chest radiographs. *Scientific Reports (Nature Research)*, 10(13590), Aug. 12, 2020. 2

[4] Lawrence Carin and Michael J. Pencina. On Deep Learning for Medical Image Analysis. *JAMA*, 320(11):1192–1193, 09 2018. 1

[5] Domenico Cucinotta and Maurizio Vanelli. WHO declares covid-19 a pandemic. *Acta Biomed*, 91(1):157–160, March 19, 2020. 1

[6] Marleen de Bruijne. Machine learning approaches in medical image analysis: From detection to diagnosis. *Medical Image Analysis*, 33:94–97, 2016. 1

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 3

[8] Bradley J. Erickson, Panagiotis Korfiatis, Zeynettin Akkus, and Timothy L. Kline. Machine learning for medical imaging. *RadioGraphics*, 37:505–515, Feb. 17, 2017. 1

[9] Mustafa Ghaderzadeh, Farkhondeh Asadi, Ramezan Jafari, Davood Bashash, Hassan Abolghasemi, and Mehrad Aria. Deep convolutional neural network–based computer-aided detection system for covid-19 using multiple lung scans: Design and implementation study. *J Med Internet Res*, 23(4):e27468, Apr 2021. 2

[10] Ghulam Gilanie, Usama Ijaz Bajwa, Mustansar Mahmood Waraich, Mutyyba Asghar, Rehana Kousar, Adnan Kashif, Rabab Shereen Aslam, Muhammad Mohsin Qasim, and Hamza Rafique. Coronavirus (covid-19) detection from chest radiology images using convolutional neural networks. *Biomed Signal Process Control*, 66(102490), Feb. 10, 2021. 2

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2

[12] Dina M. Ibrahim, Nada M. Elshennawy, and Amany M. Sarhan. Deep-chest: Multi-classification deep learning model for diagnosing covid-19, pneumonia, and lung cancer chest diseases. *Computers in Biology and Medicine*, 132:104348, 2021. 2

[13] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 448–456. JMLR.org, 2015. 3

[14] Ronaldus Morgan James, Kusrini, and M. Rudyanto Arief. Classification of x-ray covid-19 image using convolutional neural network. In *2020 2nd International Conference on Cybernetics and Intelligent System (ICORIS)*, pages 1–6, 2020. 2

[15] Guangyu Jia, Hak-Keung Lam, and Yujia Xu. Classification of covid-19 chest x-ray and ct images using a type of dynamic cnn modification method. *Comput Biol Med.*, 134(104425), April 29, 2021. 2

[16] Sameer Khan and Suet-Peng Yong. A deep learning architecture for classifying medical images of anatomy object. In *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1661–1668, 2017. 2

[17] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 4

[18] Dimitrios Kollias, Anastasios Arsenos, Levon Soukissian, and Stefanos Kollias. Mia-cov19d: Covid-19 detection through 3-d chest ct image analysis. *arXiv preprint arXiv:2106.07524*, 2021. 1, 2, 4

[19] Dimitrios Kollias, N Bouas, Y Vlaxos, V Brillakis, M Seferis, Ilianna Kollia, Levon Sukissian, James Wingate, and S Kollias. Deep transparent prediction through latent representation analysis. *arXiv preprint arXiv:2009.07044*, 2020. 1

[20] Dimitrios Kollias, Athanasios Tagaris, Andreas Stafylopatis, Stefanos Kollias, and Georgios Tagaris. Deep neural architectures for prediction in healthcare. *Complex & Intelligent Systems*, 4(2):119–131, 2018. 1

[21] Dimitris Kollias, Y Vlaxos, M Seferis, Ilianna Kollia, Levon Sukissian, James Wingate, and Stefanos D Kollias. Transparent adaptation in deep medical image diagnosis. In *TAILOR*, pages 251–267, 2020. 1

[22] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. 2

[23] Shuang Liang, Huixiang Liu, Yu Gu, Xiuhua Guo, Hongjun Li, Li Li, Zhiyuan Wu, Mengyang Liu, and Lixin Tao. Fast automated detection of covid-19 from medical images using convolutional neural networks. *Communications Biology*, 4(35), Jan. 4, 2021. 2

[24] Xiaoqing Liu, Kunlun Gao, Bo Liu, Chengwei Pan, Kongming Liang, Lifeng Yan, Jiechao Ma, Fujin He, Shu Zhang, Siyuan Pan, and Yizhou Yu. Advances in deep learning-based medical image analysis. In *Health Data Science*, page 14 pages, June 16, 2021. 1, 2

[25] Caio B. S. Maior, João M. M. Santana, Isis D. Lins, and Márcio J. C. Moura. Convolutional neural network model based on radiological images to support covid-19 diagnosis: Evaluating database biases. *PLoS ONE*, 16(3), March 1, 2021. 2

[26] Michael Roberts, Derek Driggs , Matthew Thorpe, Julian Gilbey, Michael Yeung, Stephan Ursprung, Angelica I. Aviles-Rivero, Christian Etmann, Cathal McCague, Lucian Beer, Jonathan R. Weir-McCall , Zhongzhao Teng, Effrossyni Gkrania-Klotsas, AIX-COVNET, James H. F. Rudd, Evis Sala, and Carola-Bibiane Schönlieb. Common pitfalls and recommendations for using machine learning to detect and prognosticate for covid-19 using chest radiographs and CT scans. *Nature Machine Intelligence*, 3:199–217, March 15, 2021. 1, 5

[27] Ahmed T. Sahlol, Dalia Yousri, Ahmed A. Ewees, Mohammed A. A. Al-qaness, Robertas Damasevicius, and Mohamed Abd Elaziz. Covid-19 image classifcation using deep features and fractional-order marine predators algorithm. *Scientific Reports (Nature Research)*, 10(1364), Sep. 21, 2020. 2

[28] Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization? In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. 3

[29] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015. 2, 3, 4

[30] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Computer Vision and Pattern Recognition (CVPR)*, 2015. 2

[31] Linda Wang, Zhong Qiu Lin, and Alexander Wong. Covid-net: a tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images. *Scientific Reports (Nature Research)*, 10(19549), Nov. 11, 2020. 2

[32] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *CoRR*, abs/1505.00853, 2015. 3

[33] Samir S. Yadav and Shivajirao M. Jadhav. Deep convolutional neural network based medical image classification for disease diagnosis. *Journal of Big Data*, 6(113), December 17, 2019. 2