An Effective Machine Learning Solution to Detect COVID-19, Viral Pneumonia, and Lung Opacity

Joshua Green (250912612), Thakshanth Uthayakumar (251158422),
Zeyad Abdelmageid (250898290), Menaka Kahawalage (251235196),
Rashinda Wijethunga Mudiyanselage (251261477)

***Abstract*—2019 Novel Corona virus 19 (COVID-19), has affected the whole world by having an impact on economic sectors, social norms, daily life and loosing loved ones. In order to defend humanity against this viral disease, an efficient, less costly, less time consuming COVID-19 detection system is required. On the other hand, there are other similar lung diseases such as Viral Pneumonia, and Lung Opacity which affects the lungs in a similar way as of COVID-19 which makes the diagnosis of COVID-19 through lung examination more difficult. As a result additional tests are required to accurately diagnose COVID-19. However, these additional tests are costly, and time consuming. This urgent complex requirement motivates us to develop an sub-optimal machine learning solution to predict COVID-19, Viral Pneumonia, Lung Opacity disease in a patient using only the CXR images. To solve this problem, this paper presents two different approaches, namely convolutional neural network (CNN), and Transferred learning (TL) based CNN. Through simulation results we have shown that a lightweight TL approach such as MobileNetV2, delivers a 'better' accuracy with lower complexity.**

***Index Terms*—COVID-19, convolutional neural network, transfer learning**

I. Introduction

2019 Novel Corona virus also known as COVID-19 caused by severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) virus which has spread to every nook, and corner of our planet is big threat to the mere existence of human lives. The firstly detected virus in 2019 is less powerful, less contagious, less deadly. However, so far 5 different mutated version of this virus have been found, and with each mutation, the virus becomes more deadly, contagious, and more resilient towards the current vaccines. Therefore, an efficient mass scale vaccination, and quick, accurate COVID-19 detection techniques are required to defend our lives against this deadly virus.

The COVID-19 firstly affects the lung in a human body, and it creates the lung condition similar to Viral Pneumonia, and Lung Opacity diseases. It makes very difficult for a medical professional to diagnose COVID-19 in a patient using only the CXR image. Therefore, in current practise, additional tests such as molecular polymerase chain reaction (PCR), rapid antigen tests are performed to detect COVID-19 in a patient accurately. Even though those tests can detect COVID-19 accurately, they are time consuming, costly, and they increase medical waste globally due to their one-time use only nature. This complex, and urgent problem motivates us to design an effective machine learning solution to detect COVID-19, Viral Pneumonia, Lung Opacity disease in a patient using only the CXR images. Such machine learning solution will be less complex, less time consuming, accurate, environment friendly with minimal cost.

A. Related Works

From literature, there are three major techniques that successfully employ CNNs to medical image classification problems. First approach is to model, and train CNNs from sctrach. In [1] a Multi-crop Convolutional Neural Network (MC-CNN) model has been built 'from scratch' to classify lung nodule malignancy. Second approach is to implement "off-the-shelf CNN" features (without retraining the CNN) and applying them directly to a problem of similar domain. This has been proposed in [2] for chest X-rays in [2] and for CT lung nodule identification in [3]. Third approach is to perform an unsupervised pre-training on natural or medical images and fine-tune it on to medical target images using CNN or other types of deep learning models. In [4], [5], the authors have investigated and evaluated a semi-supervised learning approach using data from multiple domains .

In many scenarios, collecting sufficient training data can be costly in terms of money and time, and in some cases may be unrealistic. Transfer learning (TL) solves this problem by applying knowledge from one domain to another. TL takes the knowledge (also known as, weights, network structure) to solve a problem and applies it to solve a a new problem. In [6], the authors have compared VGG16, VGG19 and DenseNet201 models with a large data set and achieved an accuracy of 99.62% on the binary classification and 95.48% on the multi-class classification.

B. Contribution

- We investigate CNN, Tuned-CNN, and TL approaches to detect COVID-19, Viral Pneumonia, and Lung Opacity using CXR images.
- We propose the best sub-optimal model to solve this problem in terms of accuracy, and training complexity
- We present an explainability approach to convey our proposed solution to health care workers

II. System Model

To solve the aforementioned problem, six neural network models such as CNN, Tuned-CNN, MobileNetV2, VGG-16, VGG-19, and DenseNET201 were implemented. Their

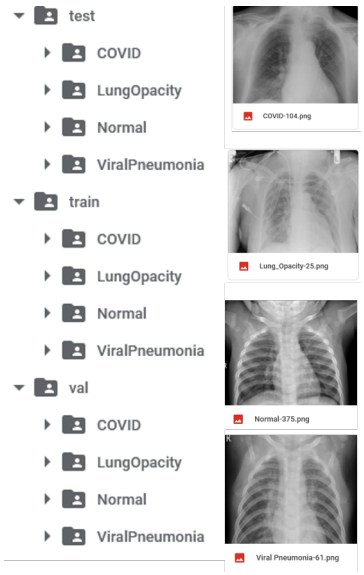
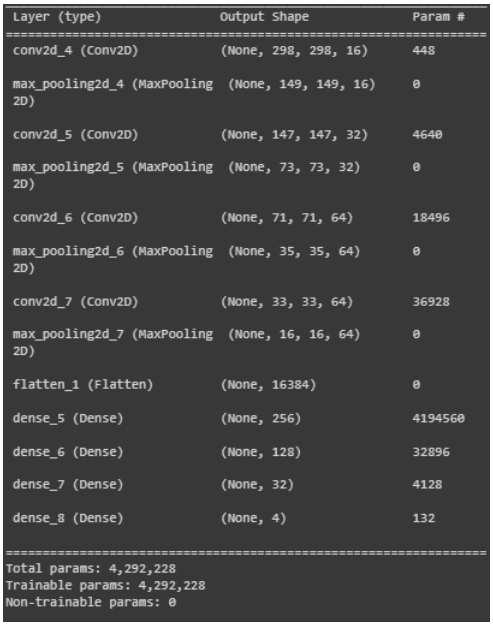


Fig. 1. Data set preview



Layer (type)	Output Shape	Param #
conv2d_4 (Conv2D)	(None, 298, 298, 16)	448
max_pooling2d_4 (MaxPooling 2D)	(None, 149, 149, 16)	0
conv2d_5 (Conv2D)	(None, 147, 147, 32)	4640
max_pooling2d_5 (MaxPooling 2D)	(None, 73, 73, 32)	0
conv2d_6 (Conv2D)	(None, 71, 71, 64)	18496
max_pooling2d_6 (MaxPooling 2D)	(None, 35, 35, 64)	0
conv2d_7 (Conv2D)	(None, 33, 33, 64)	36928
max_pooling2d_7 (MaxPooling 2D)	(None, 16, 16, 64)	0
flatten_1 (Flatten)	(None, 16384)	0
dense_5 (Dense)	(None, 256)	4194560
dense_6 (Dense)	(None, 128)	32896
dense_7 (Dense)	(None, 32)	4128
dense_8 (Dense)	(None, 4)	132

Total params: 4,292,228
Trainable params: 4,292,228
Non-trainable params: 0

Fig. 2. CNN architecture

performance, and complexity were analysed to propose the best neural network (NN) model for this prediction problem.

A. Data set Analysis and preparation

We used a dataset found at [7] for this project. It consists of Chest X-Ray (CXR) images of 3616 COVID-19 positive patients, 10,192 normal people, 6012 lung opacity patients, 1345 viral pneumonia patients. All the images were in grey scale with a resolution of 299×299 pixels.

We selected 1000 CXR images for each class, to prevent the class imbalance problem, faster model tuning, and training. Then, the selected images were split into train, validation, and test sets with the ratio of $0.7 : 0.15 : 0.15$. During the data set analysis, we notices discrepancies in CXR images such as manually printed labels on the images, distortions due to imperfect scans, and capture of irrelevant body parts (E.g chin, neck). Therefore, we selected only well scanned images to ensure fair training, testing process. The preview of the data set is illustrated in Fig. 1.

B. CNN

CNN is a type of neural network model which incorporates convolutional, and pooling layers on top of the hidden layers of a neural network. The filters in convolutional layers are used to extract higher order features from an image. CNN models are well suited for object detection, and image classification applications. The pooling layers are embedded in-between 2 convolutional layers to reduce the image size from a convolutional layer to another, such that only the required features in the image could be activated for along the training process.

In our system model, we design the CNN model with an input layer, 4 convolution layers, 4 pooling layers, 3 dense layers, and an output layer as illustrated in Fig. 2

C. Tuned CNN

Then, we tuned the hyper parameters of the CNN hyper parameters to optimize the performance for this prediction problem.

In general, grid search method, and random search method are used for hyper-parameter tuning. In grid search method, all the combinations of all hyper parameters are considered and tried out (similar to a brute force approach). In this approach, the number of trials depends on the total number of possible combinations of the hyper parameters. In random search, the total number of trials to be tried out are specified at the beginning of tuning process. In each trial, a unique combination of hyper parameters are considered (like a trial and error method), and the best hyper parameter combination which delivers the highest accuracy is proposed as the best model after the tuning.

Grid search based tuning would require a minimum $4 \times 5 \times 7 = 140$ trials for our system. Due to the limited computational resources , the maximum number of trials we were able to run was 93. Therefore, in our system, the most crucial hyper parameters for image processing listed as follows were tuned using random search method with 93 trials.

- Number of filters : Decides the number of feature maps generated (32 to 256 , step size = 32)
- Number of convolutional layers : Learns the features detected (5 to 9, step size = 1)
- Filter size : Reduces the image dimensionality of the image between CNN layers $(3, 3), (4, 4), (5, 5), (6, 6)$

Due to limited computational resources, the other hyper parameters such as number of dense layers, learning rate were not tuned in this project.

D. Transferred Learning (TL)

In addition to the CNN approach, we implemented TL approaches to solve this problem. TL method is essentially applying a 'Learned' (pre-weighted and pre-made architecture) across different overlapping domains. For our problem, four TL models, namely: MobileNetV2, VGG16, VGG19, and DenseNET201; were implemented to propose the best sub-optimal solution for our problem.

1) Data Pre-processing: The data pre-processing for the TL models were performed differently due to the TL models being trained on RGB images from Imagenet, with an image size of (224,224,3) tuple. Since our CXR dataset contains only a single channel (grey-scale images), we stacked each gray-scale image to a 3-channel 'pseudo-color' image to mimic the red-green-blue (RGB) structure of natural images. However, the stacked 3-channel gray-scale image does not contain any color information [8]. Therefore, the data had to be reshaped from (299,299,1) to (224,224,1) and then converted to a 'pseudo-color' RGB image to get to (224,224,3). Then, the image data and the classification markings were made. Since our dataset is labeled as 'Normal', 'COVID-19', 'Viral Pneumonia', and 'Lung Opacity', we assigned class labels to training and test data as $0, 1, 2, 3$ respectively. The images and classes were then converted into an array, with their respective labels. The images were then scaled by a factor of $\frac{1}{255}$, and all images and labels were concatenated into 2 separate variables. The concatenation was performed to ensure that the labels and images corresponded with each other.

2) Implementation: The MobileNetV2 model was implemented by importing the pre-trained model of MobileNetV2 without top layer from [9]. Then, a sequential model was built by adding fully connected trainable layers on top of the frozen layers. As found in [10], a fully-connected layer of neuron size 512 with 'relu' activation function after the feature extractor model was implemented with a dropout layer of value 0.2 embedded in between the convolutional layers to avoid overfitting. Then, the final output layer with 4 neurons and 'Softmax' activation function was employed. After model compilation, the model was trained while keeping the transferred layers frozen. Training was regulated with an early stopping mechanism, and a validation split of 0.2 was used. The validation split holds a fraction of the training data to be used as validation data. The model will not learn from validation data, however, it will be used to evaluate the model performance at the end of each epoch [11]. Once the model was trained, it was then used to predict the class for each test data image. The implemented MobileNetV2 model architecture is shown in Fig. 3.

Similarly, the VGG16, VGG19, and DenseNET201 models were built, trained, tested, and evaluated. However, they all differed from the MobileNetV2 construction in the case they were not built as a sequential class object. Rather a keras model class object was created for each TL model, which requires an input stage, and an output stage structure and creates a full model. The input stage is what contains the pre-trained weights and architecture of the TL model. The output stage contains an average pooling with a pool size of 4×4, a layer of flattening, and finally like the the MobileNetV2 output, a fully-connected layer of neuron size 512 with 'relu' activation, and a 4 neuron classification output layer with a 'softmax' activation function to conclude the architecture.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 keras_layer (KerasLayer)    (None, 1280)              2257984

 dense (Dense)               (None, 512)               655872

 dropout (Dropout)           (None, 512)               0

 dense_1 (Dense)             (None, 4)                 2052

=================================================================
Total params: 2,915,908
Trainable params: 657,924
Non-trainable params: 2,257,984
_________________________________________________________________
```

Fig. 3. MobileNet Architecture

E. Interpretability

One of the main issues with ML models is interpretability which means explaining how the model makes decisions. It is extremely important for two main reasons. Firstly, for the data scientist to make sure that the model is making decisions in a way that coincides with the goal of the model. For example, in our case we want to make sure that the model does the diagnosis based on the actual condition of the lungs, not based on the rib cage size for example which could be due to a bias in the dataset. In our data set, if the majority of COVID scans are for male patients and on average male patients have larger rib cages than female patients, the model could just be making this diagnosis based on the rib cage size which is unacceptable as the virus does not impact the rib cage size. Secondly, for our problem, the doctors need to see what the model looks at to make the decision. Such understanding gains trust on the model. Our deep learning model has a tough interpretability due to the complexity of the model. To tackle this issue, we used an interpretability algorithm for CNN known as Gradient Classification Activation Maps (Grad CAM) [12], [13].

III. Results and Analysis

The CNN, Tuned-CNN, and TL models are compared in this section. We used sparse categorical accuracy (SCA) metric, and sparse categorical cross entropy loss function. To compare the computational-time complexity, the train wall time which considers the time taken to complete training in reference to an external timer was used. The confusion matrix presented in Fig. 4 provides a further insight into the model's performance. We used it to determine accuracy, precision and F1 score. We can also observe that in Fig. 2,3, the parameters trained in CNN are $4,292,228$, and in MobileNetV2 are $657,924$ which makes the MobileNetV2 stand out from from CNN model

trained from scratch in terms of training complexity. We can see in the Fig. 5, the a COVID x-ray scan, which was predicted correctly by our model, and the heat map which corresponds to parts of the image based on which the model makes the decision. The feature maps activated along the convolutional, and pooling layers is illustrated in Fig. 19. The accuracy, and training loss of all the models are illustrated in Fig. 6 - 15. It is clear from Table I, and Fig. 18 that MobileNetV2 based TL model delivered the highest accuracy, with lower number of parameters to be trained, lower training loss, lower train time.

TABLE I. TL performance table

TL model	Train Wall Time	Loss	SCA
CNN	40min 23s	0.3557	0.9231
Tuned CNN	205min 32s	0.2828	0.9329
MobileNetV2	**2min 57s**	**0.2242**	**0.9353**
VGG16	13min 9s	0.3197	0.8647
VGG19	10min 44s	0.3846	0.8529
DenseNet201	9min 11s	0.3251	0.9015

```
[[155   1   2   0]
 [  4 156   0   9]
 [ 12   2 154   2]
 [  1  13   0 169]]
              precision    recall  f1-score   support

           0       0.90      0.98      0.94       158
           1       0.91      0.92      0.91       169
           2       0.99      0.91      0.94       170
           3       0.94      0.92      0.93       183

    accuracy                           0.93       680
   macro avg       0.93      0.93      0.93       680
weighted avg       0.93      0.93      0.93       680
```

Fig. 4. MobileNet Confusion Matrix and Classification report

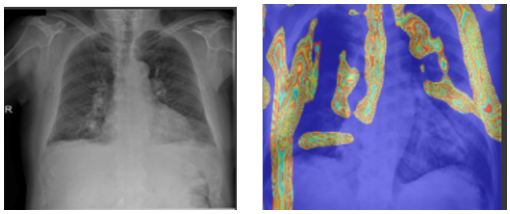


Fig. 5. GRAD CAM interpretation

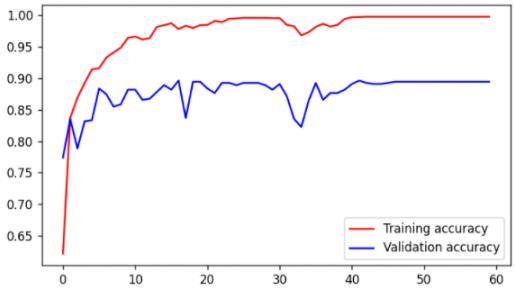


Fig. 6. CNN Accuracy

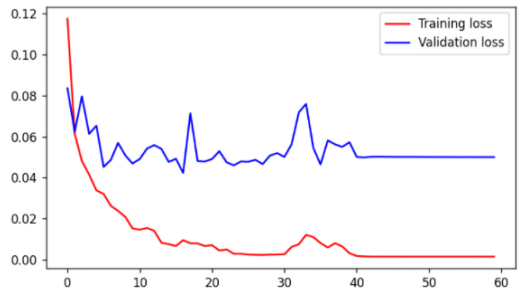


Fig. 7. CNN Loss

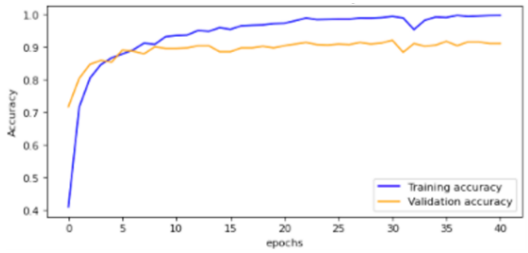


Fig. 8. Tuned CNN Accuracy

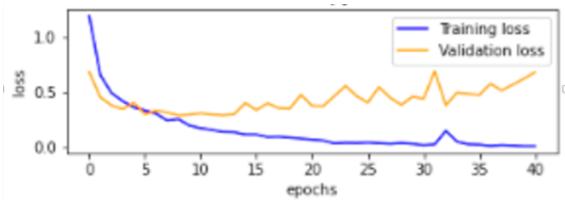


Fig. 9. Tuned CNN Loss

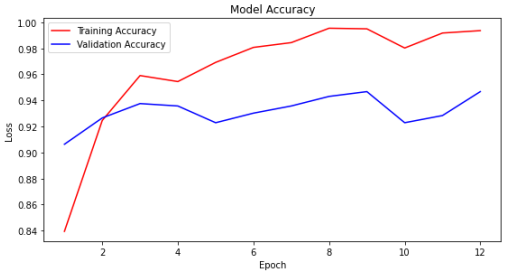


Fig. 10. MobileNet Accuracy

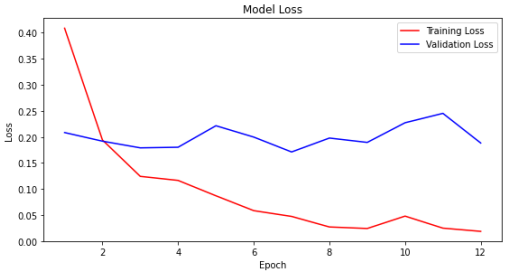


Fig. 11. MobileNet Loss

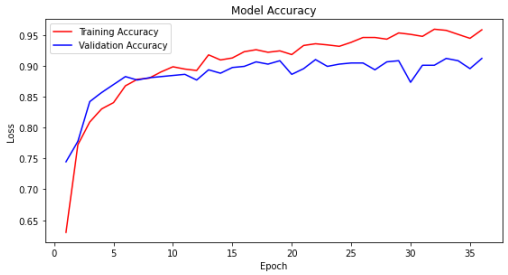


Fig. 12. VGG16 Accuracy

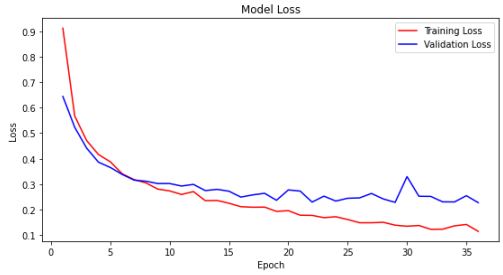


Fig. 13. VGG16 Loss

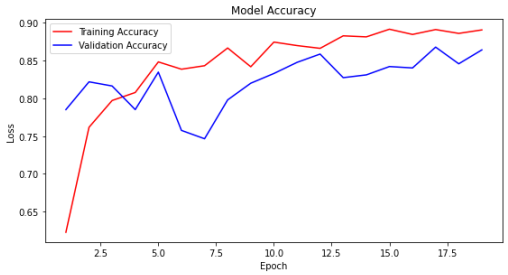


Fig. 14. VGG19 Accuracy

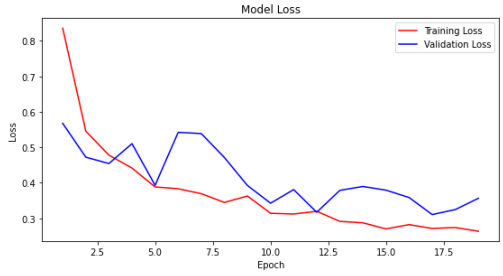


Fig. 15. VGG19 Loss

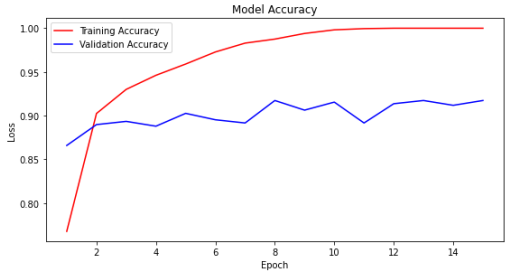


Fig. 16. DenseNet201 Accuracy

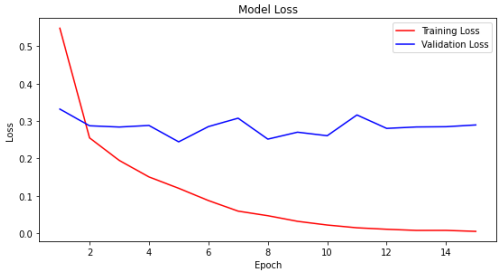


Fig. 17. DenseNet201 Loss

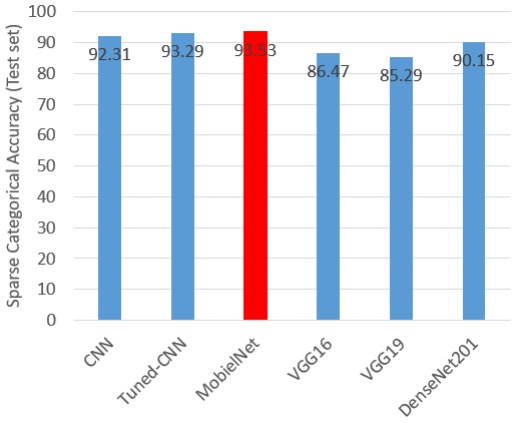


Fig. 18. Test Accuracy comparison

IV. Discussion and Conclusion

In this project, various machine learning approaches such as CNN, Tuned-CNN, TL based MobileNetV2, VGG-16, VGG-19, and DenseNET201 were implemented to solve this problem. From the results analysis, we proposed that TL approach using MobileNetV2 is well suited for this application as it delivers highest accuracy with lowest training complexity. We have also presented an approach to convey, and interpret the proposed machine learning solution to people with non-machine learning background as an effort to gain their trust on the proposed solution. The presented interpretability approach takes machine learning applications one step closer to the health care industry. As future work, the proposed model could be tuned further to predict complicated disease conditions such as COVID-19 induced Lung Opacity, and COVID-19 induced Pneumonia.

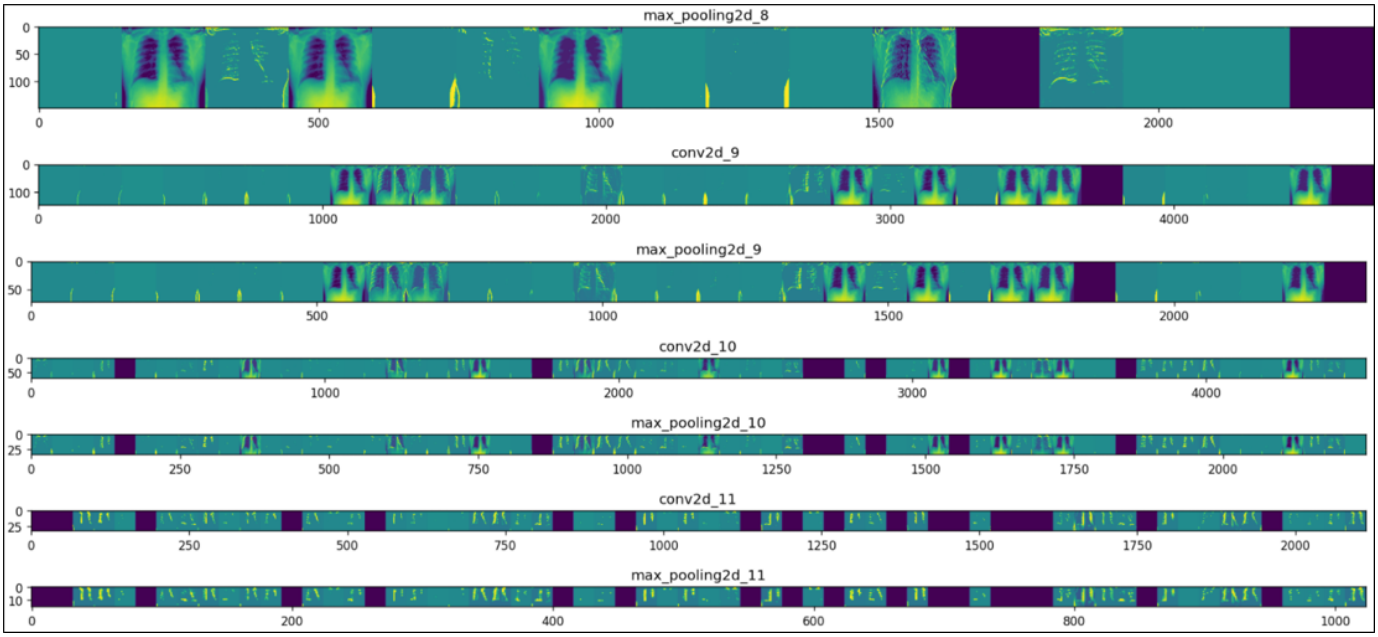


Fig. 19. Activation Map through convolution, and pooling layers

V. References

[1] W. Shen, M. Zhou, F. Yang, C. Yang, and J. Tian, "Multi-scale convolutional neural networks for lung nodule classification," in *Information Processing in Medical Imaging*, S. Ourselin, D. C. Alexander, C.-F. Westin, and M. J. Cardoso, Eds., Cham: Springer International Publishing, 2015, pp. 588–599, ISBN: 978-3-319-19992-4.

[2] Y. Bar, I. Diamant, L. Wolf, S. Lieberman, E. Konen, and H. Greenspan, "Chest pathology detection using deep learning with non-medical training," in *2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI)*, 2015, pp. 294–297. DOI: 10.1109/ISBI.2015.7163871.

[3] B. van Ginneken, A. A. A. Setio, C. Jacobs, and F. Ciompi, "Off-the-shelf convolutional neural network features for pulmonary nodule detection in computed tomography scans," in *2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI)*, 2015, pp. 286–289. DOI: 10.1109/ISBI.2015.7163869.

[4] T. Schlegl, J. Ofner, and G. Langs, "Unsupervised pre-training across image domains improves lung tissue classification," in *MCV*, 2014.

[5] F. Zhuang, Z. Qi, K. Duan, *et al.*, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2021. DOI: 10.1109/JPROC.2020.3004555.

[6] A. Badawi and K. Elgazzar, "Detecting coronavirus from chest x-rays using transfer learning," *COVID*, vol. 1, no. 1, pp. 403–415, 2021, ISSN: 2673-8112. DOI: 10 . 3390 / covid1010034. [Online]. Available: https://www.mdpi.com/2673-8112/1/1/34.

[7] T. Rahman. "Covid-19 radiography database." (2020), [Online]. Available: https : / / www . kaggle . com / tawsifurrahman / covid19 - radiography-database.

[8] Y. Xie and D. Richmond, "Pre-training on grayscale imagenet improves medical image classification," in *Computer Vision – ECCV 2018 Workshops*, L. Leal-Taixé and S. Roth, Eds., Cham: Springer International Publishing, 2019, pp. 476–484, ISBN: 978-3-030-11024-6.

[9] Google. (2021), [Online]. Available: https : //tfhub.dev/google/tf2-preview/mobilenet_v2/feature_vector/4 (visited on 11/28/2021).

[10] M. M. Taresh, N. Zhu, T. A. A. Ali, A. S. Hameed, and M. L. Mutar, "Transfer learning to detect covid-19 automatically from x-ray images using convolutional neural networks," *International journal of biomedical imaging*, vol. 2021, p. 8 828 404, 2021, ISSN: 1687-4188. DOI: 10 . 1155 / 2021 / 8828404. [Online]. Available: https://europepmc.org/articles/PMC8203406.

[11] (2021), [Online]. Available: https : / / keras . rstudio . com / reference / fit . html (visited on 11/28/2021).

[12] D. Reiff. "Understand your algorithm with grad-cam - towards data science." (Jul. 2021), [Online]. Available: https://towardsdatascience.com/understand-your-algorithm-with-grad-cam-d3b62fce353.

[13] wiqaaas. "Gradient visualization." (Jul. 2020), [Online]. Available: https://github.com/wiqaaas/youtube/blob/master/Deep_Learning_Using_Tensorflow / Demystifying_CNN / Gradient%20Visualization.ipynb.