CT DIAGNOSIS OF COVID-19: COMPARISON OF TRANSFER LEARNING AND A BASIC CNN MODEL

Adar Bayan[*1], Özgün Ozan Nacitarhan[*2],

[*]*Computer Engineering Department, Koc University*
Istanbul,Turkey

[1]abayan17@ku.edu.tr

[2]onacitarhan17@ku.edu.tr

I. INTRODUCTION

Coronavirus Disease (COVID-19) is an infectious disease that was identified on 31 December 2019 in Wuhan City, China. The virus is transmitted through viral particles that contain the novel Coronavirus, and the number of cases for the Coronavirus is spreading dramatically. It has already infected more than 173 million and caused the death of 3.71 million people. In order to detect the patients and quarantine them before they infect others, the detection of the Coronavirus is crucial. For detecting Coronavirus, traditional molecular and antigen tests are used to determine whether a patient has an active infection or not. These traditional methods can be inefficient and paucity. An effective screening method for screening Covid-19 is required. For this reason, the use of chest Computed Tomography (CT) has proven that screening and detecting coronavirus is accurate. The analysis of a CT image can be complex and time consuming. In order to eliminate these, we propose a method Convolutional Neural Network (CNN) based classification framework for detecting COVID-19 using the CT images. Since the goal consists of binary classification, the proposed method classifies each CT image into positive COVID-19 or negative COVID-19. We have proposed two models, one is using transfer learning, and the second one is building from scratch. For implementing the proposed methods, we have used an open-source dataset with train, test, and validation sets[1] . The dataset has 764 CT images containing clinical findings of both non-COVID-19 and COVID-19 from 592 patients, and it's prepared by a senior radiologist in Tongji Hospital, Wuhan, China, who has performed diagnosis and treatment of a large number of COVID-19 patients [2].

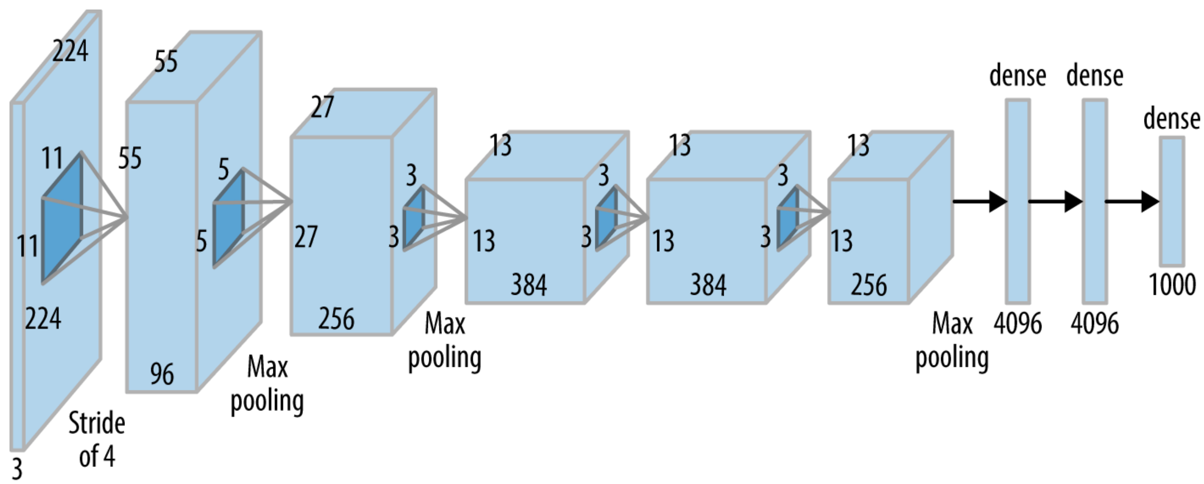

Figure 1: AlexNet Architecture

Source: Adapted from [3]

II. TRANSFER LEARNING MODEL WITH ALEXNET

A **Convolutional Neural Network** (**CNN**) is a special type of multi-player Neural Network that specializes in image classification. The CNN consists of pooling layers and convolution layers occurring in an alternating fashion [4]. The structure that has the layers is called CNN architecture. The proposed method that we have mentioned in the introduction part was created by the pretrained AlexNet network. AlexNet is a convolutional neural network (CNN) architecture that consists of 3 max-pooling layers, 2 normalization-layers, 5 convolutional layers, and 2 fully connected layers that can be seen in Figure 1. Since AlexNet accepts the input images in the proper format, we have applied some of the preprocessing methods such as resizing and cropping in the affirmed format before applying the AlexNet. To get better results we have normalized the image. For normalization, we have subtracted the AlexNets' mean from the current value of the image and divided this by AlexNets' standard deviation. AlexNet is designed for the 1000-class classification task, and our goal is to find a binary classification. In order to make the classification binary, we have modified the last fully connected layers of the network. After the modification part, we freeze the weights of the first layers, so that we do not have to retrain the model entirely. Since the CNN is not a goal of the network, we have used a cross-entropy loss function to evaluate how well our algorithm models work in our dataset. The cross-entropy loss function works well for the classification problems because it minimizes the distance between two probability distributions. For removing inaccuracy and noise, we have used an optimizer by restructuring datasets. Selected optimizer for the model called Stochastic Gradient Descent (SGD). It is an iterative algorithm for unconstrained optimization problems. To accelerate the model to learn faster, we have used a learning rate scheduler. Afterward the desired format folder structure of the dataset was changed for our training model method. The format accepts image datasets as inputs, and it contains train, test, and validation images.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_6 (Conv2D)            (None, 254, 254, 32)      896
_________________________________________________________________
max_pooling2d_6 (MaxPooling2 (None, 127, 127, 32)      0
_________________________________________________________________
conv2d_7 (Conv2D)            (None, 125, 125, 32)      9248
_________________________________________________________________
max_pooling2d_7 (MaxPooling2 (None, 62, 62, 32)        0
_________________________________________________________________
conv2d_8 (Conv2D)            (None, 60, 60, 64)        18496
_________________________________________________________________
max_pooling2d_8 (MaxPooling2 (None, 30, 30, 64)        0
_________________________________________________________________
flatten_2 (Flatten)          (None, 57600)             0
_________________________________________________________________
dense_4 (Dense)              (None, 64)                3686464
_________________________________________________________________
dropout_2 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_5 (Dense)              (None, 1)                 65
=================================================================
Total params: 3,715,169
Trainable params: 3,715,169
Non-trainable params: 0
_________________________________________________________________
```

Figure 2

III. CNN BINARY IMAGE CLASSIFIER

To test our model's success rate, we need to check the accuracy values which are crucial in image classification. Our first model was pretrained AlexNet network by CNN architecture, and the accuracy values are shown in the above section. For comparing our results with a different model, we proposed a new model. This model is trained from the scratch with TensorFlow, and it is different from the first model in a way of not having transfer learning. Since we want to see the difference between the first and our new model, we have applied some of the preprocessing methods. First, all images such as train, test, and validation are rescaled. After rescaling, we have changed the size, batch, and class mode. The preprocessing methods help us to compare the two models in the proper format and increase the model's accuracy. Our new CNN model has a convolutional 2D layer with 32 filters, a kernel size of 3x3, input size of an image 256x256, and the activation function of the (rectified linear function) 'ReLU'. Since we are dealing with binary classification, ReLu is a piecewise function that will output if it is positive, otherwise vice versa. After adding a convolutional 2D layer, we have added a max pooling layer that halves the dimension of the image. The desired output changed to 128x128. To get better results, these layers operations repeated respectively two more times but the last one has 64 filters instead of 32. Finally, our model has a flatten layer, a dense layer with 64 neurons, a dropout layer, and a final dense layer for binary output. The summary of our model can be seen in Figure 2. Since CNN is not a goal of the network, we have used a binary-cross-entropy loss function for binary classification to evaluate how well our algorithm models work in our dataset. For removing inaccuracy and noise, we have used an optimizer by restructuring datasets. Selected optimizer for the model called 'Adam'. It is a replacement optimization method for deep learning models. Then, we trained the models with 15 epochs which is the number of times that the model sees the dataset.

	Transfer Learning	CNN Classifier
Covid	0.826086956	0.713523543
Non-Covid	0.818181818	0.629857384
Overall	0.821428571	0.666071403

Table 1

IV. CONCLUSION

By understanding the crucial role of CT scans in detecting COVID-19, we have applied a transfer learning technique and a CNN that is built from scratch. As it can be shown in table 1, the transfer learning accuracy results range from 81% to 82%, the new model that we build ranges from 62% to 71%. The transfer learning accuracy results are higher than our version. There are several reasons for this. The first one is, AlexNet trained more than 1.2 million high resolution images, and these images are trained with a powerful GPU which takes more than 5 days. Another reason is that the architecture of AlexNet is much more complex than our new model. Therefore, the model that is built with the transfer learning technique gets higher accuracy results for detecting COVID-19 from the CT scans.

REFERENCES

[1] Covid-CT Grand Challenge,Grand Challenge, Mar.2020. Accessed on: June.6, 2021. [Online]. Available:https://covid-ct.grand-challenge.org/CT-diagnosis-of-COVID-19/

[2] Covid-CT Grand Challenge DataSet,Github, Mar.2020. Accessed on: June.6, 2021. [Online]. Available: HTTPS://GITHUB.COM/UCSD-AI4H/COVID-CT

[3] A. KRIZHEVSKY, I. SUTSKEVER AND G. E. HINTON, "IMAGENET CLASSIFICATION WITH DEEP CONVOLUTIONAL NEURAL NETWORKS", *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS*, VOL. 2, PP. 1097-1105, 2012

[4] M. Ammar, B. Mohammed, A. Senjian, S. Ferdous, B. Farid, H. Renae, K.Gary,,R. K.Gary, (2017). "Handbook of Neural Computation." Academic Press.pp. 383-401.Available: https://www.sciencedirect.com/science/article/pii/B9780128113189000211