Application of CNN in Covid-19 Diagnosis using Chest X-ray Images

[1]Lim Zi Heng, [2]Lim Jia Qi
[1,2]UniversitiTunku Abdul Rahman (UTAR), Malaysia

Abstract:- The onset of the Coronavirus Disease 2019 (COVID-19) outbreak in early December 2019 has had profound and far-reaching repercussions on global public health. Despite being the gold standard for diagnosis, reverse transcription-polymerase chain reaction (RT-PCR) alone is unable to address the pandemic's urgent need for rapid and efficient diagnostic methods because of its time-consuming and complex nature. In this study, we propose a novel convolutional neural network (CNN) model, which is trained with a publicly available dataset, with targets of the normal, COVID-19, and viral pneumonia classes. The trained model achieved accuracy of 97.17% and specific recall of 94% in COVID-19 cases. A web application developed using the Python Flask framework is developed, whereby the users are able to upload X-ray images and acquire the prediction results and gradient activation map (Grad-CAM) of the images. This web app can help to provide a second opinion to medical practitioners regarding COVID-19 diagnosis.

Keywords:- *CNN, COVID-19 Diagnosis, GradCAM, Web Application, X-ray İmages.*

I. INTRODUCTION

According to the figures updated by the World Health Organization (WHO) till 2nd Nov 2023, a staggering700 million cases of COVID-19 and 6 million casualties had been reported[1]. To enhance diagnostic accuracy despite this highly transmissible global pandemic, it is strongly recommended to integrate clinical findings and results from medical imaging modalities, such as chest X-ray images (CXRs)[2], alongside the widely adopted diagnostic method, reverse transcription-polymerase chain reaction (RT-PCR)[3].Because RT-PCR has relatively low sensitivity ranging from 60%-70% [4] and long waiting time for the test results, detecting COVID-19 cases by examining medical images can lead to early management of the disease[5]. Thus, early detection of COVID-19 cases can curb the spread of the virus as the people infected can be isolated from the community.

Breakthroughs in artificial intelligence (AI) have significantly advanced the field of computer vision. One of the crucial achievements is the development of deep learning models, especially convolutional neural networks (CNNs). These CNNs have revolutionized image recognition tasks by learning intricate and robust features from vast amounts of data, leading to enhanced accuracy. In the same vein, deep learning has found its application in medical image analysis tasks. A deep learning model trained with medical images such as CXR can be useful for quick assessment of patients with potential symptoms. According to [6], the presence or absence of pathological findings on the CXR can determine whether a patient can be sent home or kept in hospital for further monitoring. The issues with manual inspection of CXR are that the process can be cumbersome and relies heavily on the knowledge and experience of the medical practitioners. Another challenge of using radiology for diagnosis is theambiguity and commonalities of clinical and radiological features that characterize COVID-19 and other similar lung diseases. Nonetheless, CXR can be a suitable modality because abnormalities caused by COVID-19 are visible in CXR in the form of ground-glass opacities, which can assist in the detection of COVID-19 [7].

Therefore, the development of computerized methods for faster and more accurate COVID-19 diagnosis is highly desirable. CNNs have emerged as the preferred method for analyzing medical images, demonstrating superior performance over traditional ML techniques.

➢ *The Objectives of this Study are Three-Fold:*

- Design and train a custom CNN model using CXR to diagnose COVID-19.
- Develop a simple and user-friendlyweb application with the aid of Flask, a web app framework in Python. The custom trained CNN model will work at the back end to produce predictions once the user has uploaded and submitted an image.
- Use the gradient-weighted class activation mapping (Grad-CAM) technique on CXR to pinpoint the specific lung regions that the deep learning model is focusing on to make its prediction.

II. RELATED WORKS

The application of DL models generally follows the general workflow of the supervised learning framework which includes using chest X-ray as an image dataset, data augmentation, model training, and model evaluation. The biggest challenge of applying computer vision DL models in the diagnosis of COVID-19 is the insufficiency of labeled image data.

We will organize the discussion of the literature review based on the major stages in the supervised learning approach, as shown in Figure 1 and Table 1 in the following subsections.

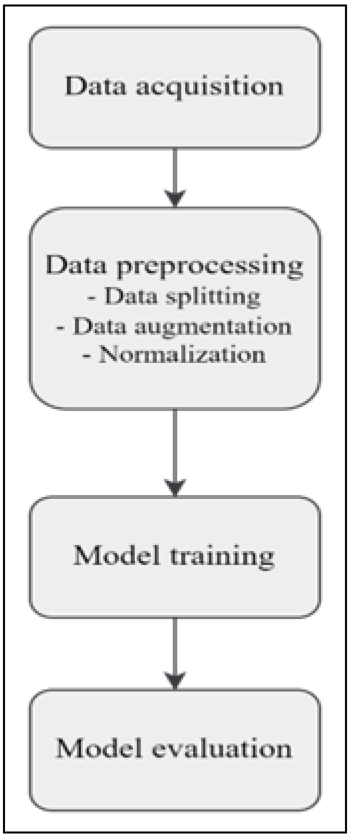

Fig 1 Primary Workflow of Supervised Learning

Table 1 Review of Previous Studies on COVID-19 Classification using CXR İmages with CNN Model

Papers	Stages of supervised learning	Details	
[8]	Data sources	*Classes*	*Number of samples*
		Normal	27,228
		Pneumonia	5.794
		COVID-19	209
	Data preprocessing	Data augmentation of radiograph images (DARI), resizing, and rotation.	
	Model	Proposed custom CNN model	
	Evaluation metrics	93.94% COVID-19 detection accuracy and 0.9525 AUC score.	
[9]	Data sources	*Classes*	*Number of samples*
		Normal	315
		Bacterial Pneumonia	300
		Viral Pneumonia	350
		COVID-19	250
	Data preprocessing	Resizing, normalization, and data augmentation (rotation and Gaussian blur).	
	Model	ResNet50 to classify bacterial pneumonia, normal case, and viral pneumonia. ResNet101 to classify viral pneumonia cases into COVID-19 and non-COVID-19.	
	Evaluation metrics	98.93% accuracy	
[7]	Data sources	*Classes*	*Number of samples*
		Normal	1,341
		Pneumonia	1,345

		COVID-19	361
	Data preprocessing	1. Fuzzy color image enhancement 2. Stack the fuzzy color images and original images	
	Model	InstaCovNet-19 with multiple fine-tuned pretrained models.	
	Evaluation metrics	99.08% detection accuracy	
[5]	Data sources	*Classes*	*Number of samples*
		Normal	3,108
		Pneumonia	1,439
		COVID-19	2,843
	Data preprocessing	Resizing and data labeling for binary, 3-class and 4-class classification tasks.	
	Model	Custom 22-layer CNN, called CoroDet.	
	Evaluation metrics	94.2% overall accuracy, with 97.47% COVID-19 recall.	
[6]	Data sources	6 subsets of data are constructed from 3 data repositories.	
	Data preprocessing	*	
	Model	Pretrained model, AlexNet, GoogleNet, and SqueezeNet.	
	Evaluation metrics	Dataset 4 (3 classes, all models): 96% accuracy Dataset 6 (3 classes, all models): 96% accuracy	
[10]	Data sources	*Classes*	*Number of samples*
		Normal	145
		Pneumonia	145
		COVID-19	145
	Data preprocessing	Data augmentation and synthesis usingcGAN.	
	Model	Custom lightweight CNN	
	Evaluation metrics	98.3% accuracy (3-class classification)	
[11]	Data sources**	*Classes*	*Number of samples*
		Normal	300
		Bacterial pneumonia	300
		Viral pneumonia	300
		COVID-19	300
	Data preprocessing	*	
	Model	Feature extraction with ShuffleNet and SqueezeNet + downstream classification with multiclass SVM.	
	Evaluation metrics	94.44% accuracy	

* Not mentioned in the paper.
** Chest radiography includes CXR and CT images.

III. METHODS

The workflow used to train and evaluate the CNN model is shown in Figure 2. The CXR image dataset was downloaded from theKaggle repository [12, 13]. Next, the image dataset is partitioned into 3 parts: training, validation, and test sets. 80% of the CXR is randomly sampled to form training data, while another 20% is randomly and evenly allocated to form validation and test data.

Then, the training dataset will undergo data preprocessing and data augmentation. The resulting images will be used to train the proposed custom CNN model. When the validation loss curve indicates overfitting, the CNN model is subjected to hyperparameter tuning to refine its performance and boost its predictive abilities.The CNN model trained with the optimal hyperparameter settings will then be evaluated against the test dataset.Performance metrics such as accuracy, sensitivity, precision, and F1 score are calculated.

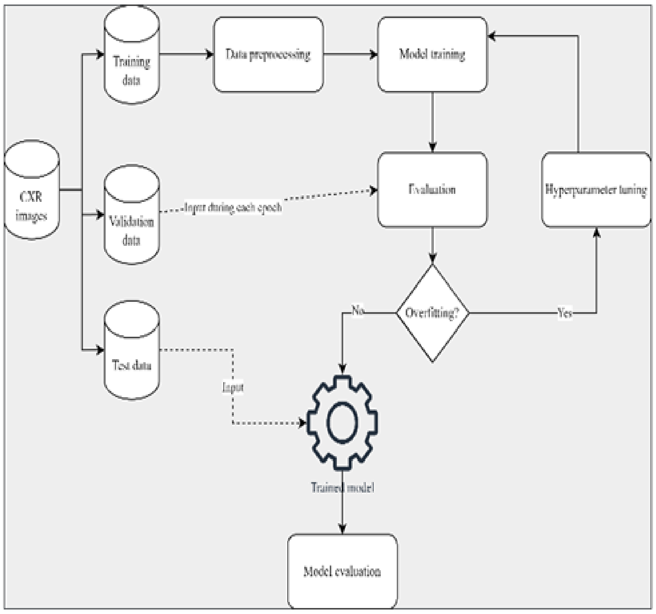

Fig 2 Experiment Workflow

A. Data Acquisition

The dataset used in this study contains **3616 COVID-19-positive samples**, **10,192 normal samples**, and **1345 images of viral pneumonia** complete with associated lung masks.

B. Data Preprocessing

Before feature extraction and model training, the acquired CXRs undergo a series of preprocessing steps to enhance their quality, standardize their format, and ensure compatibility with the deep learning model. In this project, the input images were resized to a uniform resolution of **256x256**. This ensures that the model can process inputs of consistent size while reducing computational complexity and mitigating the risk of biases introduced by varying image dimensions. In addition, normalization, a crucial data preprocessing step, plays a pivotal role in ensuring that feature values are uniformly scaled across the dataset. This process aligns diverse features to a common scale, ensuring comparable magnitudes and preventing any single feature from dominating the model's learning process[14]. In this project, the **mean** valuesof**(0.5, 0.5, 0.5)are subtracted from the pixels of each channel,** and then divided bythe value of **standard deviation** as**(0.5, 0.5, 0.5).**

➢ *Data Augmentation*

Deep learning models require a vast amount of data due to the large number of parameters and learning complex patterns from data. This is done by creating new data points from existing data using various transformations, such as cropping, rotating, flipping, and adding noise[15]. By augmenting the data, we can provide the deep learning model with more diverse examples. Enhancing the resilience and generalizability of deep learning models is the primary goal of data augmentation, particularly when dealing with limited or imbalanced datasets[16]. We apply two common data augmentation techniques employed for image classification tasks: random croppingof the original image to **224 X 224 in dimension** and horizontal flipping. By exposing the model to these geometric transformations, we prevent it from memorizing spurious patterns and thus help mitigate overfitting.Data augmentation helps to overcome this overfitting tendency by introducing new variations and perspectives into the training process.

C. CNN Model Training

➢ *Model Architecture*

The proposed CNN model backbone is made up of a convolutional block, as shown in Figure 3. The convolutional block comprises alternating convolutional layers and the ReLU activation function.

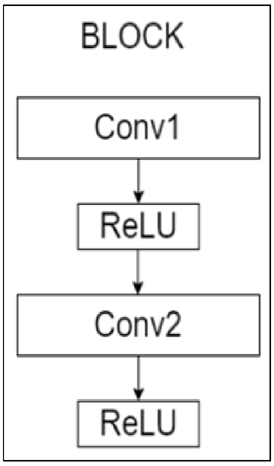

Fig 3 Convolutional Block of the Proposed CNN Model

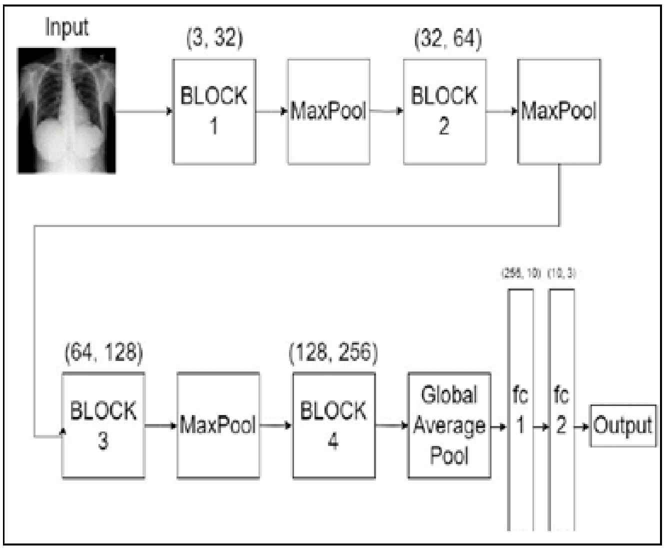

Fig 4 Proposed CNN Model Architecture. The Tuple above the Block Denotes the Number of İnput and Output Filters

Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 32, 224, 224]	896
Conv2d-2	[-1, 32, 224, 224]	9,248
BLOCK-3	[-1, 32, 224, 224]	0
Conv2d-4	[-1, 64, 112, 112]	18,496
Conv2d-5	[-1, 64, 112, 112]	36,928
BLOCK-6	[-1, 64, 112, 112]	0
Conv2d-7	[-1, 128, 56, 56]	73,856
Conv2d-8	[-1, 128, 56, 56]	147,584
BLOCK-9	[-1, 128, 56, 56]	0
Conv2d-10	[-1, 256, 28, 28]	295,168
Conv2d-11	[-1, 256, 28, 28]	590,080
BLOCK-12	[-1, 256, 28, 28]	0
AdaptiveAvgPool2d-13	[-1, 256, 1, 1]	0
Linear-14	[-1, 10]	2,570
Linear-15	[-1, 3]	33

Total params: 1,174,859
Trainable params: 1,174,859
Non-trainable params: 0

Input size (MB): 0.57
Forward/backward pass size (MB): 68.91
Params size (MB): 4.48
Estimated Total Size (MB): 73.96

Fig 5 Snapshot of the Model Architecture, Number of Parameters, and Memory Footprint

➢ *Hyperparameter Setting*

Optimizing hyperparameter selection is paramount for achieving accurate COVID-19 identification using CXR images because it profoundly influences the performance and effectiveness of the deep learning model. Each hyperparameter setting exerts a profound impact on various facets of the model's training process, ultimately determining its ability to distinguish COVID-19 cases from other pulmonary conditions. The final hyperparameter settings are shown in Table 2.The CNN model was trained on Google Colab in PyTorch (2.1.0+cu118) framework.

Table 2 Final Hyperparameter Settings

Hyperparameters	**Values**
Number of Epochs	100
Batch size	32
Optimizer	Stochastic gradient descent with momentum
Momentum	0.9
Initial learning rate	0.01
Learning rate scheduler	Piecewise constant function
Gamma	0.1
Step size	40

D. *Model Evaluation*

A few common model evaluation metrics, such as accuracy, precision, recall and F1-score were calculated for the models trained. The formula of these metrics can be expressed as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1_{score} = \frac{2(Precision \times Recall)}{Precision + Recall} \tag{4}$$

E. *Web Application*

After the model training and evaluation, the trained model will be deployed using the Python Flask API. Figure 6 shows the home page of the web application.

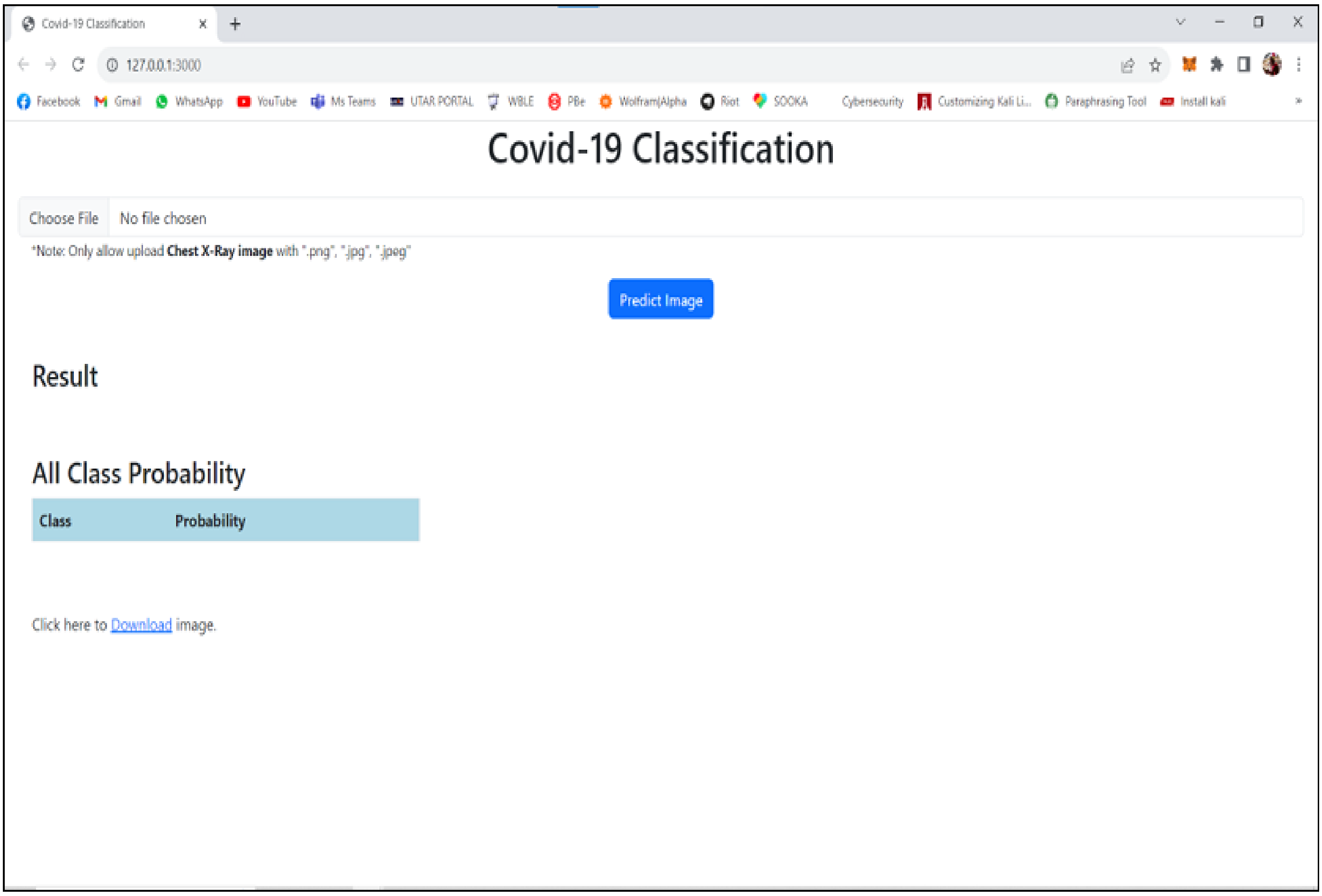

Fig 6 Design of the Home Page

This app lets users upload a JPEG image and obtain a heatmap called gradient-weighted class activation mapping (Grad-CAM)[17], which shows the image regions most important for the prediction, as well as the predicted class and probabilities for all classes. Users can also download the Grad-CAM image, as shown in Figure 7.

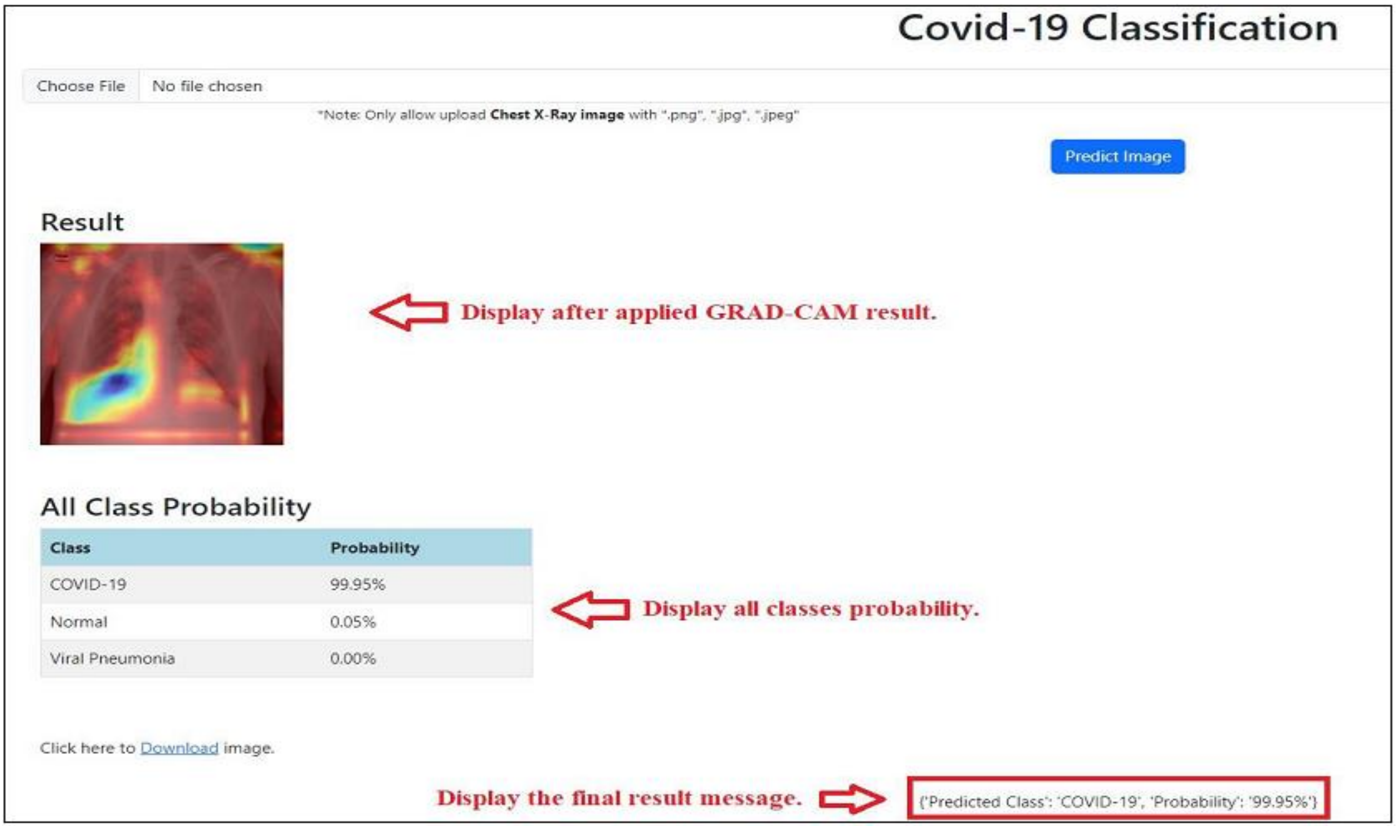

Fig 7 Prediction Outputs are shown on the user Interface (UI)

IV. RESULTS AND DİSCUSSİON

As shown inFigure 8, the training and validation loss graph is from the CNN model trained with the last hyperparameter setting shown as the last row of table 3. The validation loss curve follows the trend of the training loss curve and a relatively small gap when the loss curve starts to plateau, indicating that the model is not overfitting.

Table 3 Results with Different Hyperparameter Settings

<table>
<tr><th rowspan="2">Trials</th><th rowspan="2">Epoch</th><th rowspan="2">Learning rate</th><th colspan="2">Learning rate scheduler</th><th rowspan="2">Dropout (probability)</th><th rowspan="2">Test accuracy (%)</th></tr>
<tr><th>Step size</th><th>Gamma</th></tr>
<tr><td>1</td><td rowspan="3">50</td><td rowspan="5">0.01</td><td rowspan="2">10</td><td rowspan="5">0.1</td><td>0</td><td>78.25</td></tr>
<tr><td>2</td><td>0.5</td><td>70.67</td></tr>
<tr><td>3</td><td>20</td><td rowspan="3">0</td><td>91.43</td></tr>
<tr><td>4</td><td rowspan="2">100</td><td>30</td><td>96.37</td></tr>
<tr><td>5</td><td>40</td><td>97.17</td></tr>
</table>

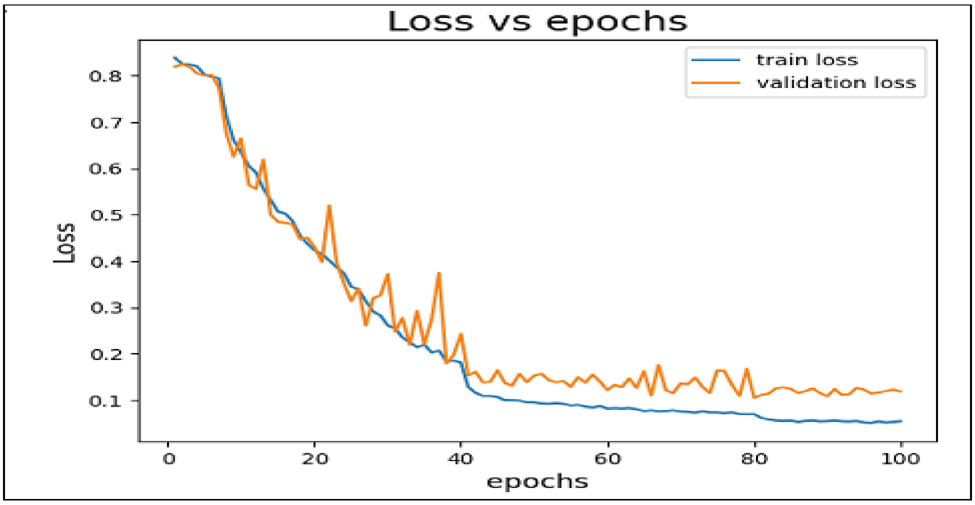

Fig 8 Training and Validation Loss Curves

Based on the entries of the confusion matrix shown in Figure 9, the precision, recall, and F1-score of COVID-19 can be calculated using equations (1-4). The precision, recall and F1-score of the COVID-19 class were**0.9522**, **0.9365** and **0.9443,** respectively.

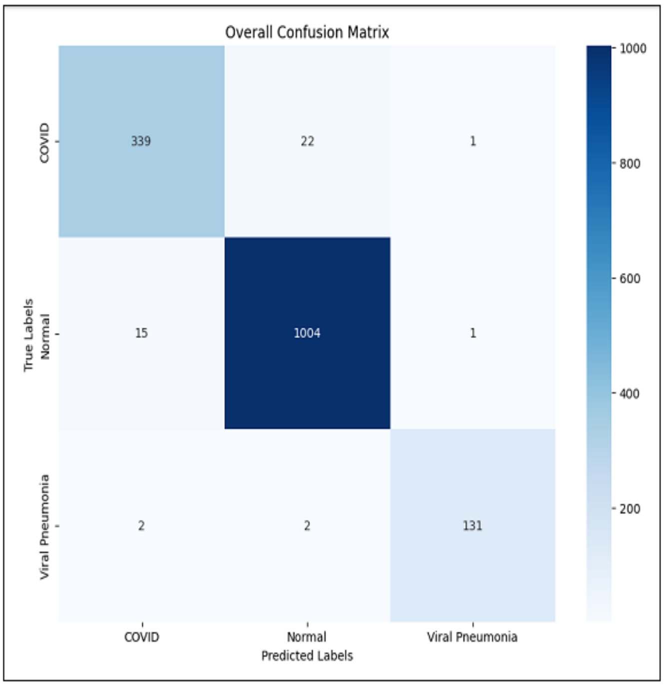

Fig 9 Confusion Matrix of the Model Performance

V. CONCLUSION

In this study, we propose and develop a novel and accurateCNN model for COVID-19 diagnosis using CXR images. The proposed model achieves a test accuracy of 97%, demonstrating its ability to differentiate COVID-19 cases from other lung conditions. Furthermore, our approach dramatically reduces the training time to a mere 2h, establishing it as a resource-efficient solution. In addition, we implemented GradCAM, a technique that generates heatmaps superimposed on CXR images, highlighting the most crucial regions for the CNN model's prediction. To further enhance the project's utility, we introduce a simpleuser interface that empowers medical professionals to diagnose COVID-19 through lung image analysis.

FUTURE WORKS

➢ *The Research Scope of this Project is Limited; thus, Several Promising Directions for Future Development can be Explored:*

- Incorporating multiple diagnostic modalities, such as clinical information and additional imaging techniques,could result in a more comprehensive and precise COVID-19 diagnostic tool.
- Model interpretability should be considered to make sense of the prediction results [18].
- Expanding the model's capabilities to identify and differentiate between various lung diseases by collecting more images from various lung conditions.

REFERENCES

[1]. (2023). WHO Coronavirus (COVID-19) Dashboard. Available: Https://Covid19.Who.İnt/

[2]. Pascarella Et Al., "COVID-19 Diagnosis And Management: A Comprehensive Review," J Intern Med, Vol. 288, No. 2, Pp. 192-206, Aug 2020.

[3]. M. Shen Et Al., "Recent Advances And Perspectives Of Nucleic Acid Detection For Coronavirus," J Pharm Anal, Vol. 10, No. 2, Pp. 97-101, Apr 2020.

[4]. Tahamtan And A. Ardebili, "Real-Time RT-PCR İn COVID-19 Detection: İssues Affecting The Results," Expert Review Of Molecular Diagnostics, Vol. 20, No. 5, Pp. 453-454, 2020/05/03 2020.

[5]. E. Hussain, M. Hasan, M. A. Rahman, I. Lee, T. Tamanna, And M. Z. Parvez, "Corodet: A Deep Learning Based Classification For COVID-19 Detection Using Chest X-Ray İmages," Chaos, Solitons & Fractals, Vol. 142, P. 110495, 2021/01/01/ 2021.

[6]. T. D. Pham, "Classification Of COVID-19 Chest X-Rays With Deep Learning: New Models Or Fine Tuning?," Health Inf Sci Syst, Vol. 9, No. 1, P. 2, Dec 2021.

[7]. Gupta, Anjum, S. Gupta, And R. Katarya, "Instacovnet-19: A Deep Learning Classification Model For The Detection Of COVID-19 Patients Using Chest X-Ray," Applied Soft Computing, Vol. 99, P. 106859, 2021/02/01/ 2021.

[8]. S. Sakib, T. Tazrin, M. M. Fouda, Z. M. Fadlullah, And M. Guizani, "DL-CRC: Deep Learning-Based Chest Radiograph Classification For COVID-19 Detection: A Novel Approach," IEEE Access, Vol. 8, Pp. 171575-171589, 2020.

[9]. Jain, D. Mittal, D. Thakur, And M. K. Mittal, "A Deep Learning Approach To Detect Covid-19 Coronavirus With X-Ray İmages," Biocybernetics And Biomedical Engineering, Vol. 40, No. 4, Pp. 1391-1405, 2020/10/01/ 2020.

[10]. S. Karakanis And G. Leontidis, "Lightweight Deep Learning Models For Detecting COVID-19 From Chest X-Ray İmages," Computers İn Biology And Medicine, Vol. 130, P. 104181, 2021/03/01/ 2021.

[11]. S. Elkorany And Z. F. Elsharkawy, "Covıdetection-Net: A Tailored COVID-19 Detection From Chest Radiography İmages Using Deep Learning," Optik, Vol. 231, P. 166405, 2021/04/01/ 2021.

[12]. T. Rahman Et Al., "Exploring The Effect Of İmage Enhancement Techniques On COVID-19 Detection Using Chest X-Ray İmages," Computers İn Biology And Medicine, Vol. 132, P. 104319, 2021/05/01/ 2021.

[13]. M. E. H. Chowdhury Et Al., "Can AI Help İn Screening Viral And COVID-19 Pneumonia?," IEEE Access, Vol. 8, Pp. 132665-132676, 2020.

[14]. K. Maharana, S. Mondal, And B. Nemade, "A Review: Data Pre-Processing And Data Augmentation Techniques," Global Transitions Proceedings, Vol. 3, No. 1, Pp. 91-99, 2022/06/01/ 2022.

[15]. L. Perez And J. Wang, "The Effectiveness Of Data Augmentation İn İmage Classification Using Deep Learning," Arxiv Preprint Arxiv:1712.04621, 2017.
[16]. M. S. Jerry Tang, Ruohan Zhang, "Explaining The Effect Of Data Augmentation On Image Classification Tasks."
[17]. R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, And D. Batra, "Grad-Cam: Visual Explanations From Deep Networks Via Gradient-Based Localization," İn Proceedings Of The IEEE İnternational Conference On Computer Vision, 2017, Pp. 618-626.
[18]. X. Li Et Al., "Interpretable Deep Learning: İnterpretation, İnterpretability, Trustworthiness, And Beyond," Knowledge And Information Systems, Vol. 64, No. 12, Pp. 3197-3234, 2022/12/01 2022.