Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 32 No. 10s (2025)

Enhance Deep Learning-Based Framworks dor Efficient Covid-19 and Pneumonia Detection

Jeevan Kumar[1], Vijay Pandey[2], Rajesh Kumar Tiwari[3]

Department of CSE[1, 3], Department of ME[2]
JUT, Ranchi, Jharkhand, India[1], BIT Sindri, Jharkhand, India[2], RVSCET Jamshedpur, Jharkhand, India[3]
jeevancse01@gmail.com[1], vpandey.me@bitsindri.ac.in[2], rajeshkrtiwari@yahoo.com[3]

Article History:

Received: 02-01-2025

Revised: 25-02-2025

Accepted: 20-03-2025

Abstract: Covid-19, a contagious disease caused by a new virus not before encountered in humans, has clinical features in common with the flu, displaying symptoms such as cough and fever. In this research paper, we introduce a deep learning approach to rapidly detect Covid-19 and pneumonia from chest X-ray images. Using new object detection architecture, which was trained and tested on a 5,160-image dataset, the model distinguishes between non-infected patients and Covid-19 and pneumonia-affected patients. The new Convolutional Channel Spatial Attention Feature Extraction Model (CCSAFEM) obtained an accuracy of 98.70% in Covid-19 and pneumonia case classification. This shows the capability of deep learning to precisely detect respiratory diseases, an important aspect in the battle against Covid-19 and pneumonia.

Keywords: Convolutional neural network, Covid19, Pneumonia, ResNet50, X-ray images, deep learning.

1. Introduction

Toward the close of 2019, a novel coronavirus emerged in Wuhan, China, subsequently named Covid-19. The Covid-19 has a profound global impact, affecting nearly every part of the world. Notably, older individuals are disproportionately affected compared to their younger counterparts. The virus primarily causes symptoms related to the respiratory system, including fever, coughing, taste and smell loss, and shortness of breath. These symptoms are often shared with other viruses like the common cold. Mitigating the spread of the virus involves practices such as frequent hand washing with sanitizer, wearing masks, maintaining physical distance, and receiving vaccinations. Covid-19 differs from other viruses in its lengthy incubation period, ranging from three to thirteen days, with an average onset of symptoms occurring around five to six days after exposure. The convergence of these factors renders Covid-19 particularly challenging to detect, trace, and contain, contributing to its rapid transmission.

Globally, approximately two million individuals are affected by pneumonia annually, with the potential for mortality if left untreated. Timely identification of pneumonia is crucial. For accurate diagnosis and to minimize misinterpretation, prompt evaluation by a trained person using chest X-rays is essential [1]. Pneumonia can range from mild to severe, regardless of the causative bacteria, and can be influenced by factors such as age and overall health. Common symptoms include cold and flu-like symptoms, which can escalate to life-threatening conditions. Other symptoms may include difficulty breathing, chest pain, cough with phlegm, fatigue, nausea, vomiting, shortness of breath, sweating, and shaking chills. The purpose of this study is to improve the rapid and accurate diagnosis of Covid-19 and pneumonia patients. The proposed model made use of X-ray images and

deep learning methods, including CNNs. There is a pressing need for a reliable and precise solution to aid healthcare decision-making based on medical imaging. The use of deep learning techniques to help treat Covid-19 and pneumonia patients has been shown to have a research gap, particularly in light of the large number of instances that call for an automated approach for prompt diagnosis.

1.1 Motivation for the Work

The primary motivation behind this research is to develop a predictive model for Covid-19 and pneumonia occurrence. Identifying the best classification method to forecast a patient's risk of developing these illnesses is another goal of the project. Even if deep learning algorithms are frequently employed for these kinds of tasks, it is still crucial to maximize accuracy when predicting Covid-19 and pneumonia.This research endeavor seeks to empower researchers and medical practitioners to enhance diagnostic capabilities and patient care.

1.2 Problem Statement

Detecting Covid-19 and pneumonia poses a significant challenge due to the limitations of available instruments, which are either expensive or lack efficiency in accurately assessing the likelihood of these diseases in individuals. In order to reduce death rates and the problems associated with these disorders, early detection is important.

The main contributions of our research are as follows:

- Developed a deep learning methodology using the Convolutional Channel Spatial Attention Feature Extract Model (CCSAFEM) for accurate detection of Covid-19 and Pneumonia in chest CT scans.
- Integrated an attention mechanism into features extracted from ResNet 50, with hyper parameter tuning to enhance performance.

This paper is organized into a number of sub-sections: Section 2 provides an extensive discussion on deep learning models for Covid-19 and pneumonia prediction. The overview of the data, together with the number of attributes and their descriptions, is explained in Section 3. Section 4 includes the results and analysis. Lastly, Section 5 gives the conclusion of the paper.

2. Related Work

In the field of medical data processing, data scientists play a vital role in advancing medical sciences through their significant contributions. They employ deep learning, machine learning, and data extraction techniques extensively to extract specific features from image datasets, facilitating disease diagnosis and prediction. As shown in table 1.

Table 1. Existing Literature Review

Author & Reference	Year	Method Used	Description
Wang et al. [2]	2020	CNN	Radiographs of pneumonia patients along with CT images of Covid19-positive patients are fed into deep learning algorithm architecture.

Xiaowei et al [3]	2020	Pulmonary CT scans and several CNN	Several CNN models and pulmonary CT scans are used in this model's execution.
Netzahualcoyotl Hernandez-Cruz [4]	2021	Neural style transfer	It shows the cycle-generative adversarial network's effectiveness, which is pervasive in neural style transfer.
Shah Siddiqui et al [5]	2022	CNN	The purpose of classifying and identifying Covid-19 in X-ray and CT pictures, CNNs are the most widely utilised deep learning architecture.
Rajit Nair et al [6]	2021	ResNet 50	Carried out a multicenter, retrospective research to identify Covid-19 by extracting visual features from volumetric chest CT scans.
Mohammad Farukh Hashmi et al [7]	2021	ResNet50	The suggested model has the potential to assist in disease detection and support radiologists in their clinical decision-making.
Zhenjia Yue[8]	2020	CNN	In terms of image identification, CNNs have outperformed humans, and artificial intelligence recognition is highly quick.
Aakash Shah [9]	2022	CNN	Using chest CT and X-ray images to distinguish between Covid-19 and pneumonia.
Puneet Gupta [10]	2021	CNN	The capabilities of CNN models that have already been trained, which are used as feature extractors and then classified diverse chest X-rays into abnormal and normal categories.
Avnish Panwar[11]	2021	CNN	Several machine learning classifiers were trained to classify chest radiographs as either normal, pneumonia, or Covid-19 positive.
Waleed Al Shehri [12]	2022	CNN	Processed X-ray and CT scan pictures using Darknet and pre-implemented convolutional neural networks.

3. Materials and methods

Compared to smaller datasets, deep learning consistently performs better when trained on larger ones. Despite the prevalence of Covid-19 and pneumonia, there are very few publicly available chest X-ray images. In order to tackle this problem, researchers have assembled a sizable dataset of chest X-ray images [13], which they have complemented with easily accessible normal and pneumonia images for training and testing examination. The proposed deep learning model and the datasets used for predictive analysis are the two main features of this research.

3.1 Dataset Collection

The database included normal chest X-rays as well as images of Covid-19 and pneumonia [15]. The popular Kaggle chest X-ray database has 5,160 resolutions ranging from 800 to 1900 pixels, showing both bacterial and viral pneumonia as well as normal or healthy instances. This source provided chest X-ray pictures for Covid-19, normal, and pneumonia cases, which were used in the creation of the most recent database collection [13, 14]. The database consists of 5,160 images categorized into three classes, with a batch size set at 20. The training, validation, and testing data for the database collection are graphically shown the following figure 1.

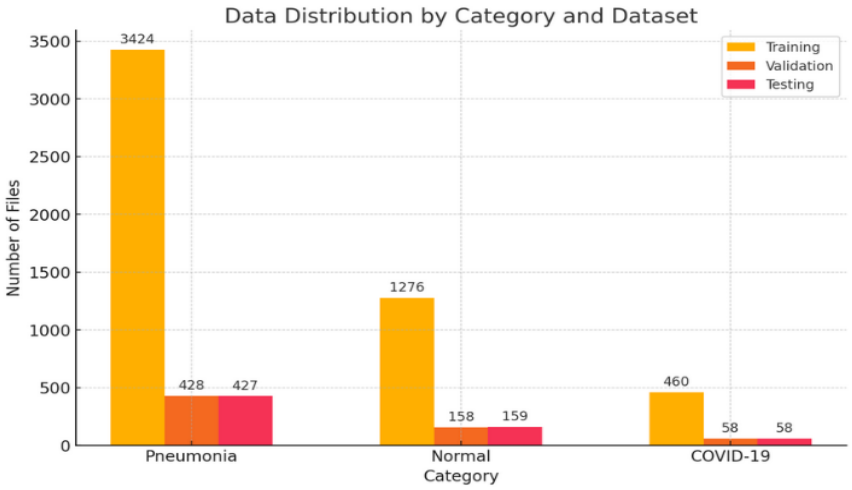


Fig.1. Graphically Represent Dataset

The training directory contains 3,424 files in the Pneumonia category, 1,276 files in the Normal category, and 460 files in the Covid-19 category. In the validation directory, there are 428 files for Pneumonia, 158 files for Normal, and 58 files for Covid-19. The testing directory includes 427 files for Pneumonia, 159 files for Normal, and 58 files for Covid-19. With a batch size of 20, a total of 5,160 images across three classes were found.

3.2 Features processing

In the features processing various techniques such as grouping, average pooling (avgpool), maximum pooling (maxpool), Group Normalization (GroupNorm) [16], and other procedures to tackle the

aforementioned challenges. The concept of grouping was initially introduced by AlexNet [17] and later evolved into modern grouping convolution. Convolutional grouping divides the feature map F into G groups, unlike conventional convolution, where the weight term is considered as C_{in} ×k×k×C_{out} for an input feature map F ϵ [C/G, H, W]. There are G feature maps in total as each group creates its own feature map. All these categories make use of a uniform convolution kernel with size k×k×C/G, and the size of each group is represented by F ϵ [C/G, H, W]. Parameter number for grouped convolution is then given by $C_{in}\times k\times k\times C_{out}/G$. Our model employs the concept of grouping to divide the feature map, effectively reducing extra parameters. Downsampling is achieved through pooling, which helps retain overall feature information while reducing calculation parameters and speeding up computation, particularly beneficial for handling large feature maps. Given the immense input image data in these tasks, big batches often fail to perform effectively, leading to batch sizes typically ranging from 1 to 4 in deep learning applications. GroupNorm [16] serves as a normalization technique, differing from BN (batch normalization). Under significant batch sizes, GroupNorm exhibits a similar error rate to BN, but it outperforms BN by approximately 10% under small batch sizes, notably enhancing the model's detection accuracy. Moreover, transmitting feature map information among different groups is essential. Without shared information, the network's feature extraction capacity becomes restricted, hindering the desired outcomes. To enable feature communication within each group, we employ the channel shuffle [18] operation, reorganizing features to facilitate information flow between groups.

3.3 Attention Mechanism

Attention models have been used extensively in most deep learning problems in recent years. Speech recognition, image processing, natural language processing, and other applications all make use of the visual attention mechanism. It shows a special signal processing technique that exists in human vision. In order to obtain extensive knowledge while eliminating unnecessary data, the human brain quickly analyses complete images, selecting key spots or areas of interest. It then devotes more attentional resources to these areas. By compressing features using a squeeze operation and then restoring them to their original dimensions using two fully linked layers, SENet [19] is able to capture channel-wise relationships. By adaptively setting the size of a 1-D convolution field to estimate local cross-channel interactions, ECANet [20] improves on this method. While CCNet [22] lowers processing costs by restricting calculations to the same rows and columns, Non-Local Neural Networks address remote dependencies by estimating interactions between any two points in the image[21]. Channel and spatial attention processes are combined in CBAM [23] and GCNet [24] to achieve significant gains.

3.4 Proposed Model-Convolutional Channel Spatial Attention Feature Extract Model (CCSAFEM)

A CNN is composed of by several smaller units organised in a layered design, called nodes or neurones. During model training, these nodes' weights are modified using optimisation strategies like backpropagation. Convolutional (feature extraction) and classification are two fundamental components of any CNN model. This research aims to utilize chest X-ray images in developing a model for predicting Covid-19 and pneumonia. This is largely attributed to the utilization of various layers and numerous neurons within the model to process different components of the image. Such

layer and neuron configurations are instrumental in achieving improved classification results. It effectively captures representative features from both local and 3D dimensions. The CCSAFEM system comprises a backbone RestNet50 [25], which processes input from multiple CT trunks and facilitates their respective functionalities. Extracted features are then amalgamated through a pooling process across all slices. Subsequently, a fully connected layer with softmax activation is employed for each class in the final feature map. For preprocessing and lung region removal, a unit-based segmentation method is utilized due to the focal area of concern in 3D CT scans [26]. The pre-processed image is then fed into the CCSAFEM for forecasting. In this process, a ResNet-50 model serves as the backbone, employing the Max Pooling approach with shared weights. The proposed CCSAFEM conducts multiclass classification, distinguishing between Normal, Covid-19, and pneumonia cases, as illustrated in Figure 2. This architecture represents deep learning model architecture for classifying medical images into the categories of Covid-19, Pneumonia and Normal, using a combination of ResNet-50, attention mechanisms, and deep learning layers.

3.5 Convolutional Channel Spatial Attention Feature Extract Model (CCSAFEM)

CSAFM first draws out world features X from intermediate feature map $\boldsymbol{F \epsilon R^{C\times H\times W}}$, where C refers to the amount of channels and H and W represent the spatial height and width, respectively, in order to have optimized feature representation and extraction. By combining the sub-feature Xk with the global feature information, this phase seeks to improve the quality of the extracted features. Consequently, it enables improved exchange and circulation of information when channel and spatial attention are fused. In particular, $\boldsymbol{X_{k1}, X_{k2} \epsilon R^{C/2G\times H\times W}}$, The importance coefficients for each sub-feature are then calculated using an attention module. After combining all of the sub-features, additional processing is done to allow information to flow across groups along the channel dimension. A thorough explanation of each attention module is given in the sections that follow.

3.5.1 Channel Attention Module

In simulating how individuals choose the objects to attend to, the attention mechanism is inspired by the human visual attention system. The primary goal of channel attention is to pinpoint an input image's most important features. According to CBAM [23], combining max-pooled and average-pooled features yields a much more potent network representation than applying each technique separately. This is supported by empirical data. In order to extract global information from a feature map, we first create two different descriptors: X^C_{avg} and X^C_{max} by using the average pooling and max pooling approaches. A channel attention map is then created by processing these descriptors via a scale network; this is shown as Mc ϵ $R^{C/2G\times 1\times 1.}$ In order to facilitate element-wise summing with sub-features, this map is used to restructure x. The average pooling and max pooling methods are then applied to each branch for every sub-feature x_k. The figure 3 diagram represents a channel attention mechanism typically used in CNNs for enhancing the feature representation by focusing on important channels.

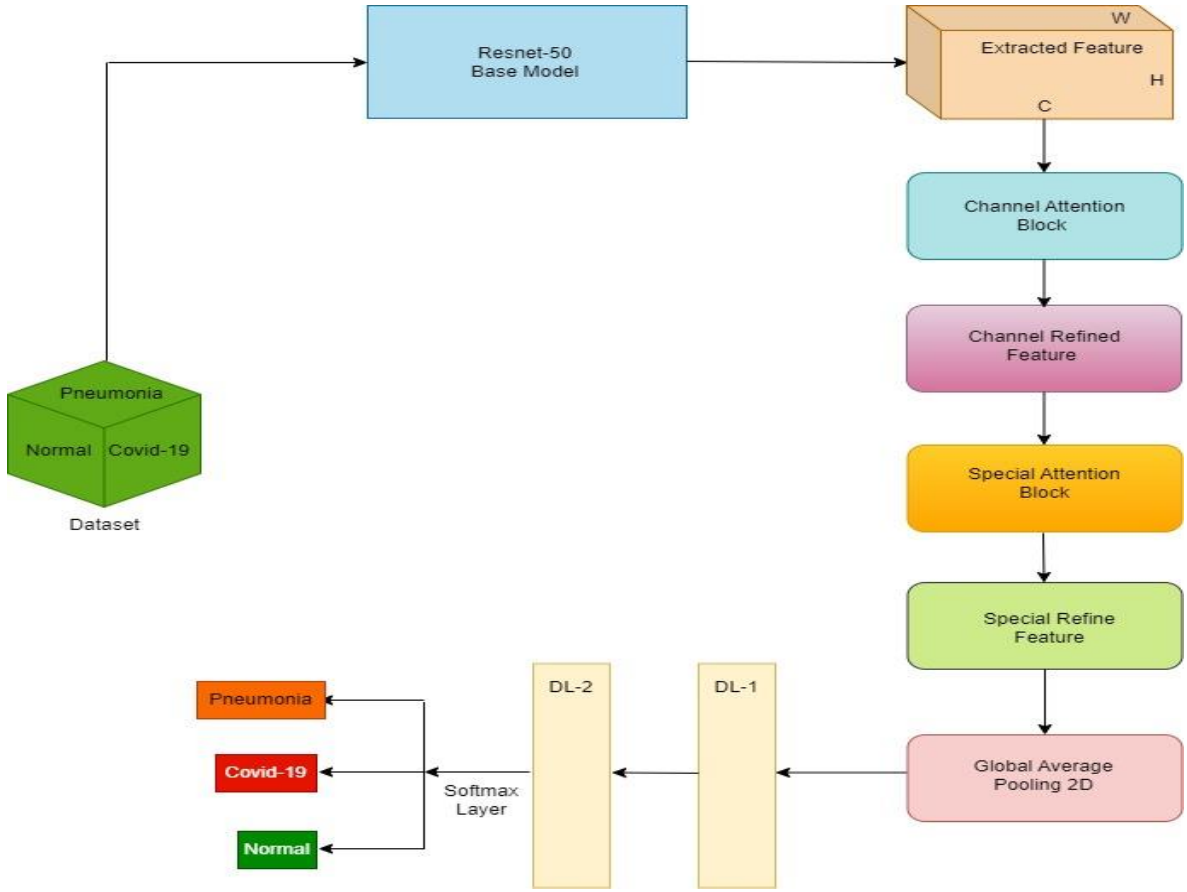


Fig. 2. Architecture of CCSAFEM

3.5.2 Spatial Attention Module

Spatial attention focusses on specific regions of the feature map, identifying the locations of important information. In contrast, channel attention emphasizes the important features within the feature map, highlighting the most significant aspects. To compute spatial attention, we use Group Normalization (gn) [16] across two branches, x_{k1} and x_{k2}, reducing the computational burden.

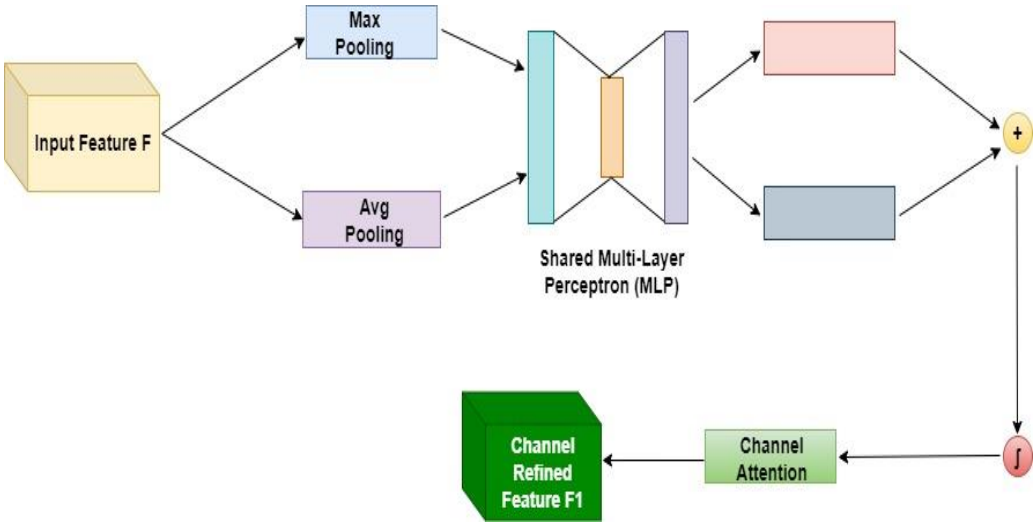


Fig.3. Architecture of Channel Attention Module

The two branches are combined to match the number of channels with the inputs after these parameters have been defined. Once fused, all sub-features are combined. We use an operator called "shuffle" to allow the interchange of cross-group information along the channel dimension [27]. The feature X and the CSAFM module's output are the same size. We mostly use the faster R-CNN [28] architecture as the detector in our experiments, using frameworks ResNet50. The figure 4 illustrates a **spatial attention mechanism** that is applied to the channel-refined feature.

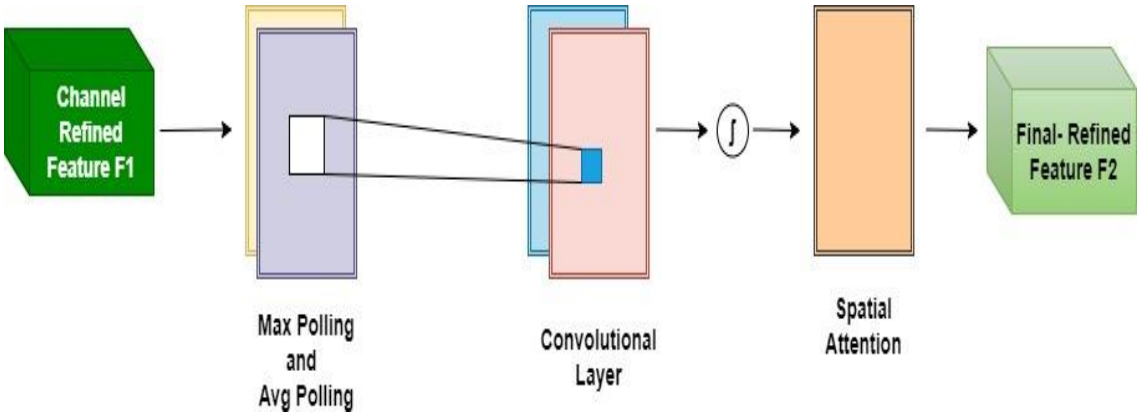


Fig.4. Architecture of Spatial Attention Module

This spatial attention mechanism allows the network to concentrate on both important spatial regions and crucial channels inside each feature map, which improves on the prior channel attention method. The network performs better because to this dual attention approach.

3.6 Process Diagram

This figure 5, clearly illustrates each stage of the procedure, from parameter definition to experimental setup. A process flow is indicated by arrows, and each box indicates a crucial phase. Understanding how each component of the system contributes to its overall performance is made easier with the help of this format.

3.7 Algorithms of the Proposed Model

The proposed model's algorithms encompass a series of computational procedures designed to achieve specific tasks. These algorithms are meticulously crafted to address the objectives outlined within the model's framework.

Algorithm 1: Channel_Attention (Input: F_C = H x W x C)

{

Channel ← F_C. get_shape ( ) [-1]

Avg_pool← gloval_avg_pooling ( ) (F_C)

Max_pool← gloval_max_pooling 3D ( ) (F_C)

F_{CA}=Add ( ) [avg_pool + max_pool]

Return (F_{CA})

}

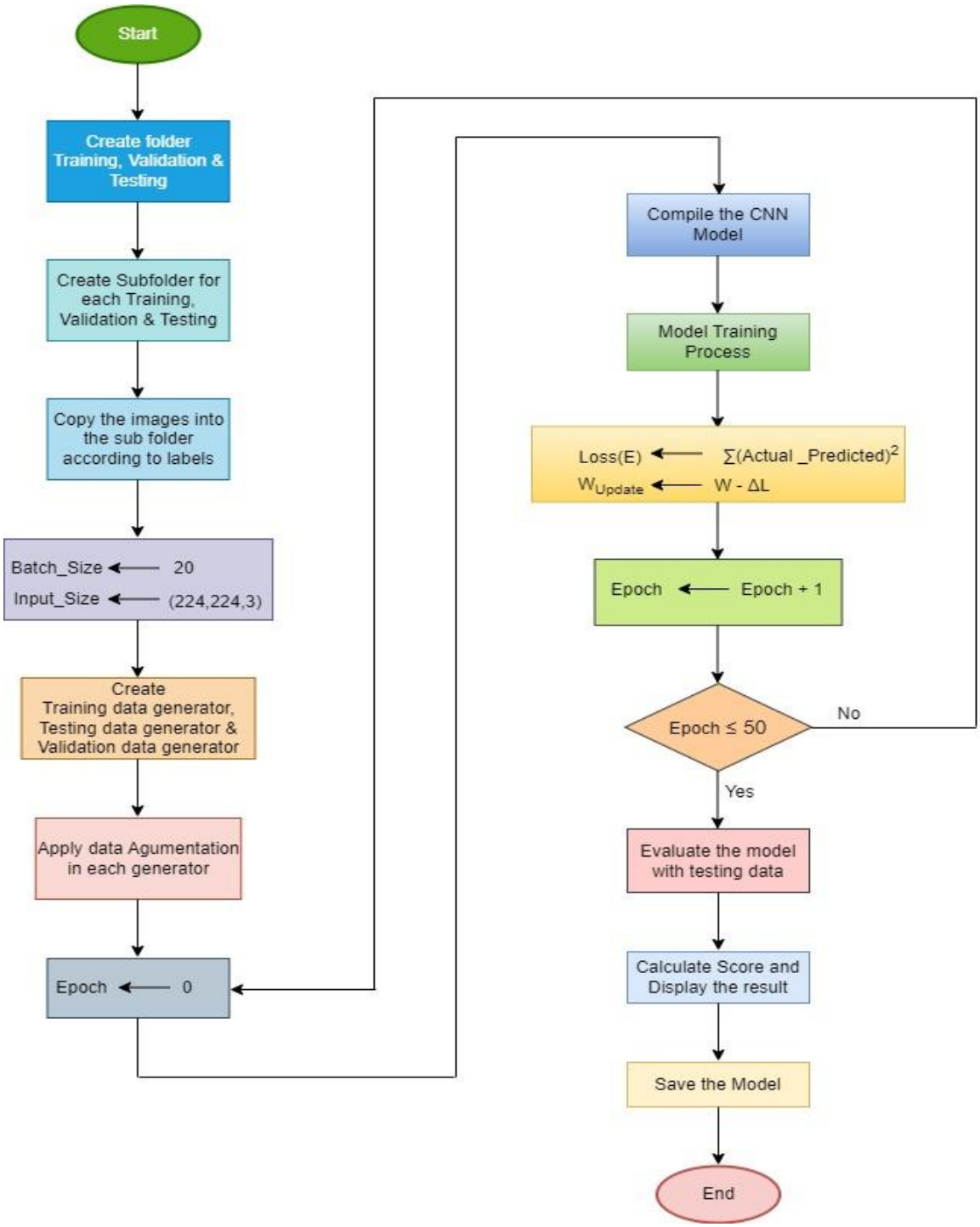


Fig.5. Architecture of Process diagram

Algorithm1, Channel_Attention, is designed to compute the channel attention vector for a given feature map FC. The global average pooling and global max pooling, which reflect the average and maximum activation values across each channel, respectively. Different facets of channel-wise information are captured by these two pooling procedures. The channel attention vector is then

created by combining the outcomes of each channel. The meaning of every channel in the feature map is captured in this vector.

Algorithm 2: Special _attention (Input: $F_{C'}$)

{

avg_pool← avg_pool ($F_{C'}$)

max_pool← max_pool ($F_{C'}$)

$C_{concatenate}$ ← Concatenate ( ) [avg_pool, max_pool]

$F_{Resultant}$ ← Conv 3D ($F_{concatenate}$)

Return ($F_{Resultant}$)

}

Conv 3D (filler = 1, kernel_size = 7x7, Stride = 1, Padding = ‘same’, activation = ‘sigmoid’)

The Spatial_Attention algorithm 2 is intended to calculate spatial attention for an input feature map FC'. The max pooling and average pooling FC' of the feature map are first calculated. These pooling extract spatial information by calculating the average and maximum activation values in each spatial region. Max pooling and average pooling outputs are concatenated along the channel dimension. This concatenated feature map is then subjected to a 3D convolution operation with a kernel size of 7x7, a stride of 1, and 'same' padding. A sigmoid activation function, which limits the output values between 0 and 1, is used to represent the attention weights for various spatial regions following the convolution operation. The resulting feature map with spatial attention information is finally returned.

Algorithm 3:CNN_model_creation (Input_shape, num_classes = 3)

{

Base_model ← ResNet 50 (Input_shape, include_top = ‘False’)

$F_{ResNet\ 50}$ ← Base_model. Output

F_{CA} ← Channel _attention ($F_{ResNet\ 50}$)

$F_{Resultant}$ ← Special_attention (F_{CA})

X ← Layer.Dense1 (64) ($F_{Resultant}$)

X ← Layer.Dense2 (32) (X)

Prediction ← Layer.Dense (3, activation = ‘softmax’) (X)

Model ← Model (Input_shape, Prediction)

Return (Model)

}

The algorithm 3, CNN_model_creation, outlines the procedure for creating a CNN model for classification tasks. Firstly, the base ResNet50 model is instantiated with specified input shape and without the top classification layers. Then, the output feature map from ResNet50 is processed through the Channel_Attention algorithm to compute channel attention. In order to incorporate spatial attention, the feature map with channel attention is then subjected to the Spatial_Attention algorithm. For additional feature extraction and dimensionality reduction, the resultant feature map is subsequently processed through completely connected layers. The model is constructed using the input shape and the final prediction layer, and a dense layer with softmax activation is added for classification.

Algorithm 4: Training & Testing

{

Input: Training, Validation data

epoch ← 50, batch ← 20

For epoch 1 to 50:

    For each batch in training dataset

    Feature _extraction (Model)

Back_Propogation, calculate the gradient ΔL

Update the model parameter W: W-α ΔL

// α← Learning Rate

Save the Trained model

Model Testing

Initialize the model & load the parameter

Test_ Prediction ← model (test_data)

Return (test_prediction)

}

Algorithm 4 illustrates the training and testing of a CNN—a deep learning model—for classification. The model is trained initially on batches of the training data by going through a set number of epochs. For each batch, features are extracted using the previously constructed CNN model. The derivative of the loss function with respect to model parameters is calculated by backpropagation. Gradient descent is employed to update these parameters, the learning rate controlling the step size of the updates. After training, the model is stored for further use. The stored model is loaded with its pre-trained parameters for assessment during testing. Inference is then performed on the test dataset to generate predictions. Finally, the algorithm returns the predictions made by the model on the test data.

4. Results analysis

The primary programming language utilized in the experiment was Python 3.6, which was conducted on a 64-bit version of Windows 10. Keras was the backend for the TensorFlow framework, and it was utilized for model construction as well as training. The implementation was done on a machine with an Intel® Core™ i7-8700 CPU at 4.60GHz, a 12MB cache, and 16GB of RAM. The assessment of the ResNet50 CNN model was conducted using a batch size of 20 and over 100 epochs. The evaluation metrics and performance results were recorded accordingly. The plot illustrates the iterative steps involved in feature extraction from the images during model training. A remarkable accuracy of 98.70% is shown by the results.

4.1 X-ray image after Training and Testing

A grid of chest X-ray pictures classified as Covid-19, pneumonia, and normal is displayed in the image. The training and testing procedure of the suggested model for grouping chest X-rays into these three categories is depicted in figure 6, which also shows the X-ray picture analysis. Firstly, the X-rays labeled Normal show healthy lungs without any signs of infection. Secondly, the X-rays labeled Pneumonia show signs of lung infection, characterized by the presence of fluid or inflammation. Finally, the X-rays labeled Covid likely show the typical patterns of lung involvement seen in Covid-19 patients.

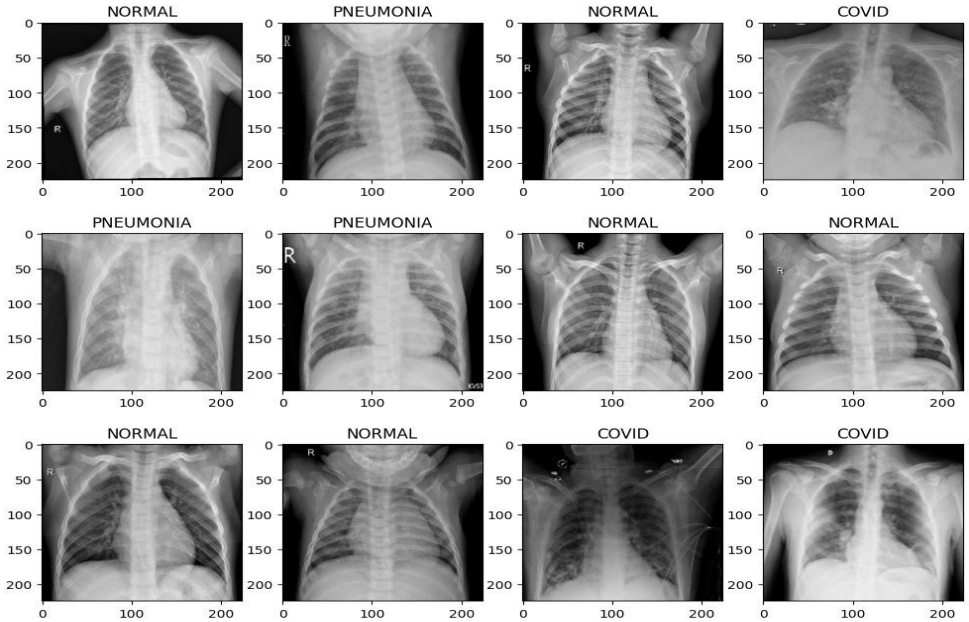
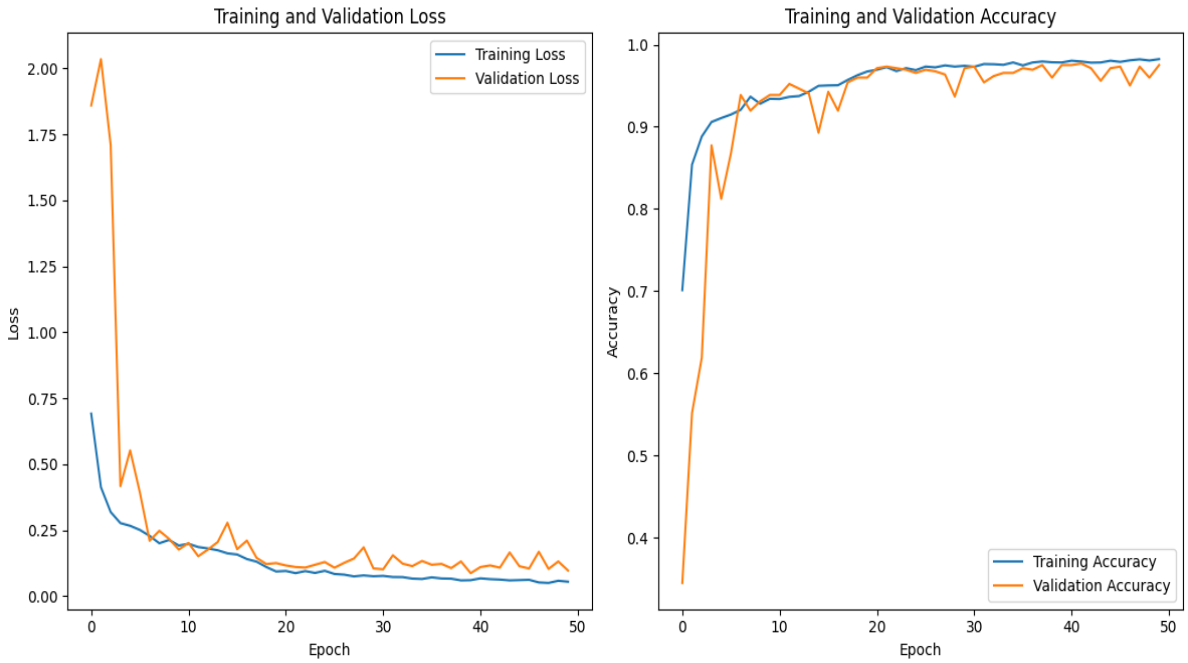


Fig.6. Chest X-rays images analysis

4.2 Performance of model over 10, 30 & 50 epochs

Training and validation metrics are compared in the graph. Whereas figure 7(a) presents the training and validation loss, figure 7(b) presents the training and validation accuracy. As summarized in table 2, the data indicates the results of training a deep learning model for 10, 30, and 50 epochs, along with corresponding metrics for training accuracy, validation accuracy, training loss, and validation loss.



(a) (b)

Fig.7. (a) and (b) Shows two graphs comparing training and validation metrics over 10, 30, & 50 epochs

Table 2. Represents the results of training of deep learning model over 10, 30, and 50 epochs

Parameters	**10 Epochs**	**30 Epochs**	**50 Epochs**
Training Loss	0.25	0.15	0.10
Validation Loss	0.25	0.17	0.15
Training Accuracy	0.91	0.95	0.987
Validation Accuracy	0.93	0.94	0.97

4.3 Classification Report

The table 3 appears to be an evaluation report for a classification model, possibly for predicting Covid, Normal, and Pneumonia classes.

Table 3. Evaluation report of classification model

	precision	**recall**	**f1-score**	**Support**
Covid-19	1.00	0.98	0.99	163
Normal	0.99	0.96	0.97	180
Pnemonia	0.95	0.99	0.97	180
Accuracy			0.98	523
Macro avg	0.98	0.98	0.98	523
Weighted avg	0.98	0.98	0.98	523

4.4 Confusion Matrix

A confusion matrix showing a classification model's performance across three categories Covid, Normal, and Pneumonia is shown in figure 8. The projected labels Covid, Normal, and Pneumonia are shown on the x- and y-axes. The values in table 4 are as follows:

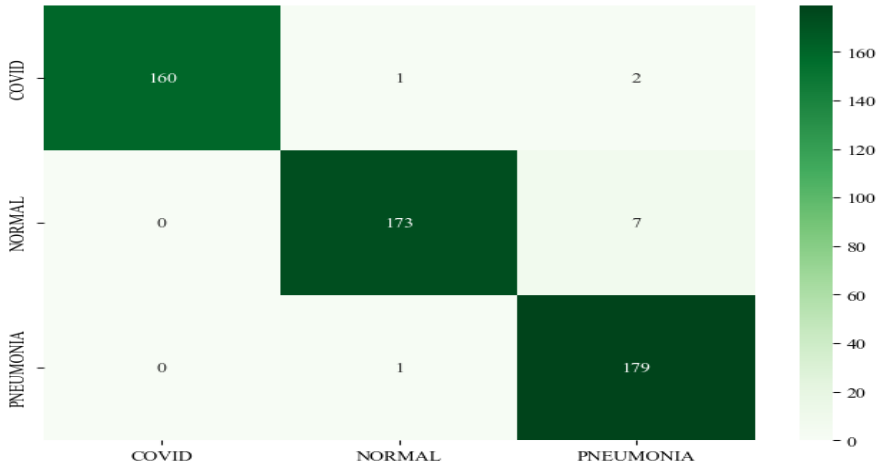


Fig .8. Visualizing the Performance of a classification model

Table 4. Predicted Labels

Sl no	labels	Covid-19	Normal	Pneumonia
1.	Predicted	Covid: 160	Covid: 0	Covid: 0
2.	Predicted	Normal: 1	Normal: 173	Normal: 1
3	Predicted	Pneumonia: 2	Pneumonia: 7	Pneumonia: 179

4.5 Sensitivity and Specificity report

The table 5 describes the sensitivity (recall) and specificity of a classification model for each class.

Table 5. Describes the sensitivity and specificity

Class	Sensitivity (Recall)	Specificity
Covid	0.98	1.00
Normal	0.96	0.99
Pneumonia	0.99	0.97

(a) Sensitivity: The ratio of actual positive cases correctly identified by the model is indicated by this measure: 98% of all actual Covid-19 cases are rightly identified by the model. This model also correctly classifies 96% of all true normal instances and classifies 99% of all cases of actual pneumonia with AUCs of 1.00; this outstanding performance reflects almost perfect classification.

(b) Specificity: This measure indicates the percentage of actual negative cases that the model correctly labeled: 100% of all non-Covid-19 cases are correctly classified by the model. This model provides 99% of the total cases not Normal are appropriately detected by the model. The model correctly identifies 97% of all non-pneumonia diagnoses.

4.6 Comparative Result of Different Models

Comparison of various deep learning models in detecting diseases such as Covid-19 and pneumonia is presented in table 6. Accuracy and loss are employed to measure the performance of the models during training and validation. The suggested model, ResNet50, Inception V3, ResNet-18, ResNet50 coupled with SVM, and all other models were run for 50 epochs in order to assess validation metrics and accuracy. The proposed approach performed better than the other models, according to the results shown in table 6.

Table 6. Comparative Result of Different Models for Covid-19 and Pneumonia

Model	**Accuracy (Train)**	**Loss (Train)**	**Accuracy (Val)**	**Loss (Val)**
ResNet50	96.87	0.0310	96.60	0.0511
Inception V3	97.36	0.0412	96.30	0.0701
ResNet-18	97.61	0.0311	97.51	0.1254
ResNet50 & SVM	97.80	0.0412	96.98	0.0131
Proposed Method	98.10	0.0415	97.91	0.1217

4.7 Comparative analysis

The suggested framework is evaluated against other models. A comparison of the suggested and current methods is given in Table 7.

Table 7. Comparative Study of existing methods with proposed method

Sl.No	Author and References	Methodology	Overall Test accuracy
01	Mohammad Farukh Hashmi et.al [07]	ResNet50	98.14
02	Avnish Panwar et.al [11]	Inception V3	88.80
03	Xuan CAI et.al [29]	ResNet-18	94.30
04	Vamsidhar Enireddy et.al [30]	ResNet50 and SVM	94.00
05	**Proposed Method**	**CCSAFEM**	**98.70**

5. Conclusion

As the results of this research indicate, the ResNet50 CNN model has demonstrated encouraging performance in separating Covid-19 and pneumonia from X-ray images. In addition to predicting with accuracy the presence or absence of Covid-19 and pneumonia, the CNN model developed in this research extracts information from X-ray images effectively. The methods designed in this research enable more accurate Covid-19 and pneumonia prediction, with a staggering 98.70% accuracy rate. CNN models can be improved even further in the future by trying out various transfer learning methods and tuning hyperparameters.

References

[1]. Stephen, O., Sain, M., Maduh, U. J., & Jeong, D. U, "An efficient deep learning approaches to pneumonia classification in healthcare", Journal of healthcare engineering, (2019).

[2]. Shuai Wang, Bo Kang, Jinlu Ma, Xianjun Zeng, Mingming Xiao, Jia Guo, Mengjiao Cai et al, " A deep learning algorithm using CT images to screen for Corona Virus Disease (COVID-19)", MedRxiv (2020).

[3].Xiaowei Xu, Xiangao Jiang, Chunlian Ma, Peng Du, Xukun Li, Shuangzhi Lv, Liang Yu et al, "A deep learning system to screen novel coronavirus disease 2019 pneumonia", Engineering 6, no. 10 (2020).

[4]. Netzahualcoyotl Hernandez-Cruz, David Cato, Jesus Favela, "Neural Style Transfer as Data Augmentation for Improving COVID-19 Diagnosis Classification", SN Computer Science (2021). https://doi.org/10.1007/s42979-021-00795-2

[5]. Shah Siddiqui et al.: Deep Learning Models for the Diagnosis and Screening of COVID-19, "A Systematic Review", SN Computer Science (2022). https://doi.org/10.1007/s42979-022-01326-3

[6]. Rajit Nair et al, "Deep learning-based COVID-19 detection system using pulmonary CT scans", Turkish Journal of Electrical Engineering and Computer Sciences: Vol. 29: No. 8, Article 9, (2021). https://doi.org/10.3906/elk-2105-243

[7]. Mohammad Farukh Hashmi et al, “Pneumonia detection in chest X-ray images using compound scaled deep learning model”, Automatika, 62:3-4, 397-406, (2021). https://doi.org/ 10.1080/00051144.2021.1973297

[8]. Zhenjia Yue, “Comparison and Validation of Deep Learning Models for the Diagnosis of Pneumonia”, Hindawi Computational Intelligence and Neuroscience Volume, (2020). https://doi.org/10.1155/2020/8876798

[9]. Aakash Shah, Manan Shah, “Advancement of deep learning in pneumonia/Covid19 classification and localization: A systematic review with qualitative and quantitative analysis”, Chronic Diseases and Translational Medicine, (2022). https://doi.org/ 10.1002/cdt3.17

[10]. Puneet Gupta, “Pneumonia Detection Using Convolutional Neural Networks”, International Journal for Modern Trends in Science and Technology, (2021).

https://doi.org/10.46501/IJMTST070117

[11]. Avnish Panwar et al, “Deep Learning Techniques for the Real Time Detection of Covid19 and Pneumonia using Chest Radiographs”, IEEE, (2021).

https://doi.org/ 10.1109/EUROCON52738.2021.9535604

[12]. Waleed Al Shehri, “A Novel COVID-19 Detection Technique Using Deep Learning Based Approaches", MDPI, Sustainability, (2022). https://doi.org/10.3390/su141912222

[13]. J. C. Monteral. “COVID-Chestxray Database Available”, (2020).

https://github.com/ieee8023/covid-chestxray-dataset

[14]. D. Demner-Fushman, M.D. Kohli, M.B. Rosenman, S.E. Shooshan, L. Rodriguez, S. Antani, G.R. Thoma, and C.J. McDonald, “Preparing a collection of radiology examinations for distribution and retrieval”, J Am Med Inform Assoc, (2016). https://doi : 10.1093/jamia/ocv080. Epub 2015 Jul 1

[15]. Alqudah, Ali Mohammad; Qazan, Shoroq, “Augmented COVID-19 X-ray Images Dataset”, Mendeley Data, (2020). http://dx.doi.org/10.17632/2fxz4px6d8.4

[16]. Wu Y, He K, “Group normalization”, Proceedings of the European conference on computer vision (ECCV), (2018).

[17]. Krizhevsky A, Sutskever I, Hinton G E, “Imagenet classification with deep convolutional neural networks ”, Advances in neural information processing systems, 2012.

[18]. Zhang X, Zhou X, Lin M, et al, “Shufflenet: An extremely efficient convolutional neural network for mobile devices”, Proceedings of the IEEE conference on computer vision and pattern recognition, (2018).

[19]. Hu J, Shen L, Sun G, “Squeeze-and-excitation networks”, Proceedings of the IEEE conference on computer vision and pattern recognition, (2018).

[20]. Qilong Wang, Banggu Wu, Pengfei Zhu, Peihua Li,Wangmeng Zuo, and Qinghua Hu, "Eca-net: Efficient channel attention for deep convolutional neural networks", IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR (2020).

[21]. Wang X, Girshick R, Gupta A, "Non-local neural networks", Proceedings of the IEEE conference on computer vision and pattern recognition. (2018).

[22]. Huang Z, Wang X, Huang L, "Criss-cross attention for semantic segmentation", Proceedings of the IEEE/CVF international conference on computer vision, (2019).

[23]. Woo S, Park J, Lee J Y, "Convolutional block attention module", Proceedings of the European conference on computer vision (ECCV), (2018).

[24]. Cao Y, Xu J, Lin S, "Non-local networks meet squeeze-excitation networks and beyond", Proceedings of the IEEE/CVF international conference on computer vision workshops (2019).

[25]. Liu W, Anguelov D, Erhan D, "Single shot multibox detector", European conference on computer vision., Springer, Cham, (2016).

[26]. Lin T Y, Goyal P, Girshick R, "Focal loss for dense object detection", Proceedings of the IEEE international conference on computer vision, (2017)

[27]. Ma N, Zhang X, Zheng H T, "Practical guidelines for efficient cnn architecture design", Proceedings of the European conference on computer vision (ECCV), (2018).

[28]. Ren S, He K, Girshick R, "Towards real-time object detection with region proposal networks", Advances in neural information processing systems, (2015).

[29]. Xuan Cai, Yifan Wang, Xiaoqing Sun, Wentao Liu, Yuyang Tang, WenBo Li, "Comparing the performance of ResNets on COVID-19 diagnosis using CT scans", IEEE, (2020).

[30]. Vamsidhar Enireddy, Mathe John Kenny Kumar, Babitha Donepudi, C Karthikeyan, "Detection of COVID-19 using Hybrid ResNet and SVM", IOP Conf. Series, Materials Science and Engineering 993, (2020). http://doi:10.1088/1757-899X/993/1/012046