PERFORMANCE TESTING OF COVID-19 IMAGE CLASSIFICATION USING DIFFERENT MACHINE LEARNING ARCHITECTURES

A THESIS SUBMITTED TO
THE FACULTY OF ARCHITECTURE AND ENGINEERING
OF
EPOKA UNIVERSITY

BY

ENRIKO SMOQI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

MARCH, 2023

Approval sheet of the Thesis

This is to certify that we have read this thesis entitled **"Performance Testing Of Covid-19 Image Classification Using Different Machine Learning Architectures"** and that in our opinion it is fully adequate, in scope and quality, as a thesis of Master of Science.

____

Dr. Arban Uka

Head of Department

Date: March, 09, 2023

Examining Committee Members:

Dr. Arban Uka	(Computer Engineering)	_______________________________
Assoc. Prof. Dr. Carlo Ciulla	(Computer Engineering)	_______________________________
Dr. Florenc Skuka	(Computer Engineering)	_______________________________

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name Surname: Enriko Smoqi

Signature: _______________

ABSTRACT

PERFORMANCE TESTING OF COVID-19 IMAGE CLASSIFICATION USING DIFFERENT MACHINE LEARNING ARCHITECTURES

Smoqi, Enriko

M. Sc., Department of Computer Engineering

Supervisor: Dr. Arban Uka

The outbreak of the Covid-19 pandemic brought the need for research in this field to assist in diagnosing patients more accurately and with more speed than is possible using just human resources. Deep Learning and especially Convolutional Neural Network has shown great accuracy levels in medical field tasks such as image analysis, segmentation and classification. In this research we will review several deep learning models and focus on finding the CNN architectures which achieve highest classification accuracy results. Different datasets will be used for training and testing with the purpose of classifying CT and X-Ray scans into Covid-19 and No Covid-19 to give a correct diagnosis.

Keywords: *Chest X-ray, CT-scan, image preprocessing, classification, convolutional neural networks, machine learning*

ABSTRAKT

TESTIM PERFORMANCE TE KLASIFIKIMIT TE IMAZHEVE ME COVID-19 DUKE PERDORUR ARKITEKTURA TE NDRYSHME TE MESIMIT AUTOMATIK

Smoqi, Enriko

Master Shkencor, Departamenti i Inxhinierisë Kompjuterike.

Udheheqesi: Dr. Arban Uka

Me perhapjen e pandemise lindi nevoja per kerkim shkencor qe te ndihmoje ne diagnozat me te shpejta dhe me te sakta te pacienteve pa pasur nevoje per pune te specializuar stafi mjekesor. Mesimi automatik dhe specifikisht rrjetat neurale kane demostruar saktesi te larte ne fusha te ndryshme si psh klasifikimi i imazheve.Ne kete kerkim shkencor do dokumentojme dhe permbledhim teknika te ndryshme mesimi automatik dhe do perpiqemi te nxjerrim tekniken me efikase ne klasifikim.Grupe te ndryshme imazhesh do perdoren per trajnim dhe testim modeli me qellimin qe te klasifikohen imazhe skanesh dhe radiografish te mushkerive ne dy grupe Covid dhe jo Covid per nje diagnoze te sakte.

Fjalët kyçe*: radiografi mushkerish,skaner mushkerish, procesim imazhesh, klasifikim, rrjetat neurale, mesimi automatik*

ACKNOWLEDGEMENTS

I would like to thank my supervisor Dr. Arban Uka, for his help and dedication with his advices throughout the research of the thesis. His experience and motivation have been really helpful and inspiring this whole journey of writing the research.

TABLE OF CONTENTS

Table of Contents

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Covid-19 image classification

The COVID-19 detection on medical images such as X-rays and CTs has been a field of major importance these last two years. Considering the 2%-5% mortality rate and the high number of deaths which is 12.32 M confirmed cases, this zoonotic disease remains in the focus of many Deep Learning studies. Deep Learning has already proved its high accuracy levels in other medical research such as Coronary Artery Disease, Malaria, Alzheimer's disease, different dental diseases, and Parkinson's disease which proves that it has enormous potential to be of great help in COVID-19 detection as well. Jain, Mittal D, Thakur, Mittal MK [14] COVID-19 tests are both cost and time expensive and require a huge amount of work when done for a large number of patients. This work could be reduced and simplified using a deep neural network which can detect if a patient is COVID-19 positive or not.

1.2 Thesis Objective

The goal is to have a deep learning model which needs a small number of resources and little time to detect COVID-19 with high accuracy. During this period of time there has been much research done to achieve the above-mentioned goal. In this research we will review previous research done and try to accomplish having a deep learning model which will bring improvement in classification of COVID-19 images in accuracy, time and cost. If the task of predicting positive cases of Covid-19 early can be completed successfully, there would be a reduction of the pandemic spread along with a decrease in economic losses. (Degadwala, Vyas and Dave [3])

1.3 Organization of Thesis

The thesis is mainly organized in chapters which are listed below with an addition of what each includes. Apart from that there are also several navigation facilities such as a table of contents, list of figures, references and the appendix for the code used.

- Chapter 1 introduces general information about the COVID-19 detection and classification field and states the objectives of this thesis.
- Chapter 2 focuses on explaining in detail important knowledge needed to simplify reading this thesis, previous research done in this field of study and challenges encountered during this work along with possible solutions.
- Chapter 3 presents the methodology followed during this research including the dataset information, the image analysis and preprocessing that is used and finally the architectures of the deep learning models used.
- Chapter 4 reports the training and testing results for the models and comparison of the results achieved.
- Chapter 5 finalizes the conclusions of the work and presents future work ideas.

CHAPTER 2

2.LITERATURE REVIEW

2.1 Deep Learning

Zhang, Bamakan, Qu and Li [22] Deep learning came as a result of people trying to mimic the neural networks of the human brain. It is different from machine learning which is efficient in simple regression problems and has limitations in complicated problems due to their shallow architectures that often conduct only linear transformation.

Deep learning has gotten a lot of attraction lately and is being used in many different fields [5],[8], [11],[22]. The main fields are speech recognition, object detection and classification . Its architecture is mainly constructed by stacking multiple layers of simple modules. Each module conducts non-linear operations from data fed from the previous layer. DL relies heavily on good hardware and a high amount of data. The hardware will heavily affect the time it takes for it to train the model and the amount of data will affect how many features it will extract. Most used deep learning architecture are:

-Convolution Neural Network (CNN)

-Recurrent Neural Network (RNN)

-Restricted Boltzmann Machine (RBM)

-AutoEncoder (AE)

2.2 Convolutional Neural Networks

Convolutional Neural Network was inspired by how the animal visual cortex works [5],[9], [11],[12],[13],[14],[22]. CNN has 3 types of layers:

- Convolution layer which generates a two-dimensional feature map
- Polling layer for down-sampling
- Fully-connected layer connected at the end

CNN has shown promising results for image-based diagnosis in images of radiology, pathology and dermatology (Li, Zhang, Liu, Bu and Wei [4], Zhang et al. [22]).

The convolution layers are neurons arranged that will create a filter which contains pixels and moves across all the images and produces a feature map in the end [1],[2],[3],[4],[5],[6],[22]. The convolution values are later inserted in the activation function with the most used one being the ReLU. After that is the pooling layer which consists of a specific size filter that moves across the gesture map which was produced from the convolution layer.

2.4 Related Research

From 12 March 2020 when the status of a global pandemic was declared, the Covid-19 virus has been one of the most mentioned and most focused fields of study by the researchers. The outbreak of the pandemic affected most countries in rapid speed and left the medical qualified staff in difficulties in not only treating sick patients but also making correct diagnosis for more patients than they possibly could. To correctly report a Covid infected patient there is not only the need for laboratory tests such as CT images and chest X-Rays but also for healthcare staff to check the results and to decide on the diagnosis. Covid comes in the form of acute respiratory distress syndrome which means that in around 10 days of having symptoms, the patient’s chest CT should reveal lung abnormalities. These lung abnormalities detected by the CT in fact have proved to be of help for 42% of patients in detecting the condition and improving before doing a RT-PCR. This brings focus on the essential role of these CT chest scans in clinical treatment of Covid-19. (Sari, Widodo, Nugraheni and Wanda [1])

The Covid-19 diagnosis can be done by these tests which will also represent the sample images which will be used in classification (Tabik et al. [2]):

- CT scans which are defined as Computed Tomography is done by analyzing 3D radiographic images from different angles and requires not only time which is around 15 minutes for a patient but also equipment that is not found in all hospitals.
- Chest X-Ray which requires less effort in both equipment and time, taking only about 1 minute per patient, making it the most available test compared to the other two.

- RT-PCR test which is short for Reverse Transcription Polymerase Chain Reaction is done by a nasopharyngeal swab and requires specific materials and around 12 hours of time to confirm the results.

Based on the papers read related to the field it is seen that most often the researchers choose to classify Covid-19 chest CT scan images using deep learning and especially Convolutional Neural Network for the sole purpose of CNN being more efficient on solving image classification challenges by generating spatial features from samples.(Sari et al. [1])

A typical CNN architecture would be as in the Figure 1 presented below. The Convolutional neural network task is divided into two main sections in which first is Feature extraction in multiple hidden layers and the second is classification in output layer.

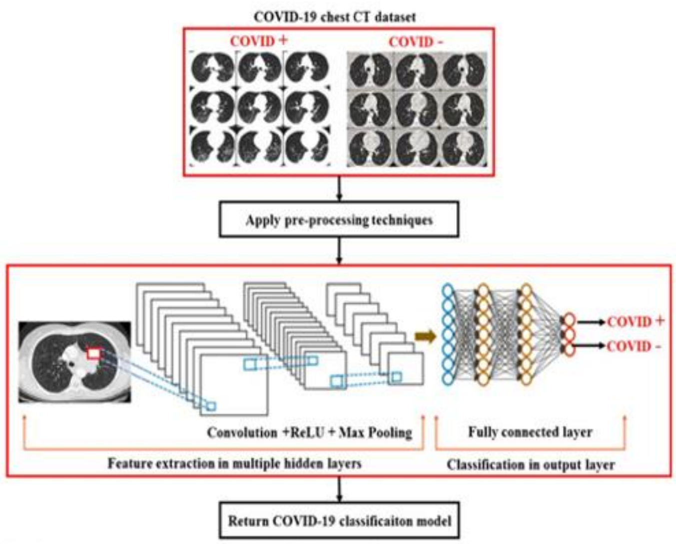

Figure 1: General CNN architecture

Sari et al. [1] do a comparison review on three different classification models: Inception ResNet-V2, Inception-V3 and their own approach. The dataset they trained the model on has 3651 samples while for the testing they have 1567 samples. The results are shown in Table 1 below.

Model	Acc	Rec	Pre	F1
Inception ResNet-V2 [20]	87	84	91	86
Inception-V3 [20]	97	94	100	96
Propose Approach	97.57	98	98	98

Table 1: Comparison ResNet-V2, Inception-V3, Sari et al. [1] model

Their model achieved better results in accuracy, recall, precision and F1. The accuracy their final model reached was 97.53%.

Sari et al. [2] and Li et al. [4] introduces us to an image preprocessing technique called Segmentation-Based Cropping: Unnecessary Information Elimination. X-ray scans, depending on where they are done, might have unnecessary information about the patient in the sides, contours that will not aid the classification or different positions and sizes of the patients which might add more parts of the body on the scan. This extra data might alter the classification results and this is why this technique is really important. It works by segmenting the lungs using the U-Net segmentation model and by calculating the smallest rectangle that delimits to the right and left segmented-lungs. Finally, the calculated rectangle is cropped. An example of this is shown in the image below.

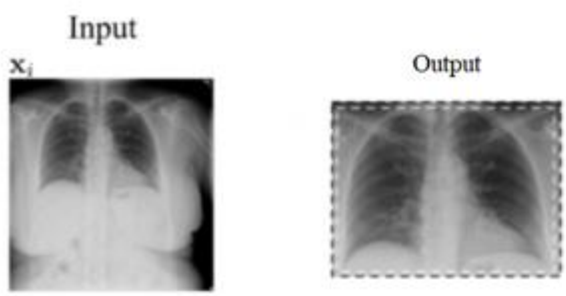

Figure 2: Segmentation-Based Cropping

Degadwala et al. [3] propose a Fine-tuned Convolution Neural Network (FT-CNN) which is based on the concept of learning transfer. Their proposed model is to train CNN first to learn the characteristics of the broader domain and using a segmentation function to reduce error. After that replacing the partition function to reduce the error in another domain. Results were those of accuracy in training 90.70% and accuracy in validation 90.54%

Tabik et al. [2] experiment with another CNN classifier which is Resnet-50 combined with ImageNet weights for a transfer learning method. To achieve this the last layer of the original model was removed and instead a ReLU activation layer and a softmax activation layer were added. Their model proved to be effective and highly accurate mostly for severe and moderate levels which according to the authors is because their approach needs important visual features. They use a really good practice to decide on when to stop the training of the model, which is by using 10% of the training set for validation.

Testing these different architectures CNN, VGG16, VGG19, InceptionV3 from Sevi and AYDIN [8], Kaya, Atas and Myderrizi [18] showing that VGG19 having the best result at 95% followed by Sevi et al.[8] VGG16 at 93% while the other two showing started not to learn the images anymore but memorizing them [8],[12],[14],[18],[21].

Islam and Matin [15] proposes LeNet-5 CNN using CT scan COVID-19 images resulting in an overall accuracy of 86%.

Eljamassi and Maghari [17] uses machine learning with HOG for feature extraction and KNN for classification. Overall results were around 85% but mostly due to low amounts of data.

Nath, Kanhe and Mishra [20] provides a deep learning method which can be used for both binary classification of Covid-19 vs No Covid-19 and multiclass classification which has an extra class for Pneumonia. The CNN model is tested on both CT scans and X-rays and attained 71.81% and 99.68% accuracy respectively for binary classification and 96.4% accuracy for multiclass classification. The model architecture is one of 24 layers containing six sets of convolution layers followed each by batch normalization and ReLU activation layer to secure non-linearity and max pooling to reduce the computation time. A SoftMax layer is used in the end to assign probabilities for each class.

The model in Dutta, Roy and Anjum [21] is a CNN with multiple layers and Transfer Learning model Inception V3 which uses convolution to extract features and involves weights of the ImageNet dataset to make the model better. The dataset with CT scans used is small so data augmentation is used to increase the number of samples. The techniques used are zoom, flip, horizontal, etc. The accuracy of the model improved 13% in comparison with the InceptionV3 model.

2.4 Research challenges and their possible solution

Drawbacks have been training on a small dataset because of the inability to have medical staff classify samples during these difficult times. Some researchers have used chest X-rays, others have taken into account CT images as well but CT images which are more difficult to obtain are more accurate. [5],[8],[11],[12],[14]

CHAPTER 3

3.METHODOLOGY

3.1 Dataset

In the beginning 2 different datasets were used. First one consists of only CT scans (102 images) of normal and COVID-19 (105 images) ones with a total of 207 images. The other dataset is from Kegel having x-ray images of COVID-19 (3’616 images), lung opacity (6’012 images), viral pneumonia (1’345 images) and normal (10’192 images).

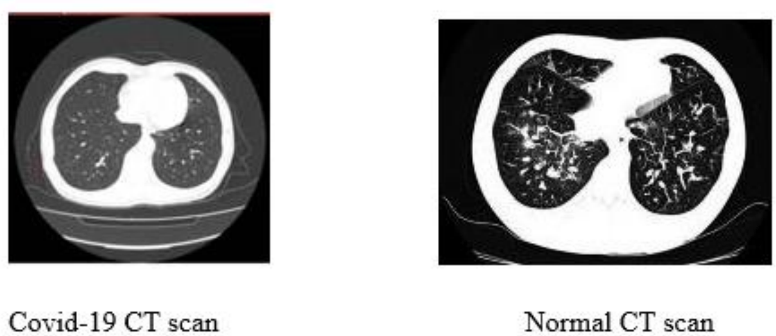

Figure 3: Covid-19 CT scan vs Normal CT scan

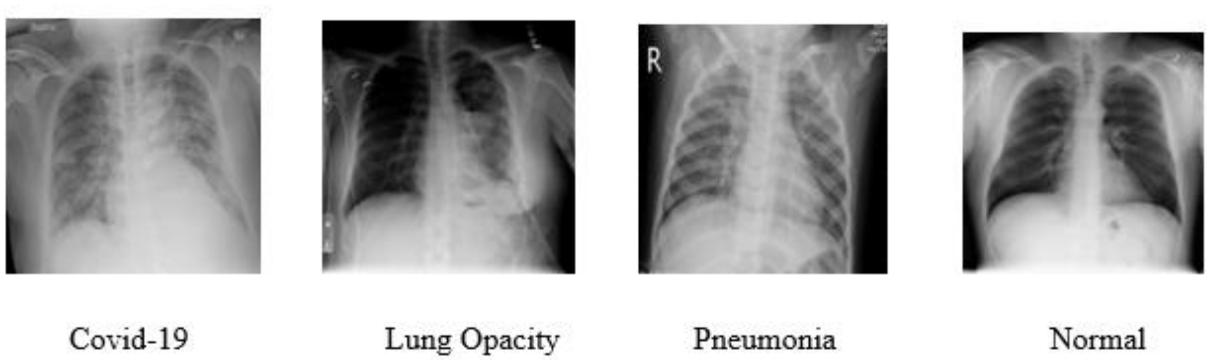

Figure 4: Dataset X-Ray Images

The x-ray dataset all images were of dimension 299x299. The CT scan one had dimensions of lowest being around 140x110 and highest around 800x500.

3.2 Image Preprocessing on the dataset

All images on the CT scan dataset were resized to 140x140. This was done to have consistent matrices when used in training and 140x140 was chosen as to not increase the resolution of a lot of images and impact the accuracy.

3.3 Deep Learning Architectures

For the Deep learning architecture used in CT-scan images, the core model is at least 2 dense layers with 100 nodes each with ReLu activation, followed by a dropout with 50% and output layer with SoftMax activation. Model uses Adam optimizer and sparse categorical cross entropy as loss function.

For the CNN architecture the core model consists of at least 2 Convolution 2D layers each having 64 nodes and using a kernel size of 3 by 3 with ReLu activation, max pooling of 2 by 2 and a dropout of 50%. Then the output of the last layer is flattened and is fed into a dense layer with 64 nodes which has a dropout of 50% and in the end is fed to a dense layer with 1 node which will be used as the output with activation sigmoid. The whole model uses binary cross entropy as loss function and Adam optimizer.

For the Classification CNN on X-Ray images the main model consists of at least 4 convolutional 2D layers each having 64 nodes, kernel size 3 by 3, with ReLu activation followed by a max pooling of 2 by 2 and a dropout of 60%. The last layer output is flattened and fed into a dense layer of 64 nodes with a dropout of 50%. Output layer uses SoftMax activation. Model uses Adam optimizer and sparse categorical cross entropy for loss function.

CHAPTER 4

4.RESULTS

4.1 Deep Learning on CT-Scan images

Using normal deep learning method with just dense layers will result in the accuracy getting stuck most of the times in ~66%. Adding dropouts, removing or adding more dense layers, increasing or decreasing the epoch size, increasing or decreasing the batch size, change the number of nodes per layer would still reach the same result. Results below are from the mentioned architecture with 20 epoch and batch size of

4.

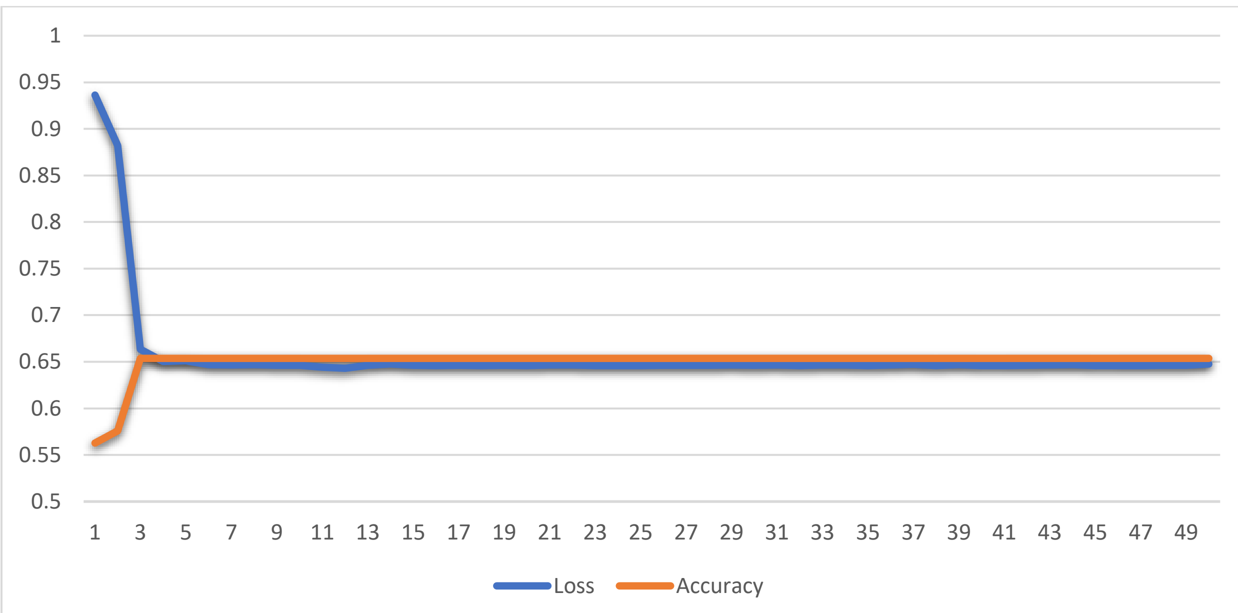

Figure 5: Deep Learning on CT-Scan Images

Sequential Model		
Layer (type)	Output Shape	Param #
Flatten	(None, 67500)	0
Dense	(None, 100)	6750100
Dropout	(None, 100)	0
Dense	(None, 100)	10100
Dense	(None, 2)	202

Table 2 : Deep Learning on CT-Scan Images Model Architecture

4.2 CNN on CT-Scan images

The used model is a Sequential Model which is similar to a stack of layers with each layer having one input and one output.

The structure of the convolutional neural network is shown in Table 12 below. The architecture used has 2 convolutional layers with kernels of size 3x3 applied followed by maxpooling layers which serves the purpose of decreasing the number of weights. The flow of the architecture consists in two sets of the following: CONV => RELU => POOL. To avoid overfitting Dropout is added to the architecture.

Sequential Model		
Layer (type)	Output Shape	Param #
Conv2D	(None, 148, 148, 64)	1792
Activation	(None, 148, 148, 64)	0
MaxPooling2D	(None, 74, 74, 64)	0
Dropout	(None, 74, 74, 64)	0
Conv2D	(None, 72, 72, 64)	36928
Activation	(None, 72, 72, 64)	0
MaxPooling2D	(None, 36, 36, 64)	0
Dropout	(None, 36, 36, 64)	0
Flatten	(None, 82944)	0
Dense	(None, 64)	5308480
Dropout	(None, 64)	0
Dense	(None, 1)	65
Activation	(None, 1)	0

Table 3 : CNN on CT-Scan Images Model Architecture

Using CNN on CT-scan shows better results but the model can easily overfit. Using 3 Convolutional Layers, 20 epochs with batch size 16 and testing of 20% result of accuracy close to 90% with slight overfitting of the model.

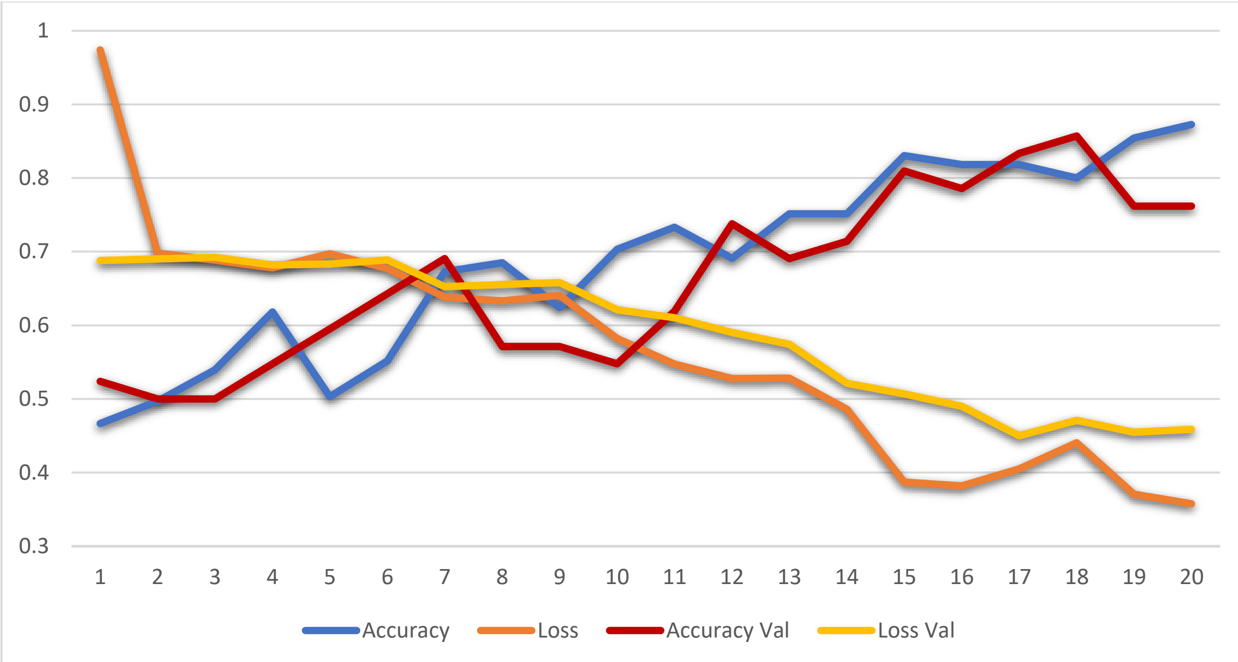

Figure 6: CNN on CT-Scan Images Model- 20 Epochs/Batch Size 16/ 3 Conv2D

Using only 2 Convolutional Layers will result in the model quickly overfitting after 12 epoch and reaching an accuracy before it overfits around 87%

Lowering the batch size to 8 and splitting the dataset to 10% testing reaches nearly the same accuracy as above of around 87% before quickly overfitting after epoch 8.

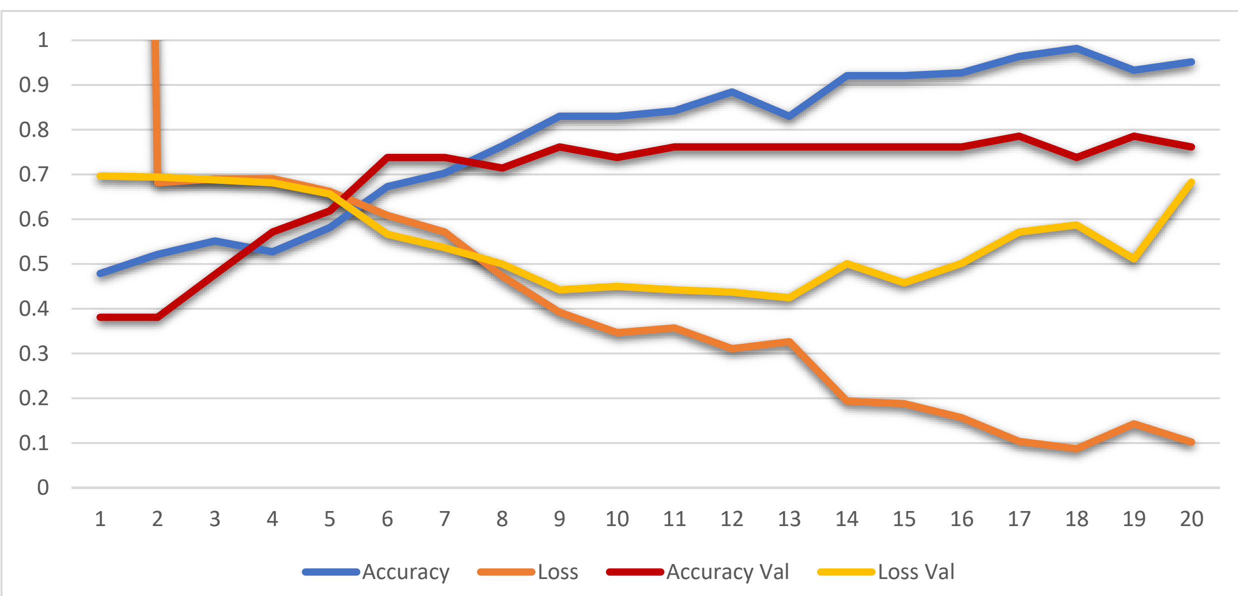

Figure 7: CNN on CT-Scan Images Model- 20 Epochs/Batch Size 16/ 2 Conv2D

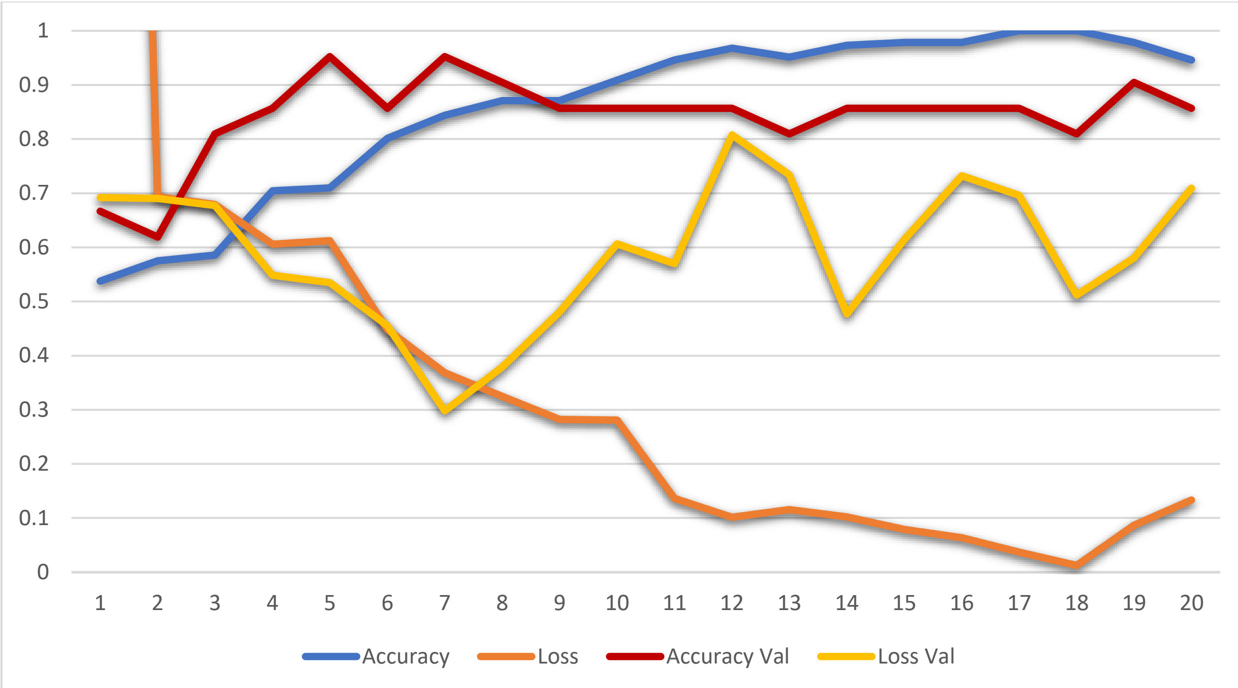

Figure 8: CNN on CT-Scan Images Model- 20 Epochs/Batch Size 8/ 3 Conv2D

Making the above model dropout more aggressive at 70% will sometimes make the model not overfit but the accuracy will drop to 85%

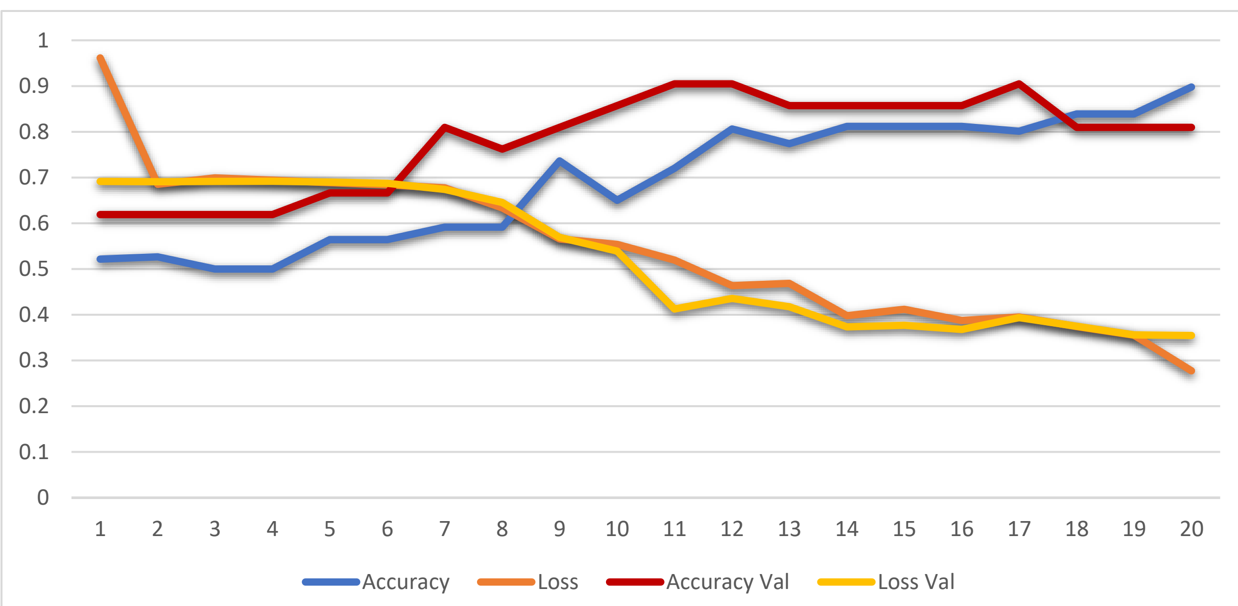

Figure 9: CNN on CT-Scan Images Model- 20 Epochs/Batch Size 8/ Adding Dropout

4.3 CNN on X-Ray images

The used model is similar to the Sequential Model used in the tests done for the CT-Scan images. It consists of 2 convolutional layers with kernels of size 3x3 applied, followed by maxpooling layers which serve the purpose of decreasing the number of weights.

Sequential Model		
Layer (type)	Output Shape	Param #
Conv2D	(None, 297, 297, 64)	1792
Activation	(None, 297, 297, 64)	0
MaxPooling2D	(None, 148, 148, 64)	0
Dropout	(None, 148, 148, 64)	0
Conv2D	(None, 146, 146, 64)	36928
Activation	(None, 146, 146, 64)	0
MaxPooling2D	(None, 73, 73, 64)	0
Dropout	(None, 73, 73, 64)	0
Flatten	(None, 341056)	0
Dense	(None, 64)	21827648
Dropout	(None, 64)	0
Dense	(None, 1)	65
Activation	(None, 1)	0

Table 4: CNN on X-Ray Images Model Architecture

For the following CNN model with epochs 50 when tested on X-Ray images, the results show the accuracy is close to ~97%. The Model is slightly overfit and it starts as soon as epoch 7.

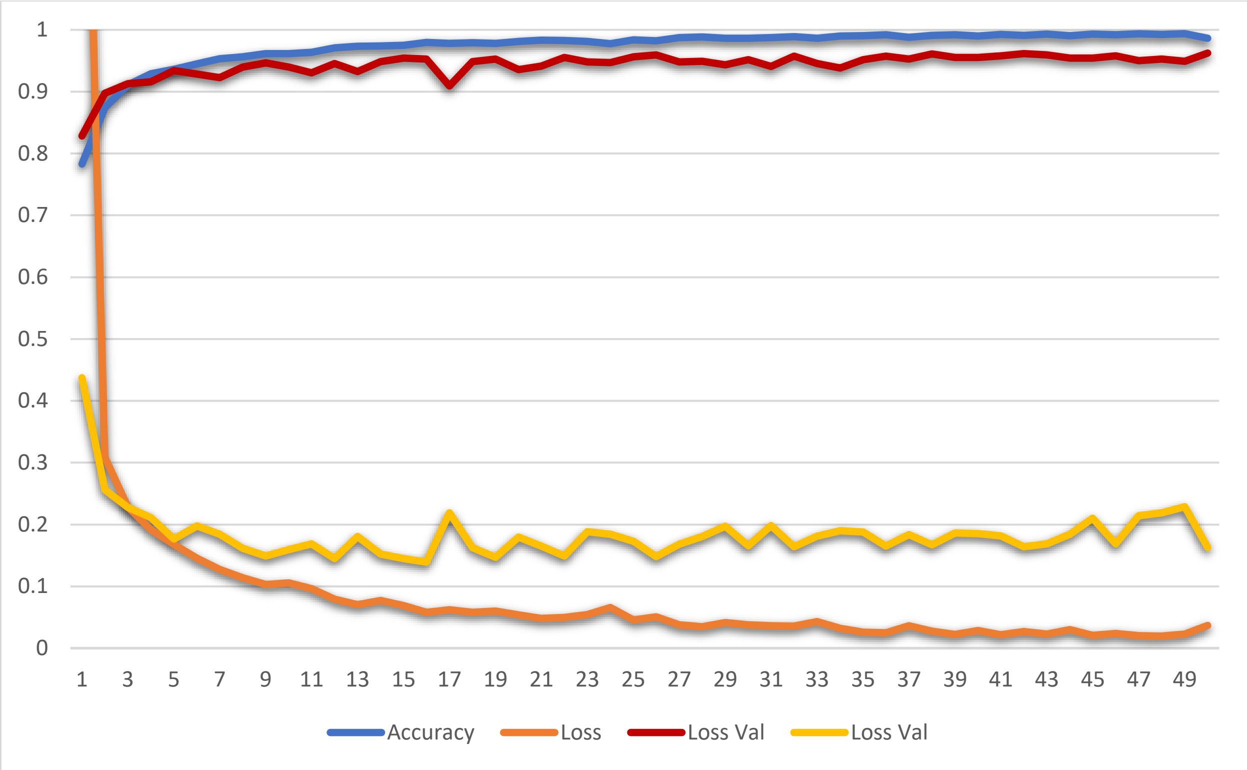

Figure 10: CNN on X-Ray Images Model- 50 Epochs/Batch Size 64

4.4 CNN Multi Class Classification

The same Sequential model which was used before for binary classification is used for multiclass classification and the results are as below.

The model tents to start really bad, to overfit a lot and have an accuracy of 75%-80% for 50 epochs.

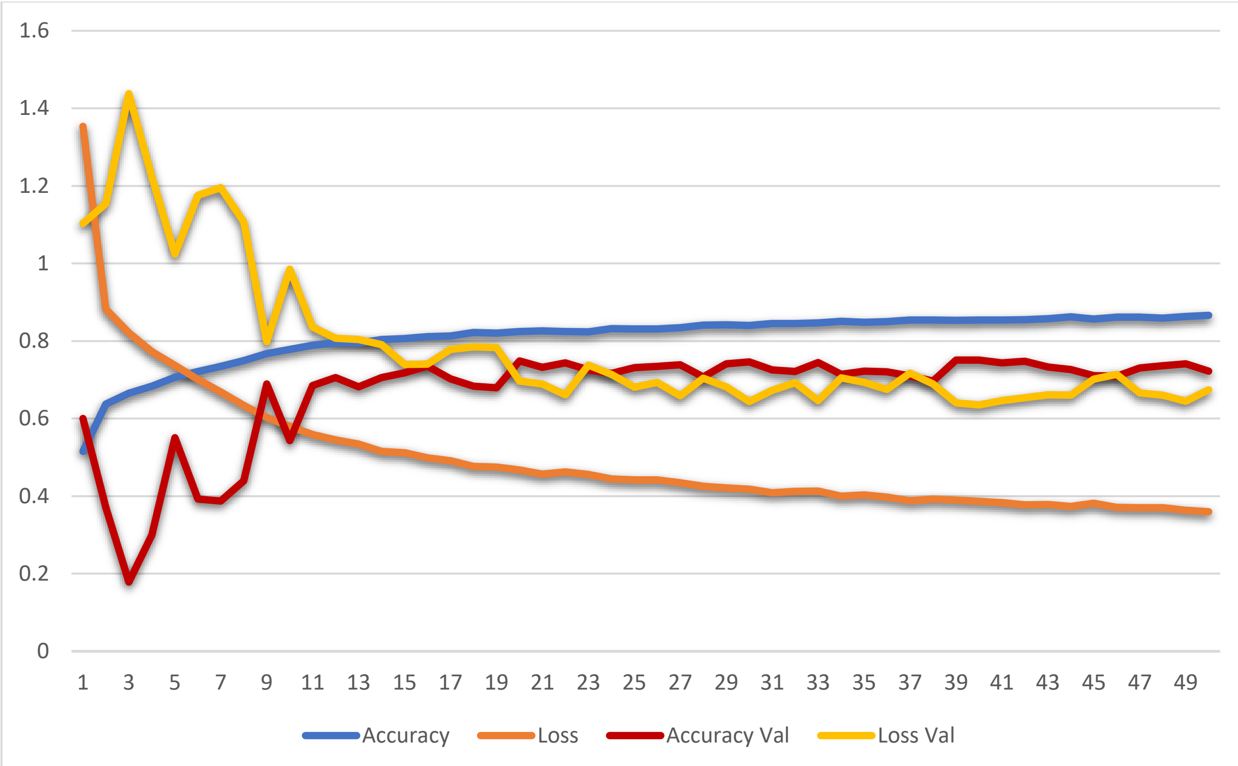

Figure 11: Multiclass classification CNN Model- 50 Epochs/Batch Size 64

Several changes were made in the architecture of the model for better performance:

- The kernel size which specifies the height and width of the 2D convolution window was changed from 3x3 to 5x5.
- Batch size after trying both variants 32 and 64 was set to be best to 64 for higher accuracy.
- The Dropout was added gradually with slight increment for each step of the model architecture.
- The epochs number was lowered since the size of the dataset led to overfitting.

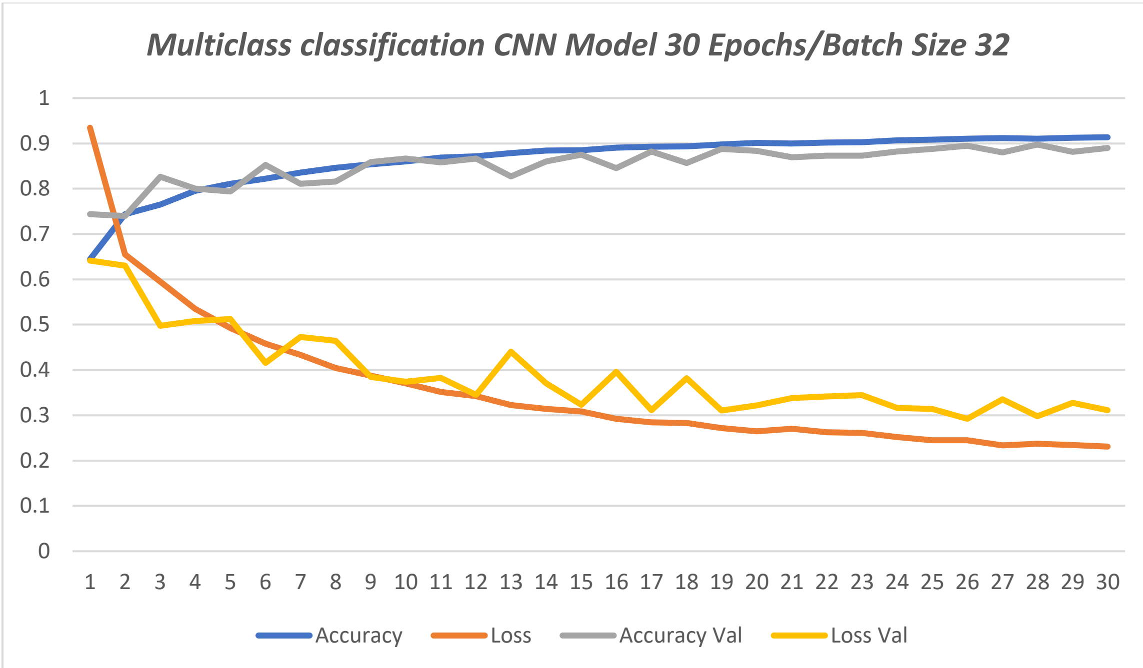

Figure 12: Multiclass classification CNN Model- 30Epochs/Batch Size 32

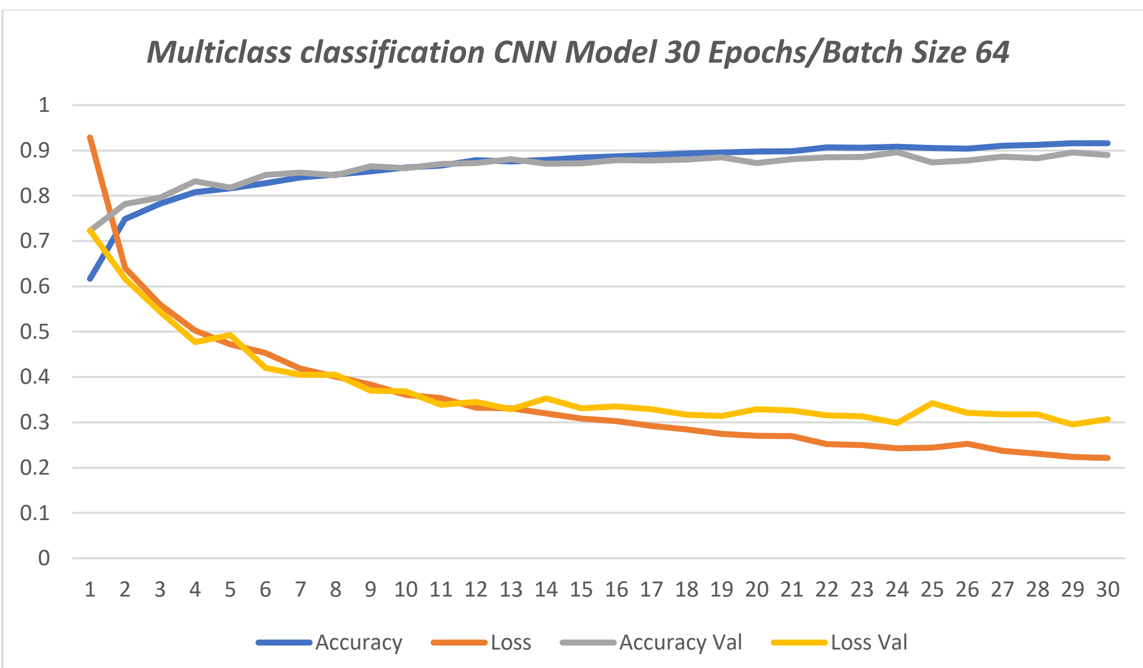

Figure 13: Multiclass classification CNN Model- 30Epochs/Batch Size 64

The CNN Multi Class Classification Model with batch size of 64 and 30 epochs reaches a training accuracy of 91% which outperforms the other previous models's accuracies for multiclass classification. Similarly, the same model with a different batch size of 32, has a close by performance in accuracy as it can also been seen in the comparison graph in Figure 14.

The main difference which also makes the model with batch size of 64 the one with the best performance so far in the research is the fact that its values are also more stable compared to the one with batch size 32. The training values of the validation loss and validation accuracy for the model with batch size 32 seem to flunctuate a lot as shown in Figure 12.

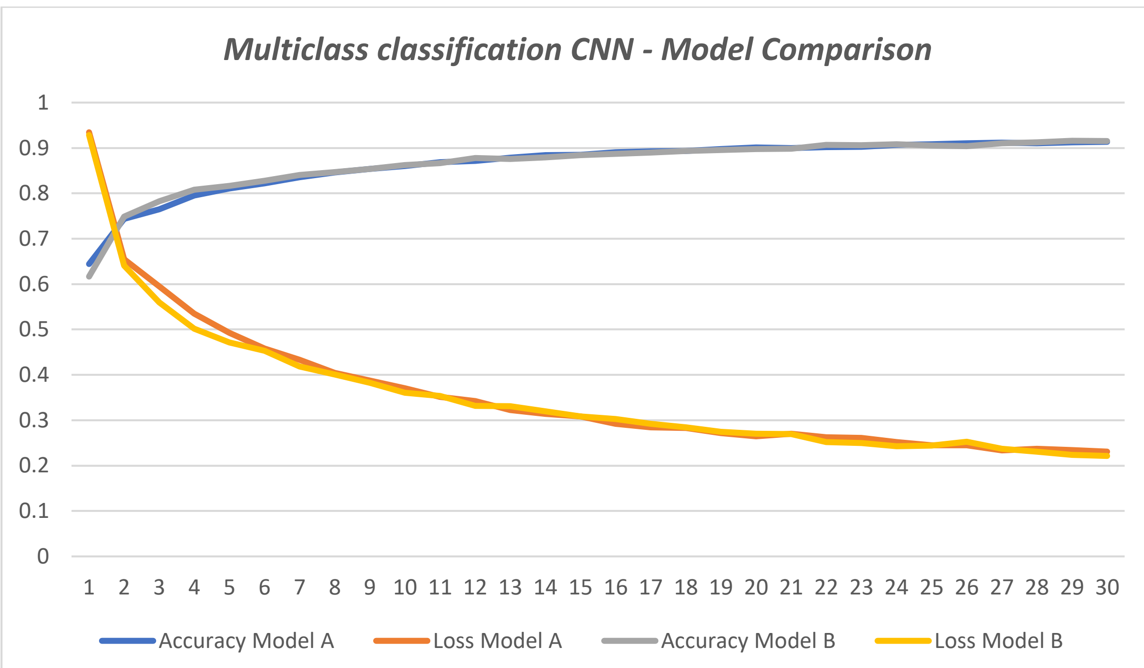

Figure 14: Multiclass classification CNN Model A/BS 32 and Model B/BS 64 Comparison

CHAPTER 5

5.CONCLUSIONS

5.1 Conclusions

The accuracy of Covid-19 image classification is heavily dependent on the availability of large datasets to be analyzed. This proves to be an issue as a result of the fact that the number of collected data samples remains too little and that there is a lack of balance in existing datasets between Covid-19, pneumonia, lung opacity and normal samples. The created models tend to easily overfit resulting in memorizing the data instead of learning it and finding features. During the experiments several batch sizes, epochs, kernel sizes, max pooling, optimizers, lost function have been tried with the aim to achieve a high accuracy for both binary and multiclass classification. The research and experiments with binary classification of the current model show promising results when used on X-Ray Scans resulting in a high accuracy close to ~97%, although a bit overfitting because of the dataset imbalance. The same CNN model was used for multiclass classification, at first reaching an accuracy of only 75% and later on with several improvements to the model it reached a higher accuracy of 91%.

5.2 Future Work

The next research goal would be to find a larger dataset for both X-Ray and CT-Scan images to make a more robust model which will not be biased to one class. The classification can be expanded even further to detect other diseases and to classify them more precisely but larger and more balanced datasets are required.

Effectively using data augmentation to create more data and to balance the dataset can be another solution but it should be further researched.

Different image preprocessing techniques can be used to further increase the accuracy and to decrease the training time. Such techniques could be noise removal, edge detection etc.

REFERENCES

1. I. P. Sari, Widodo, M. Nugraheni and P. Wanda, "A Basic Concept of Image Classification for Covid-19 Patients Using Chest CT Scan and Convolutional Neural Network," 2020 1st International Conference on Information Technology, Advanced Mechanical and Electrical Engineering (ICITAMEE), 2020, pp. 175-178, doi: 10.1109/ICITAMEE50454.2020.9398462.
2. S. Tabik et al., "COVIDGR Dataset and COVID-SDNet Methodology for Predicting COVID-19 Based on Chest X-Ray Images," in IEEE Journal of Biomedical and Health Informatics, vol. 24, no. 12, pp. 3595-3605, Dec. 2020, doi: 10.1109/JBHI.2020.3037127.
3. S. Degadwala, D. Vyas and H. Dave, "Classification of COVID-19 cases using Fine-Tune Convolution Neural Network (FT-CNN)," 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), 2021, pp. 609-613, doi: 10.1109/ICAIS50930.2021.9395864.
4. J. Li, D. Zhang, Q. Liu, R. Bu and Q. Wei, "COVID-GATNet: A Deep Learning Framework for Screening of COVID-19 from Chest X-Ray Images," 2020 IEEE 6th International Conference on Computer and Communications (ICCC), 2020, pp. 1897-1902, doi: 10.1109/ICCC51575.2020.9345005.
5. Z. KARHAN and F. AKAL, "Covid-19 Classification Using Deep Learning in Chest X-Ray Images," 2020 Medical Technologies Congress (TIPTEKNO), 2020, pp. 1-4, doi: 10.1109/TIPTEKNO50054.2020.9299315.
6. D. Nurtiyasari and D. Rosadi, "COVID-19 Chest X-Ray Classification Using Convolutional Neural Network Architectures," 2020 3rd International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), 2020, pp. 667-671, doi: 10.1109/ISRITI51436.2020.9315499.
7. A. U. Berliana and A. Bustamam, "Implementation of Stacking Ensemble Learning for Classification of COVID-19 using Image Dataset CT Scan and Lung X-Ray," 2020 3rd

International Conference on Information and Communications Technology (ICOIACT), 2020, pp. 148-152, doi: 10.1109/ICOIACT50329.2020.9332112.

8. M. Sevi and İ. AYDIN, "COVID-19 Detection Using Deep Learning Methods," 2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI), 2020, pp. 1-6, doi: 10.1109/ICDABI51230.2020.9325626.
9. D. Haritha, N. Swaroop and M. Mounika, "Prediction of COVID-19 Cases Using CNN with X-rays," 2020 5th International Conference on Computing, Communication and Security (ICCCS), 2020, pp. 1-6, doi: 10.1109/ICCCS49678.2020.9276753.
10. J. Vrindavanam, R. Srinath, H. H. Shankar and G. Nagesh, "Machine Learning based COVID-19 Cough Classification Models - A Comparative Analysis," 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), 2021, pp. 420-426, doi: 10.1109/ICCMC51019.2021.9418358.
11. D. A. Dharmawan and L. Listyalina, "COVID-19Net: A Deep Neural Network for COVID-19 Diagnosis via Chest Radiographic Images," 2020 1st International Conference on Information Technology, Advanced Mechanical and Electrical Engineering (ICITAMEE), 2020, pp. 232-237, doi: 10.1109/ICITAMEE50454.2020.9398392.
12. R. Bhadra and S. Kar, "Covid Detection from CXR Scans using Deep Multi-layered CNN," 2020 IEEE Bombay Section Signature Conference (IBSSC), 2020, pp. 214-218, doi: 10.1109/IBSSC51096.2020.9332210.
13. K. Foysal Haque, F. Farhan Haque, L. Gandy and A. Abdelgawad, "Automatic Detection of COVID-19 from Chest X-ray Images with Convolutional Neural Networks," 2020 International Conference on Computing, Electronics & Communications Engineering (iCCECE), 2020, pp. 125-130, doi: 10.1109/iCCECE49321.2020.9231235.
14. Jain G, Mittal D, Thakur D, Mittal MK. A deep learning approach to detect Covid-19 coronavirus with X-Ray images. *Biocybern Biomed Eng*. 2020;40(4):1391-1405. doi:10.1016/j.bbe.2020.08.008
15. M. R. Islam and A. Matin, "Detection of COVID 19 from CT Image by The Novel LeNet-5 CNN Architecture," 2020 23rd International Conference on Computer and

Information Technology (ICCIT), 2020, pp. 1-5, doi: 10.1109/ICCIT51783.2020.9392723.

16. P. A. K. K. Diallo and Y. Ju, "Accurate Detection of COVID-19 Using K-EfficientNet Deep Learning Image Classifier and K-COVID Chest X-Ray Images Dataset," 2020 IEEE 6th International Conference on Computer and Communications (ICCC), 2020, pp. 1527-1531, doi: 10.1109/ICCC51575.2020.9344949.
17. D. F. Eljamassi and A. Y. Maghari, "COVID-19 Detection from Chest X-ray Scans using Machine Learning," 2020 International Conference on Promising Electronic Technologies (ICPET), 2020, pp. 1-4, doi: 10.1109/ICPET51420.2020.00009.
18. A. Kaya, K. Atas and I. Myderrizi, "Implementation of CNN based COVID-19 classification model from CT images," 2021 IEEE 19th World Symposium on Applied Machine Intelligence and Informatics (SAMI), 2021, pp. 000201-000206, doi: 10.1109/SAMI50585.2021.9378646.
19. T. Anwar and S. Zakir, "Deep learning based diagnosis of COVID-19 using chest CT-scan images," 2020 IEEE 23rd International Multitopic Conference (INMIC), 2020, pp. 1-5, doi: 10.1109/INMIC50486.2020.9318212.
20. M. K. Nath, A. Kanhe and M. Mishra, "A Novel Deep Learning Approach for Classification of COVID-19 Images," 2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA), 2020, pp. 752-757, doi: 10.1109/ICCCA49541.2020.9250907.
21. P. Dutta, T. Roy and N. Anjum, "COVID-19 Detection using Transfer Learning with Convolutional Neural Network," 2021 2nd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST), 2021, pp. 429-432, doi: 10.1109/ICREST51555.2021.9331029.
22. S. Zhang, S. M. H. Bamakan, Q. Qu and S. Li, "Learning for Personalized Medicine: A Comprehensive Review From a Deep Learning Perspective," in IEEE Reviews in Biomedical Engineering, vol. 12, pp. 194-208, 2019, doi: 10.1109/RBME.2018.2864254.

APPENDIX

- **classification-cnn.py**

```
X = np.array(X).reshape(-1, 299, 299, 3)
X = X/255.0
Y = np.array(Y)

model = Sequential()
model.add(Conv2D(64, (3,3), input_shape = X.shape[1:]))
model.add(Activation("relu"))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.6))

model.add(Conv2D(64, (3,3)))
model.add(Activation("relu"))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.6))

model.add(Conv2D(64, (3,3)))
model.add(Activation("relu"))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.6))

model.add(Conv2D(64, (3,3)))
model.add(Activation("relu"))
model.add(MaxPooling2D(pool_size=(2,2)))
```

```
model.add(Dropout(0.6))

model.add(Flatten())
model.add(Dense(64))
model.add(Dropout(0.5))

model.add(Dense(4))
model.add(Activation('softmax'))

model.compile(loss="sparse_categorical_crossentropy",
optimizer="adam", metrics=['accuracy'])

epoch = 50

H = model.fit(X, Y, epochs=epoch, batch_size=64,
validation_split=0.3)
```

- **cnn_x-ray.py**

```
X = np.array(X).reshape(-1, 299, 299, 3)
X = X/255.0
Y = np.array(Y)

kernel_size = (3, 3)
pool = (2, 2)
dropOut = 0.5
optimizer = "relu"
outOptimizer = "sigmoid"
lossFunc = "binary_crossentropy"
lossOptimizer = "adam"
epoch = 50
```

```
batch = 64

model = Sequential()
model.add(Conv2D(64, kernel_size, input_shape = X.shape[1:]))
model.add(Activation(optimizer))
model.add(MaxPooling2D(pool_size=pool))
model.add(Dropout(dropOut))

model.add(Conv2D(64, kernel_size))
model.add(Activation(optimizer))
model.add(MaxPooling2D(pool_size=pool))
model.add(Dropout(dropOut))

model.add(Flatten())
model.add(Dense(64))
model.add(Dropout(dropOut))

model.add(Dense(1))
model.add(Activation(outOptimizer))

model.compile(loss=lossFunc, optimizer=lossOptimizer,
metrics=['accuracy'])


H = model.fit(X, Y, epochs=epoch, batch_size=batch,
validation_split=0.1)
```

- **resize_images.py**

```
import numpy as np
```

```python
import os
import cv2

datadir = "TEST/NON_COVID"
savedir = "Resized/Covid-F/"

image_array = []

image_size = (150, 150)

#Read all images
for img in os.listdir(datadir):
    image_array.append(cv2.imread(os.path.join(datadir, img)))

#Manual iterations of the names
name = iter(os.listdir(datadir))

#Iterate for every image in the dataset
for image in image_array:
    img_name = next(name)
    #In case of convertion failure
    try:
        image = cv2.resize(image, image_size)
        cv2.imwrite(savedir + img_name, image)
    except:
        print("Convertion Failed")
```

- **ct-scan_deep.py**

```python
model = tf.keras.Sequential([
        tf.keras.layers.Flatten(),
```

```
        tf.keras.layers.Dense(100, activation='relu'),
        tf.keras.layers.Dropout(0.5),
        tf.keras.layers.Dense(100, activation='relu'),
        tf.keras.layers.Dense(1,  activation='softmax')])

model.compile(optimizer='adam',
        loss="sparse_categorical_crossentropy",
        metrics=['accuracy'])



epoch = 20
H = model.fit(X_train, Y_train, epochs=epoch, batch_size=2)
val_loss, val_acc = model.evaluate(X_test, Y_test)
```

- **ct-scan_cnn.py**

```
X = np.array(X).reshape(-1, 150, 150, 3)
X = X/255.0
Y = np.array(Y)

model = Sequential()
model.add(Conv2D(64, (3,3), input_shape = X.shape[1:]))
model.add(Activation("relu"))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.7))

model.add(Conv2D(64, (3,3)))
model.add(Activation("relu"))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.7))
```

```
model.add(Flatten())
model.add(Dense(64))
model.add(Dropout(0.7))

model.add(Dense(1))
model.add(Activation('sigmoid'))

model.compile(loss="binary_crossentropy", optimizer="adam",
metrics=['accuracy'])

H = model.fit(X, Y, epochs=20, batch_size=8,
validation_split=0.1)
```