DETECTION OF COVID-19 USING TRANSFER LEARNING

A DISSERTATION

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE
OF

MASTER IN TECHNOLOGY
IN
(INFORMATION SYSTEMS)

Submitted by
SHIKHAR SAXENA
2K19/ISY/15

Under the supervision of
Prof. Kapil Sharma

DEPARTMENT OF INFORMATION TECHNOLOGY
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)

JULY-2021

M.TECH (INFORMATION SYSTEMS) SHIKHAR SAXENA 2021

DELHI TECHNOLOGICAL UNIVERSITY
(FORMERLY DELHI COLLEGE OF ENGINEERING)

CANDIDATE'S DECLARATION

I**, Shikhar Saxena** (2K19/ISY/15), student of M.Tech (ISY), hereby declare that the Project Dissertation titled "**Detection of COVID-19 using Transfer Learning**" is submitted by me under the supervision of **Prof. Kapil Sharma** to Department of Information Technology, DELHI TECHNOLOGICAL UNIVERSITY (formerly Delhi College of Engineering) in partial fulfillment of the requirements for the award of the degree of **MASTER OF TECHNOLOGY** is original and not copied from any source without proper citation**.**This work has not previously formed the basis of award of any degree or diploma. The work reported in this has not been submitted by me for the award of any other degree or diploma.

Shikhar

Place: Lucknow
Date: 01st July 2021

Shikhar Saxena (2K19/ISY/15)
Department of Information Technology
Delhi Technological University
New Delhi

DEPARTMENT OF INFORMATION TECHNOLOGY
DELHI TECHNOLOGICAL UNIVERSITY
(FORMERLY DELHI COLLEGE OF ENGINEERING)

CERTIFICATE

I hereby certify that the project dissertation titled "**Detection of COVID-19 using Transfer Learning"** submitted by **Shikhar Saxena** (2K19/ISY/15), student of M.Tech (ISY), which has been submitted to Department of Information Technology, DELHI TECHNOLOGICAL UNIVERSITY (formerly Delhi College of Engineering) in partial fulfillment of the requirements for the award of degree of MASTER OF TECHNOLOGY in Information Systems, is an original contribution with existing knowledge and faithful record of work carried out by him under my guidance and supervision.

To the best of my knowledge this work has not been submitted in part or full for any Degree of Diploma to this University or elsewhere.

Place: New Delhi
Date: 01st July 2021

Prof. Kapil Sharma
Head of Department
Information Technology
Delhi Technological University

ACKNOWLEDGEMENT

I am very thankful to **Prof. Kapil Sharma** (Head of Department, Information Technology) and all the faculty members of the Department of Information Technology of DTU. They all provided us with immense support and guidance for the project. My grateful thanks are extended to Delhi Technological University for providing us with the laboratories, infrastructure, testing facilities and environment which allowed us to work without any obstructions. I would also like to appreciate the support provided to us by our lab assistants, seniors and our peer group who aided us with all the knowledge they had regarding various topics.

Place: Lucknow
Date: 01st July 2021

Shikhar Saxena (2K19/ISY/15)
Department of Information Technology
Delhi Technological University

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Abstract

Coronavirus disease-19 spread has grown to almost every corner of the globe. As a result, it is necessary to take steps toward a much earlier diagnosis of its infection.

Coronavirus Illness (COVID-19) is a recently identified coronavirus infectious disease. In most COVID-19 infected persons, respiratory illnesses are mild to severe. without particular treatment they recover. The risk of serious sickness is increased among elderly and those with underlying medical conditions, such as cardiovascular disease, diabetes, chronic respiratory disease and cancer. The greatest method to avoid and delay transmission is to know the COVID-19 virus, the sickness it produces and how it is spreading well. The only protection from it's illness via washing your hands or regular use of alcohol-based rubber without touching your face.

Chest X-Rays, Computed Tomography, and RT-PCR are early-stage diagnostic techniques.Visually detecting and inspecting these clinical images for any hidden anomalies is a time-consuming task. Serology is used to identify anticorps in clinical settings and population surveillance. Due to the restricted availability of test kits every person afflicted by the virus is difficult to detect.In addition, these tests take from several hours to a day to generate the result, which in the current situation of urgency becomes excessively tedious, time-consuming and mostly mistake prone. A quicker and more accurate screening approach is therefore urgently needed, which may also be validated by the PCR test. Transfer Learning in medical imaging has a lot of research potential.

The method proposed here is a transfer learning-based binary classification model which predicts whether a Lung CT image has SARS-CoV-2 infection. It has a three-stage procedure for fine-tuning various pre-trained architectures. It uses progressive resizing an optimization technique in which our approach is to resize the input images to 128×128×3, 150×150×3, and 224×224×3 pixels and fine-tuning the neural network at each stage. As a result, CT transfer learning with progressive resizing outperforms various published models in the recent research work with improved accuracy of 97.4% with only 22 epochs. This technique may help to diagnose COVID-19 patients at an early stage and reduce the pressure on medical systems.

Keywords—Transfer Learning, COVID-19, progressive resizing, Radiology, Computed Tomography.

CHAPTER 1 INTRODUCTION

Coronavirus disease 2019 called (COVID-19), was induced by the severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2), first appeared in December 2019 and quickly spread worldwide. COVID-19 is highly infective and manifests as an acute respiratory tract infection syndrome. Critically-ill COVID-19 patients have a high mortality rate.

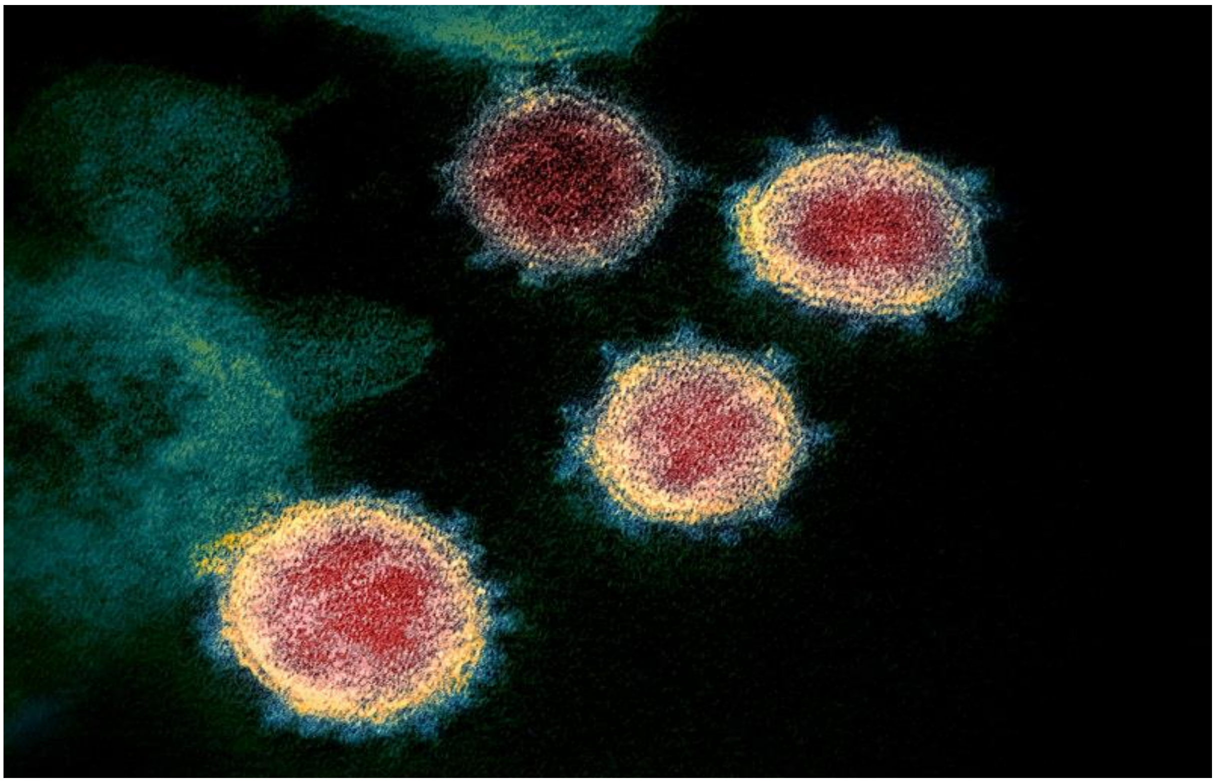


Figure 1 Transmission electron micrograph of SARS-CoV-2 virions with visible coronae

The COVID-19 virus producing the COVID-19 Pandemic is a serious coronavirus syndrome 2 (SARS ‑ CoV ‑ 2) (Coronavirus disease 2019). In 2019, the new coronavirus (2019-nCoV) and human coronavirus (known as the simple coronavirus) were named after the preliminary designation (HCoV-19 or hCoV-19). On 30 January 2020, a pandemic and international emergency was issued by the World Health Organization on 11 March 2020. SARS ‑ CoV ‑ 2 is a human-contagious RNA virus that has a single beach. The severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) is a new coronavirus associated with severe acute respiratory syndrome. It was originally discovered in three patients who had pneumonia and were linked to a Wuhan cluster of acute respiratory disease cases. In nature, all of the structural characteristics of the new SARS-CoV-2 virus particle are seen in similar coronaviruses. Household soap, which breaks the virus's protective bubble outside the human body, kills it.

The SARS-CoV-2 virus is linked to the original SARS-CoV virus. It's considered to have a zoonotic (animal-borne) origin. The coronavirus genetically clusters with the genus Beta coronavirus, in subgenus Sarbecovirus (lineage B), together with two bat-derived strains, according to genomic research. At the entire genome level, it's 96 percent identical tKills the man's body. The original SARS-CoV virus is associated to SARS-CoV. A zoonotic (animal-borne) origin is considered. In the Sarbek virus subgender (lineage B), the coronavirus cluster genetically with genetically modified beta coronavirus is combined with two genomic genomic strains. It is 96% identical to earlier bat coronavirus strains at the whole genome level (BatCov RaTG13). Diabetic glycoprotein (M), protein envellement (E), nucleocapside (N), and spike protein (S) are structural SARS-CoV-2 proteins (S). O earlier strains of coronavirus (BatCov RaTG13). The SARS-CoV-2 Structural Proteins are diabetic glycoprotein (M), protein envelope (E), protein nucleocapside (N), and spike protein (S).

More than 25 million COVID-19 cases had been confirmed in India by May 2021, with more than 164 million cases contracted globally [1].

1.1 MAIN SYMPTOMS OF COVID-19:

- Fever
- Coughing
- Shortness of breath
- Trouble breathing
- Fatigue
- Chills, sometimes with shaking
- Body aches
- Headache
- Sore throat
- Congestion/runny nose
- Loss of smell or taste
- Nausea
- Diarrhea

Pneumonia, a low WBC count, and a reduced lymphocyte count are other typical clinical symptoms of COVID-19 cases.

1.2 STANDARD TECHNIQUES FOR COVID-19 DETECTION

1.2.1 RT-PCR

The standard model for screening suspicious patients is reverse-transcription polymerase chain reaction (RT-PCR) testing [2]. RT-PCR is a laboratory technique that combines reverse transcription of RNA to DNA and enhanced special DNA targets with polymerase reaction chain (in this context referred to as complementary **DNA** or c**DNA**) (PCR). It is used mostly for measuring the quantity of a particular RNA. The amplification process is monitored with fluorescence, a method known as PCR or quantitative PCR (qPCR). Combined RT-PCR and qPCR are commonly employed in research and clinical environments for investigation of gene expression and quantification of the virus RNA.

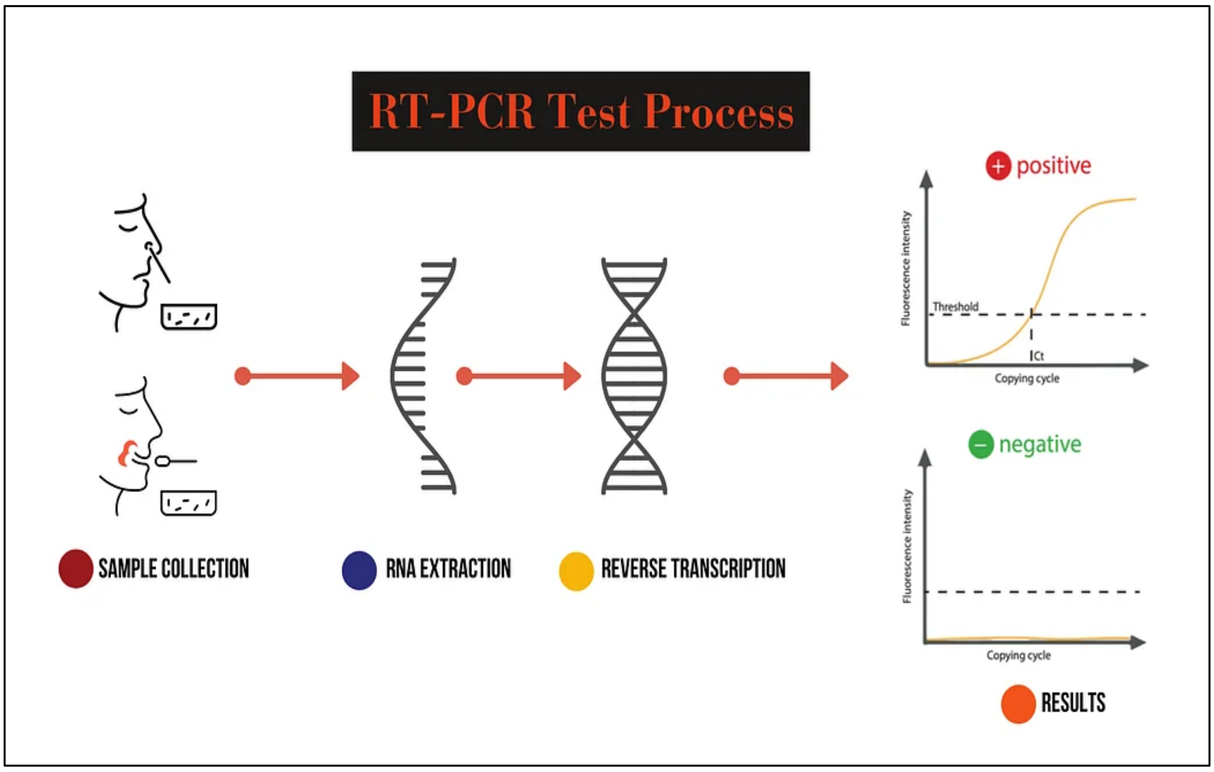


Figure 2 RT-PCR Test Process

RT-PCR screening, on the other hand, has a low sensitivity in some situations. SARS-CoV-2 infection cannot be ruled out entirely even if RT-PCR findings from a suspicious patient are negative.

1.2.2 Chest Computed Tomography Detection

As a result, medical imaging, specifically chest computed tomography (CT), is frequently used as a supplement to other tests in diagnosing and treating COVID-19. COVID-19's chest imaging data were initially published in January 2020, and the majority of hospitalised patients had bilateral lung involvement and ground-glass opacities. Since then, a slew of publications on COVID-19 chest CT findings have been released at breakneck speed. The proper use of chest CT in COVID-19 patients should be based on experience and, above all, scientific information that has emerged since the disease's onset and continues to accumulate.

The use of computed tomography (CT) in the diagnosis and treatment of COVID-19 pneumonia is critical. COVID-19 pneumonia is characterised by bilateral patchy regions of ground glass infiltration, with greater infiltration in the lower lobes. During the course of the disease, further symptoms such as consolidation, air bronchogram, crazy pavement look, and air bubble signals occur. The presence of pleural effusion or pericardial effusion is associated with a bad prognosis. In the diagnosis of COVID-19 pneumonia, thin section chest CT shows a high sensitivity but a low specificity.

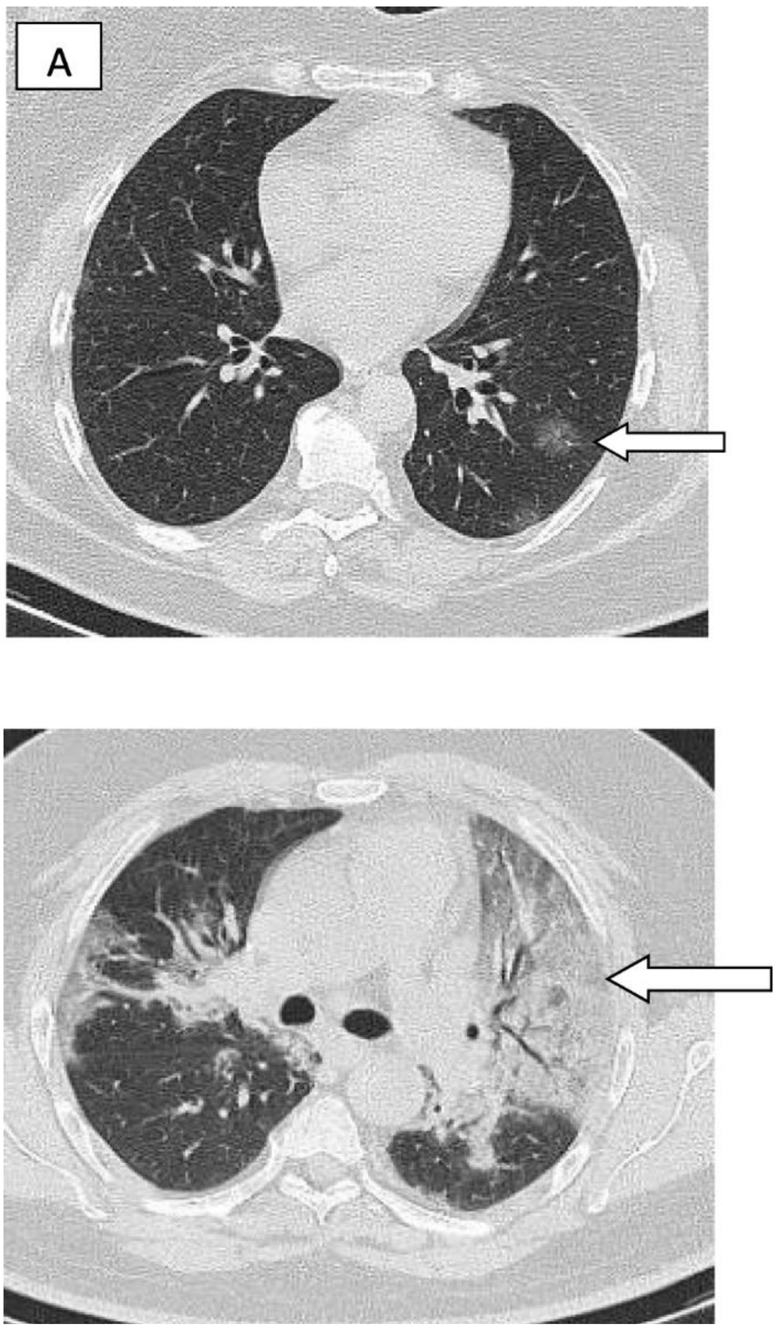


Figure 3 Covid infected lungs shows fibrosis developed in major area

The HR-CT image of a COVID-19 patient shows how the fibrosis in the lungs has progressed over time.

1.3 COMPARISON OF RT-PCR AND HRCT DETECTION

The sensitivity of CT is determined by the length of time that symptoms have been present. Bernheim et al. [27] found that negative CT scans were found in 56 percent of patients scanned during the first two days of symptom start, 9 percent of patients scanned within 3–5 days, and 4 percent of patients examined 6–12 days after symptom onset.

CT appears to have a higher sensitivity than the rRT-PCR test in general. Long et al. found that CT sensitivity was 97.2 percent, compared to 83.3 percent for the first rRT-PCR test. In an emergency, microbiological tests such as real-time polymerase chain reaction (RT-PCR) may not be accessible, and results might take up to 5 days. In the current pandemic, computed tomography (CT) can, on the other hand, be a significant supplement to RT-PCR for detecting COVID-19 pneumonia. When the viral load is inadequate, RT-PCR might sometimes provide misleading negative results.

Ai et al. discovered that 59 percent (601/1014) of patents had positive RT-PCR compared to 88 percent (888/1014) had positive CT chest in a research comparing the performance of RT-PCR and CT chest in 1014 patents. They discovered that 60% of patients had positive CT findings before to or concurrent with positive RT-PCR, and that virtually all patients (56/57) had positive CT findings prior to or within 6 days of positive RT-PCR. Furthermore, 70% of individuals with a negative RT-PCR exhibited a typical chest CT look. The sample strategy, specimen source (upper or lower respiratory tract), viral load, sampling period, and kit type can all impact the RT-PCR.

In the suggested solution, Transfer Learning trains the CT image data using pre-trained architectures such as ResNet50 [3], DenseNet121 [4], VGG16 [5], and AlexNet [6] with Imagenet weights, and Progressive resizing aids the models learning to achieve impressive results.

CHAPTER 2 RELATED LITERATURE REVIEW

Deep learning approaches may learn from basic representations to understand complicated issues. The capacity to learn precise representations and the property of learning data in a deep way where several layers are used sequentially , are the primary characteristics that have made deep learning approaches popular. Deep learning algorithms are widely utilised in medical systems, including biomedicine , smart healthcare , drug development , medical image analysis , and so on.

It's been widely utilised in the automated diagnosis of COVID-19 in patients in recent years. Data gathering, data preparation, feature extraction and classification, and performance evaluation are all processes in deep learning-based systems in general. Figure 1 depicts the main pipeline of a COVID-19 diagnostic system based on deep learning. Patients from the hospital setting are considered participants during the data collecting stage. Although the data can take many various forms, imaging methods such as CT and X-ray samples are used to diagnose COVID-19.
Data preparation, which transforms the data into an acceptable format, is the next required step.

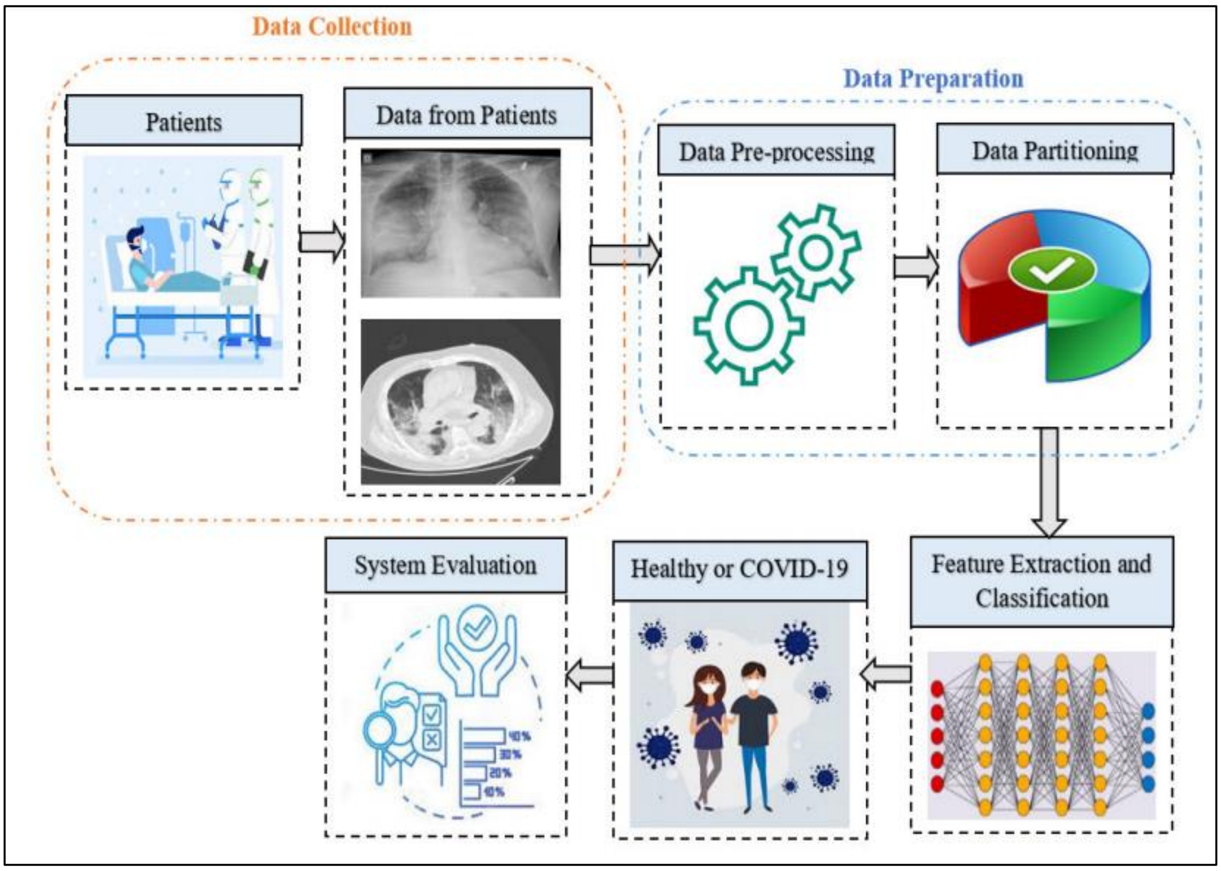


Figure 4 Steps in the diagnosis of COVID-19 using deep learning

Data pre-processing comprises procedures such as noise reduction, scaling, and augmentation, among others. The data partitioning phase divides the data into three sets for the experiment: training, validation, and testing. For data partitioning, the cross-validation approach is commonly used. The training data is used to create a specific model, which is then reviewed using validation data, and the generated model's performance is assessed using test data. The feature extraction and classification are the most important steps in a deep learning-based COVID-19 diagnosis.

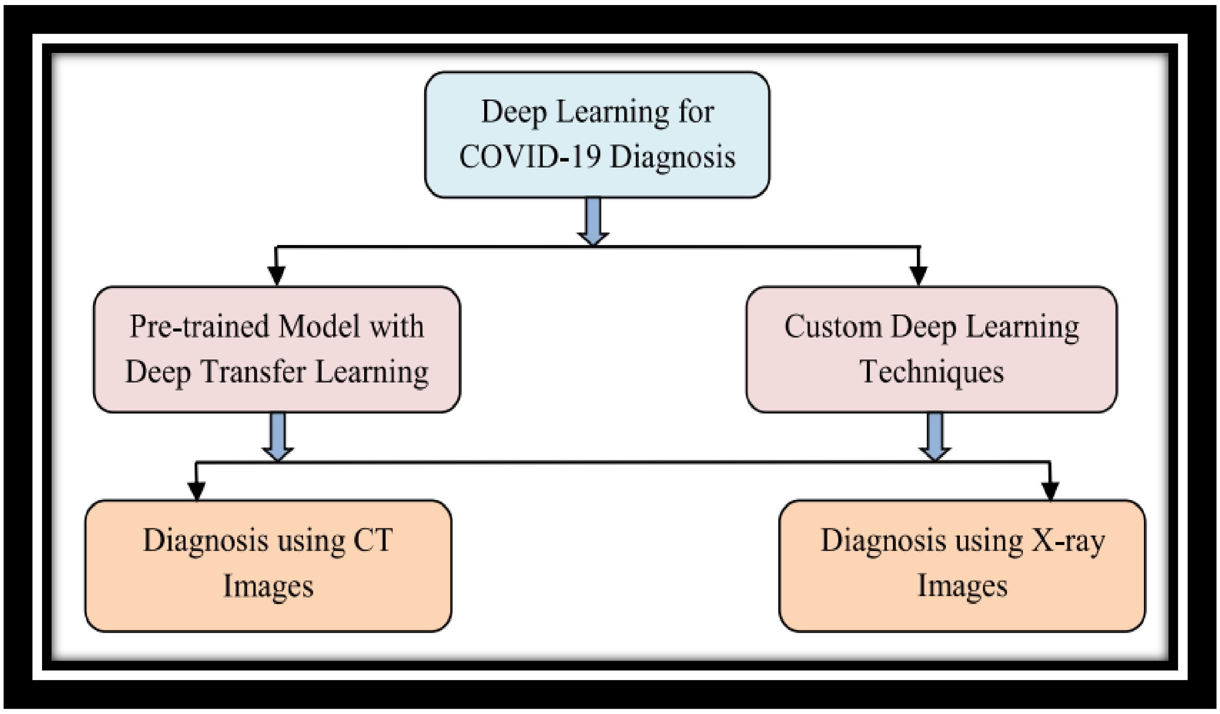

Deep Learning for COVID-19 Diagnosis
Pre-trained Model with Deep Transfer Learning
Custom Deep Learning Techniques
Diagnosis using CT Images
Diagnosis using X-ray Images

Figure 5 Classification of Deep learning methods for COVID-19 diagnosis

In this stage, the deep learning approach extracts the feature by repeating multiple operations, and then the classification is done based on class labels (healthy or COVID-19). Finally, various assessment measures such as accuracy, sensitivity, specificity, precision, F1-score, and others are used to evaluate the constructed system.

A pre-trained model has previously been trained in fields that are similar to the application's environment. Weight and bias are transferred from a big trained model to a similar new model for testing or retraining in transfer learning. Using a pre-trained model with deep transfer learning has numerous advantages. Training a model from scratch for big datasets, in general, necessitates a lot of computational resources and takes a long time . The facility can speed up the convergence with network generalisation by using a pre-trained model with transfer learning .

Based on the device and settings, 3D CT scans have a fixed number of slices (16, 32, 64, 128, etc.). In nature, the individual slices might be greyscale or colour images. The slices are usually removed first and then processed as distinct pictures .

The slices with the greatest number of lung regions are chosen, while the rest are eliminated. The centre 50% of slices from 3D CT images are chosen in . The pre-trained models are directly optimised using the individual slices or characteristics derived from these slices.

In other situations, 3D segmentation models like U-Net models are used to segment and extract information from numerous Regions of Interest (ROI) from 3D CT images.

images The pre-trained models are then optimised using these numerical characteristics. Following that, we'll go through the COVID19 diagnosis systems that have been created.

2.1 DIAGNOSIS USING COMPUTER TOMOGRAPHY (CT) IMAGES

2.1.1 Diagnosis based on multiple source data

Wu et al. proposed a multi-view fusion-based deep learning-based coronavirus screening system.

ResNet50, a CNN version, was employed in the system. The data was gathered from two Chinese hospitals. For the experiment, a total of 495 pictures were used, 368 of which were linked with proven COVID-19 cases and 127 of other pneumonia. The dataset is split into 80 percent, 10%, and 10% proportions for training, testing, and validation, respectively, under this manner. Before the network is built, each of the pictures in the system is scaled to 256 × 256 pixels.

The developed system has a 76 percent accuracy, 81.1 percent sensitivity, 61.5 percent specificity, and 81.9 percent Area under Curve (AUC) in the test scenario. The findings are compared for single-view and multi-view fusion models, however the multi-view fusion model outperforms the single-view fusion model. Li et al. presented an automated system (COVNet) for diagnosing coronavirus from CT scans using a deep learning approach called ResNet50, which is a version of CNN. The 4536 chest CT samples utilised in this study include 1296 samples for COVID-19, 1735 samples for community-acquired pneumonia (CAP), and 1325 samples for non-pneumonia.

After that, Yousefzadeh et al. presented a deep learning framework called ai-corona, which is based on CT scans, for the accurate diagnosis of COVID-19. DenseNet, ResNet, Xception, and EfficientNetB0 are among the CNN versions used in the system. The total number of CT slices in the collection was 2124, including 1418 scans of non-COVID-19 patients and 706 slices of COVID-19 infected individuals. For the training and validation sets, the dataset maintained a ratio of 80% and 20%, respectively. The experiment revealed that the suggested system has 96.4 percent accuracy, 92.4 percent sensitivity, 98.3 percent specificity, 95.3 percent F1-score, and 98.9 percent AUC.

In another study, Xu et al. used CNN variations to build a method for distinguishing healthy people from COVID-19 pneumonia and Influenza-A viral pneumonia. Resnet18 is the pre-trained model in this system. The information was gathered from three separate Chinese hospitals. This study uses 618 CT scans, 219 of which were collected from COVID-19-infected patients, 224 from Influenza-A virus pneumonia, and 175 from healthy people. A total of 85.4 percent (528) pictures were utilised to train the model, with the remaining samples being used to test the generated model. In the trial, the framework scored 86.7 percent accuracy, 81.5 percent sensitivity, 80.8 percent precision, and 81.1 percent F1-score. Furthermore, Jin et al. developed a deep learning-based medical system for COVID-19 screening. Their approach combined 3D U-Net++ with pre-trained CNN models such as DPN92, Inception-v3, ResNet-50, and Attention ResNet-50. The data was gathered from five different Chinese hospitals. A total of 139 samples are utilised in this method, including 850 samples from COVID-19 and 541 samples from other instances that are declared negative. For performance evaluation, the whole data is randomly divided into training and testing sets.

Using 3D Unet++-ResNet-50, which is regarded the best model, the system achieved sensitivity, specificity, and AUC of 97.4 percent, 92.2 percent, and 99.1 percent, respectively, as assessment metrics. In addition, Javaheri et al. [65] developed CovidCTNet, a deep learning approach for detecting coronavirus infection via CT images. The BCDU-Net architecture, which is based on U-Net, was utilised in the system. COVID-19 was differentiated from CAP and other lung diseases using this system. In all, 89,145 CT scans were used in the trial, with 32,230 CT slices confirmed with COVID-19, 25,699 CT slices confirmed with CAP, and 31,216 CT slices with healthy lungs or other disorders. The dataset is partitioned using the hold-out approach, with 90% of the data used for training and 10% for testing. The created system achieved accuracy, sensitivity, specificity, and AUC of 91.66 percent, 87.5 percent, 94 percent, and 95 percent, respectively, according to the testing findings.

2.1.2 Diagnosis based on single source data

Ardakani et al. presented a method for detecting COVID-19 in CT images using 10 different CNN algorithms. AlexNet, VGG-16, VGG-19, SqueezeNet, GoogleNet, MobileNet-V2, ResNet-18, ResNet-50, ResNet-101, and Xception are some of the most often used versions for diagnosis. A total of 1020 CT samples from COVID-19 and non-COVID-19 patients are evaluated in the proposed system. The dataset is divided into two parts: training and validation, which account for 80% and 20% of the total. ResNet101 and Xception outperformed the other ten networks in terms of performance.

The ResNet-101 model achieved accuracy of 99.51 percent, sensitivity of 100 percent, AUC of 99.4 percent, and specificity of 99.02 percent, as evidenced by the experimental findings. Xception discovered accuracy, sensitivity, AUC, and specificity of 99.02 percent, 98.04 percent, 87.3 percent, and 100 percent, respectively, in another network. Chen et al. proposed a deep learning-based approach for COVID-19 identification using high-resolution CT images and a strong pre-trained model dubbed UNet++. UNet++ first extracted valid areas from CT scans. In this study, 46,096 pictures from a hospital were gathered, with 51 individuals infected with COVID-19 and 55 with other illnesses. 35,355 pictures are chosen from the dataset after filtering out low-quality photographs and dividing them into training and testing sets. The results show that the sensitivity is 94.34 percent, the specificity is 99.16 percent, the accuracy is 98.85 percent, the precision is 88.37 percent, and the negative predictive value (NPV) is 99.61 percent. Cifci has described a method for detecting coronavirus early utilising multiple pre-trained models and deep transfer learning. AlexNet and Inception-V4, two prominent medical image analysis models, are among the pre-trained models. CT scans are used to conduct the research. 5800 CT scans were collected from a public source to help design the system.

4640 CT samples (80%) are utilised in the training phase, whereas 1160 (20%) are used in the testing phase. AlexNet outperformed InceptionV4 in terms of performance, as evidenced by experimental findings. AlexNet scored a 94.74 percent overall accuracy, with 87.37 percent sensitivity and 87.45 percent specificity, respectively.

2.2 DIAGNOSIS USING X-RAY IMAGES

2.2.1 Diagnosis based on multiple source data

Using the idea of transfer learning and five versions of CNNs, Apostolopoulos and Bessiana created a system for the automated diagnosis of COVID-19 cases.

VGG19, MobileNetv2, Inception, Xception, and InceptionResNetv2 are the pre-trained models in the system. In the first scenario, the system looked at 1427 pictures, including 224 for COVID-19, 700 for common pneumonia, and 504 for healthy patients. In the second scenario, 504 healthy individual photos are compared to 224 COVID-19 images, 714 bacterial and viral pneumonia images, and 224 COVID-19 images. The 10-fold cross-validation approach was used to split the dataset.

The second dataset was found to have the highest accuracy of 96.78 percent, sensitivity of 98.66 percent, and specificity of 96.46 percent when utilising MobileNetv2. Loey et al. presented a new method for coronavirus detection based on the Generative Adversarial Network (GAN) and pre-trained CNN models with deep transfer learning. Alexnet, Googlenet, and Resnet18 are the pre-trained models employed in the proposed system. Because the number of COVID-19 X-ray pictures is limited, GAN is employed to produce additional samples in order to identify the virus accurately. There are 307 pictures in all, divided into four classes: COVID-19, normal, pneumonia bac, and pneumonia vir. Depending on the consideration of class level, the system tested three distinct dataset scenarios. Googlenet had the highest accuracy of 80.6 percent when four classes were considered. With three and two classes, Alexnet and Googlenet obtained accuracy of 85.2 percent and 100 percent, respectively. Horry et al. presented a COVID-19 detection framework in X-ray pictures utilising the idea of a pre-trained model.

The suggested method combined transfer learning with four famous pre-trained models: VGG, Inception, Xception, and Resnet. For the tests, 100 COVID-19 patients, 100 pneumonia cases, and 200 healthy cases were utilised. As a data partition, an 80:20 ratio is maintained in this system for training and testing. The system achieved accuracy, sensitivity, and F1-score of 83 percent, 80 percent, and 80 percent, respectively, using VGG19, which was assessed as the best performance in the research when three-class data was included.

Ozcan also presented a deep learning system that combined the grid search strategy with three pre-trained CNN models (GoogleNet, ResNet18, and ResNet50). The optimal hyperparameter is chosen using the grid search approach, and pre-trained models are employed for feature extraction and classification. The system used pictures from 242 bacterium cases, 131 COVID-19 cases, 200 normal cases, and 148 viral cases from three public datasets. All of the data is divided into three groups: training, testing, and validation, in the proportions of 50:30:20.

The ResNet50 with grid search fared better, with a 97.69 percent accuracy, 97.26 percent sensitivity, 97.90 percent specificity, 95.95 percent precision, and 96.60 percent F1-score. Sethy and Behra proposed a method for diagnosing COVID-19 instances that relied on pre-trained CNN and SVM models (SVM). For automated feature extraction, the method employed eleven CNN pre-trained models and SVM for classification. Two different datasets were employed in this system, with the first dataset containing 25 positive COVID-19 X-ray pictures and the second dataset containing 25 negative COVID-19 X-ray images.

According to the experimental results, Resnet50 with SVM achieved accuracy, False Positive Rate (FPR), Matthews Correlation Coefficient (MCC), and Kappa of 95.38 percent, 95.52 percent, 91.41 percent, and 90.76 percent for the first scenario of the dataset, respectively, which is the best in the developed system. Minaee et al. developed Deep-COVID, a framework for COVID-19 prediction in X-ray pictures that uses the idea of deep transfer learning. This study looked at four common pre-trained models for COVID19 diagnosis: ResNet18, ResNet50, SqueezeNet, and DenseNet-121. A total of 5071 pictures were gathered from various open-access domains.In the studies, 2000 pictures with 31 COVID-19 patients were utilised for training, while 3000 images with 40 infected COVID-19 cases were used for testing. COVID-Xray5k was the name given to the resultant dataset. Using SqueezeNet, the system achieved the greatest results, with a sensitivity of 100% and a specificity of 95.6 percent.

Punn and Agarwal created an automated COVID-19 diagnostic system using a limited number of X-ray images utilising multiple pre-trained models such as ResNet, Inception-v3, Inception ResNet-v2, DenseNet169, and NASNetLarge. Transfer learning was utilised to fine-tune the system, which used random oversampling and a weighted class loss function. A total of 1076 chest X-ray pictures are considered for trials in this system. For training, testing, and validation sets, the dataset is divided into 80 percent, 10%, and 10% ratios, respectively. According to the results, NASNetLarge fared better than the others, with accuracy, precision, sensitivity, AUC, specificity, and F1-scores of 98 percent, 88 percent, 91 percent, 99 percent, 98 percent, and 89 percent, respectively.

2.2.2 Diagnosis based on single source data

Hemdan et al. recently presented COVIDX-Net, a method for diagnosing coronavirus utilising CNN variations in X-ray images. This study takes into account a total of seven pre-trained models. The collection included 50 pictures, 25 of which were from healthy persons and the remaining 25 from COVID-19 patients. The dataset was split in half for the experiment, with the training and testing sets each receiving 80% and 20% of the total. VGG19 and DenseNet beat the other pre-trained models with an accuracy of 90% and an F1-score of 91%, respectively, in the experiments. The poorest results were produced with InceptionV3.

Other Researchers have suggested different approaches to identify COVID-19 positive patients and isolate them as quickly as possible in recent years.Most researchers have proposed the baseline models implemented on the Keras, Twang, et al. [7] have proposed an eight-layer-based AlexNet model. It gives an accuracy of 91.32 percent. In their paper, For more extended training periods, the overfitting phenomenon will occur. Positano et al. [8] suggested To increase predictive accuracy by a decision fusion-based approach, which integrates the outcomes of each of the individual Deep Convolutional Neural Network models. The authors of [9] employed CT images and transfer learning from a pre-trained DenseNet201 network to distinguish COVID19 patients from non-COVID-19 people. The accuracy of the model was 96.25 percent. Wang et al. [10]proposed a new learning framework for COVID-19 patient identification. COVID-Net was updated to integrate the changes in architecture and learning techniques.

CHAPTER 3 HR-CT DATASET

The method of high-resolution computed tomography (HRCT) is frequently utilised to scan a variety of lung pathologies. In contrast to helical CT, HRCT takes thin slice pictures of the lung parenchyma using a narrow beam collimation. The pictures of lung alveoli, airways, interstitium, and pulmonary vasculature produced by this procedure are exceptionally high quality. Expiration pictures may help discover air-trapping in lung disease patients.

A standard CT scanner is used to perform HRCT. The imaging settings, on the other hand, are chosen to optimise spatial resolution. A small slice width (typically 1–2 mm) is employed, a high spatial resolution image reconstruction method is used, field of vision is limited to reduce pixel size, and other scan parameters (e.g. focus point) may be adjusted for resolution at the price of scan speed. The scan may be done in both inspiration and expiration, depending on the probable condition. In addition to the more common supine position, the patient may also lie prone (face down) (face up).

Because the goal of HRCT is to assess a widespread lung illness, it is traditionally done by obtaining thin slices 10–40 mm apart. As a result, there are a few pictures that should be typical of the lungs as a whole, but they only cover about a tenth of the lungs. HRCT does not employ intravenous contrast agents since the lung has a high contrast (soft tissue versus air) thus the method is inappropriate for assessing soft tissues and blood vessels, which are the main objectives of contrast agents.

Even though some lung illnesses, such as emphysema or bronchiolitis obliterans, involve relatively slight changes in lung structure in their early stages, they induce air trapping on expiration. The scan can be done in both inspiration and expiration to increase sensitivity for these situations.

HRCT can be used to diagnose diseases including emphysema and bronchiectasis. While HRCT can detect pulmonary fibrosis, it may not always be able to classify the fibrosis into a specific pathogenic form (e.g., non-specific interstitial pneumonitis or desquamative interstitial pneumonitis).

The most notable exception is UIP, which has distinct characteristics and can be reliably identified with only HRCT. When HRCT is unable to provide a conclusive diagnosis, it can assist in the location of an abnormality and therefore in the planning of a biopsy, which may offer the final diagnosis. Lymphangitis carcinomatosa, fungal, or other unusual infections, chronic pulmonary vascular disease, lymphangioleiomyomatosis, and sarcoidosis are some of the additional diseases for which HRCT is effective. Patients who have received an organ transplant, particularly a lung transplant or a heart-lung transplant, are at a higher risk of developing pulmonary problems as a result of their long-term medication and immunosuppressive therapy.

Bronchitis obliterans is the most common pulmonary complication, and it might be an indication of lung transplant rejection. HRCT has a higher sensitivity than traditional radiography for bronchiolitis obliterans. Annual HRCT screening may be arranged by some transplant facilities. **One of the major diagnostic techniques for COVID-19 is diagnostic imaging**, which includes HRCT. [10] There is some dispute concerning the use of CT as a diagnostic tool when compared to other techniques and imaging modalities. Infected patients had multifocal or unifocal involvement of ground-glass opacity on HRCT scans (GGO).

3.1 DATASET DESCRIPTION

A publicly available SARS-CoV-2 high-resolution CT scan dataset [11] with 1252 CT scans positive for SARS-CoV-2 infection and 1230 CT scans for patients not infected with COVID-19 for a total of 2482 CT scan images.

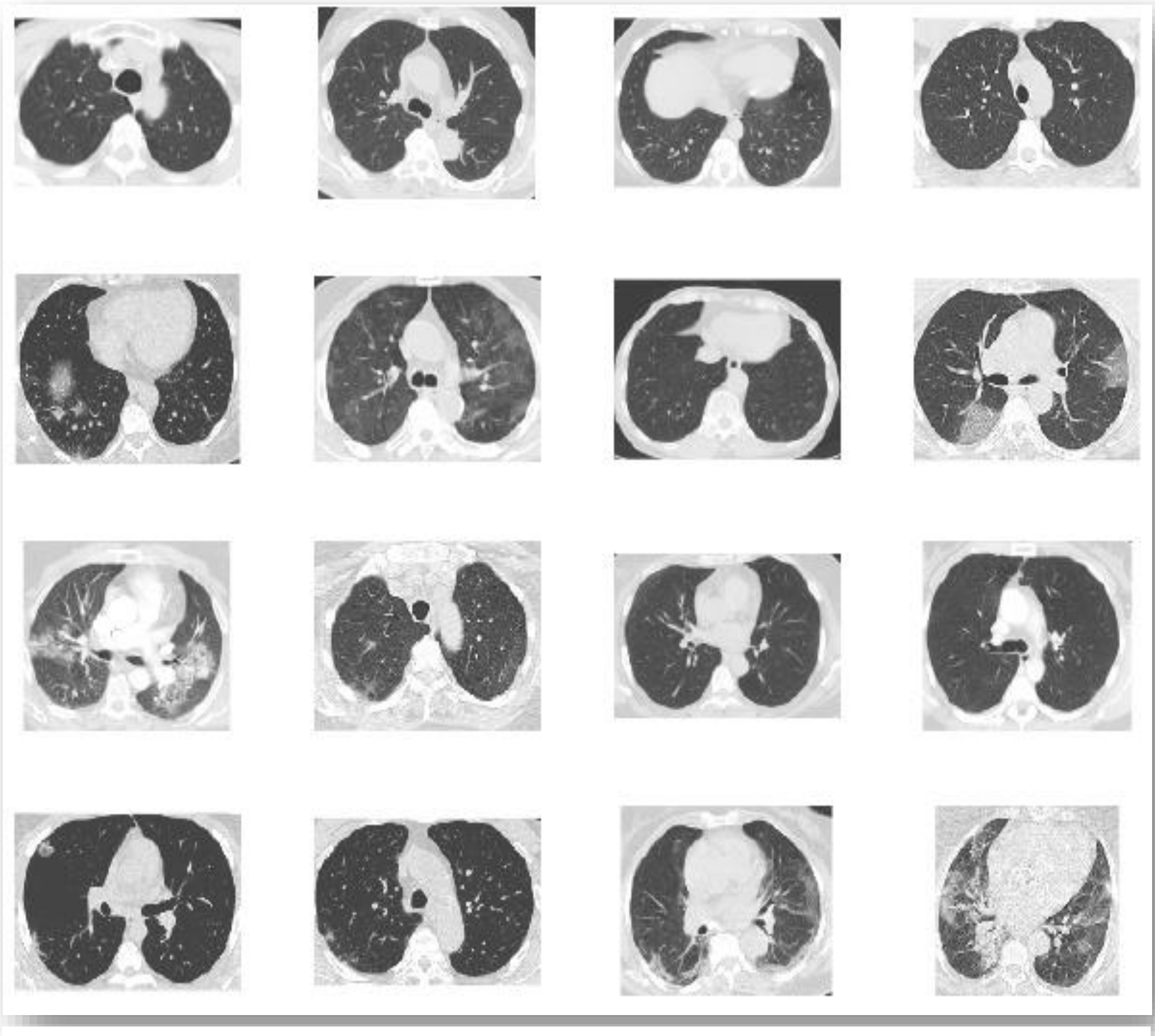


Figure 6*:* CT Images in the Dataset used in this project

The figure illustrates the example of CT scans for different patients infected and non infected by SARS-CoV-2. One can see that the lungs which are infected by covid are having high white Flecked patterns.

The HR-CT images are from actual patients in hospitals in Sao Paulo, Brazil. This collection of data aims to promote artificial intelligence research and development. This dataset is favorable because It is easy to upload a small dataset on Co-lab and other gradient spaces also it has a sufficient amount of images to train a model. Owing to ethical issues, hospitals have withheld specific information about each patient. Dataset is available at Kaggle [12]. some of the CT scans from the dataset are shown in Fig 5:

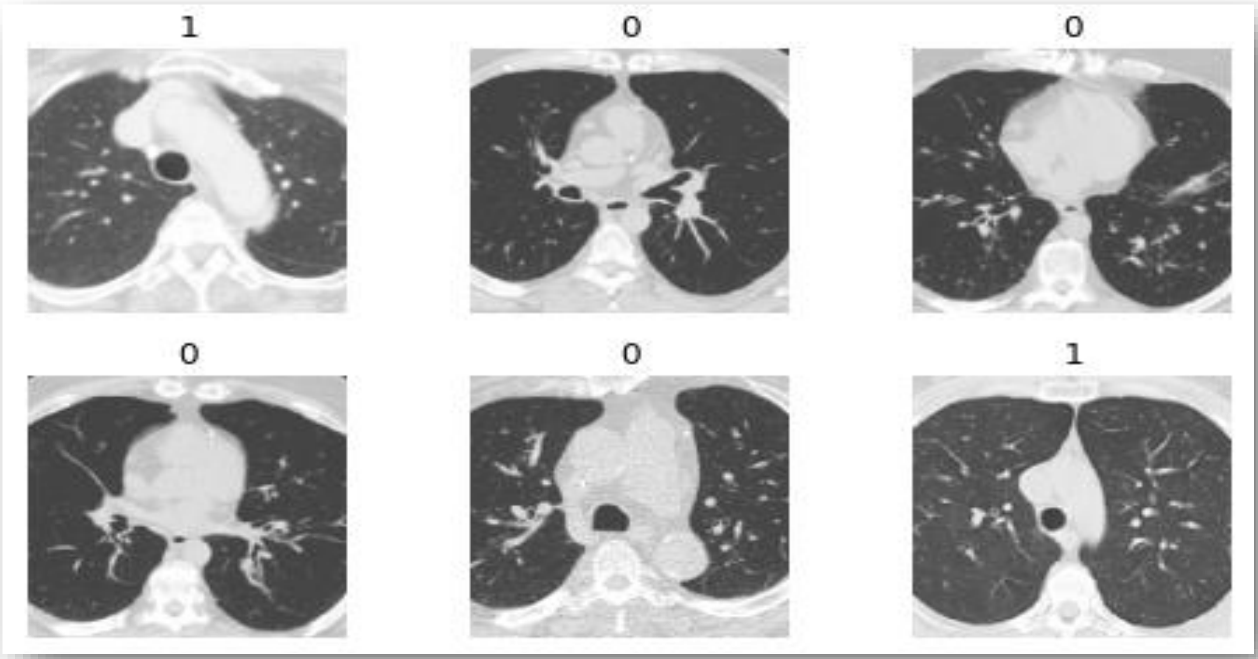


Figure 7 HR-CT images from the dataset

Here 1 denotes the lungs which are infected from SARS-Cov-2 and 0 denotes healthy lungs.

3.2 DATASET ANALYSIS

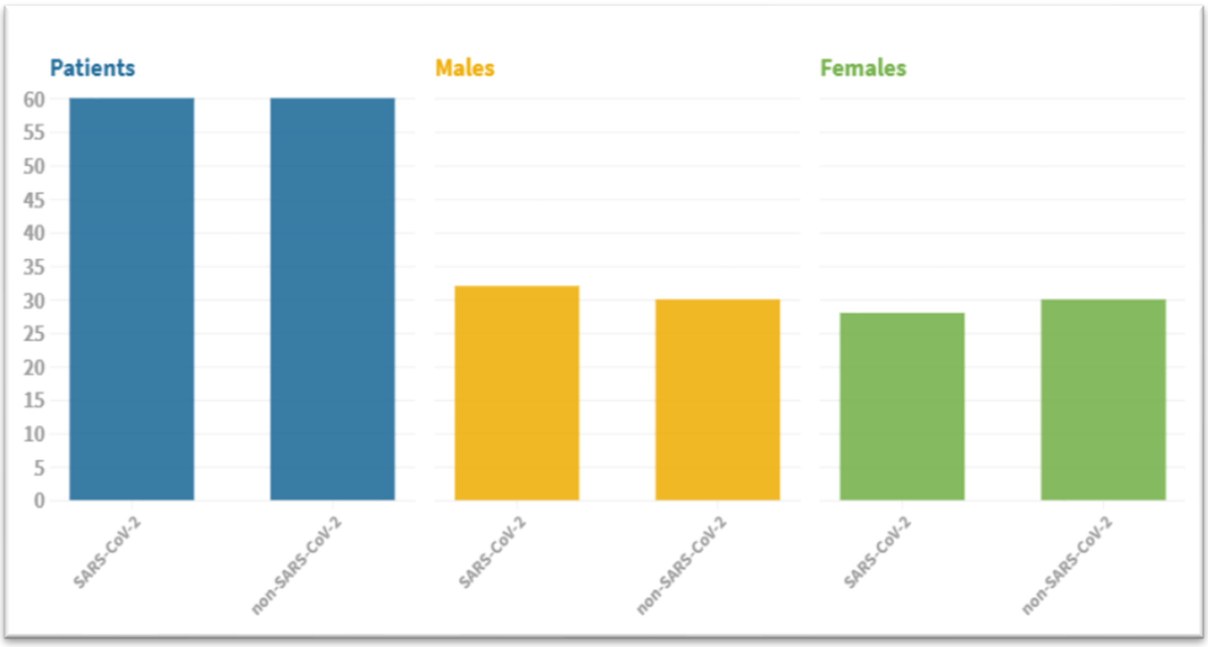


Figure 8. The total number of patients included in the dataset *[11]*.

The dataset includes CT slices from 120 patients, 60 of whom are covid positive and the rest are negative. There are 32 males and 28 females among the 60 positive cases. The covid negative patients are evenly distributed, with 30 males and 30 females as shown in Figure 2. In terms of sex ratio, the dataset appears to be evenly balanced. The original dimensions of these images range from 119×104×3 to 416×512×3 [11].

CHAPTER 4 API AND RESOURCES USED IN PROJECT

FastAI is a deep learning library that offers practitioners with high-level components that can provide state-of-the-art results in typical deep learning domains quickly and easily, as well as academics with low-level components that can be combined and matched to create novel methods. It tries to achieve both goals without sacrificing usability, flexibility, or performance. This is made feasible by a well-designed layered architecture that represents the common underlying patterns of various deep learning and data processing algorithms as disconnected abstractions.

By combining the dynamic of the underlying Python language with the flexibility of the Py-Torch library, these abstractions can be represented succinctly and unambiguously. FastAI consists of the following ingredients:

- A semantic type hierarchy for tensors, as well as a new type dispatch system for Python.
- A pure Python extension for a GPU-optimized computer vision library.
- An optimizer that refactors contemporary optimizers' common functionality into two fundamental parts, allowing optimization methods to be built in only 4–5 lines of code.
- A revolutionary two-way callback system that allows you to alter any component of the data, model, or optimizer at any time during training.

Google Research's Colaboratory, or "Colab" for short, is a product. Colab is a web-based Python editor that allows anybody to create and run arbitrary Python code. It's notably useful for machine learning, data analysis, and teaching. Colab is a hosted Jupyter notebook service that doesn't require any setup and offers free access to computational resources, including GPUs.

4.1 MACHINE AND GPU SPECIFICATIONS

Machine specs:
Processor: Intel Xeon
CPU count: 8 vCPUs
Clock speed: 2.60 Ghz
Host memory: 30 GiB
Intel AVX and AVX2 support:Yes
GPU x 1
GPU Type 1: NVIDIA QUADRO M4000
GPU memory: 8 GiB.
or
GPU Type 2: NVIDIA QUADRO P5000
GPU memory: 8 GiB.

4.2 LIBRARIES USED IN THE PROJECT

- Py-Torch version 1.4
- Py-Torchvision version 0.5.0
- OS
- FastAI version 1
- NumPy
- Glob
- scikit-learn
- pandas

CHAPTER 5 DATA AUGMENTATION

One of the most frequent regularisation approaches, especially in image processing jobs, is data augmentation. When working on a Machine Learning model, the performance of your model is only as good as your data. You'll need a varied quantity of data depending on the model and problem you're trying to solve. Data collection and processing, on the other hand, is a time-consuming procedure that isn't always possible.

In every machine learning project, we want to make sure that our code, or "model," can generalise to real-world data. Overfitting, on the other hand, occurs when your model only learns to detect features inside your training dataset. To circumvent this, we "augment" or add minor changes to our images before feeding them to the model. Even though a 2-degree rotation may not appear to make much of a difference to the naked eye, such minor alterations are sufficient to allow the model to generalise effectively.

5.1 IMAGE AUGMENTATION USING FASTAI

When constructing the “ImageDataBunch” objects, we supply the list of transforms to apply to your dataset. FastAI includes a set of default suggested transformations that have been generated from the team's extensive testing, so I'd recommend starting with these:

Having a substantial amount of data is the first step in training a successful deep learning architecture [13] [14]. It isn't always possible, however. As a result, we apply small random transformations to the data that do not affect the image's content (for the human eye) but affect the pixel values [15]. Data Augmentation-trained models are more generalizable. There are several data augmentation techniques available in the FastAI API [16]. For our approach, we have used three methods because the dataset is already High resolution.

5.1.1 **Flip:** Flipping refers to the rotation of an image around a horizontal or vertical axis. The horizontal flipping will be on the vertical axis, while the vertical flipping will be on the horizontal axis. Flipping gives our model more cases to consider.

5.1.2 **Random Rotation**: The slices do not always appear straight. They might have different orientations. Hence, we randomly rotate few images in a range of (-15, 15) degrees.

5.1.3 **Normalization on ImageNet stats:** Normalization means getting the mean, standard deviation for each channel of an image. Using custom ImageNet stats for the PyTorch pre-trained models provides better results than without normalization as it is trained on ImageNet.These are calculated based on millions of images of ImageNet [17]. ImageNet is a collection of approximately 15 million high-resolution pictures that have been categorised into around 22,000 categories. Using Amazon's Mechanical Turk crowd-sourcing technology, the photos were gathered from the internet and identified by human labelers. An annual competition named the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) has been organised as part of the Pascal Visual Object Challenge since 2010. ILSVRC makes use of a subset of ImageNet, containing around 1000 pictures in each of 1000 categories. A total of 1.2 million training pictures, 50,000 validation images, and 150,000 testing images are available. ImageNet is a collection of pictures of varying resolutions. As a result, the pictures have been reduced to a fixed resolution of 256256 pixels. A rectangular picture is rescaled and the centre 256256 patch is cut out of the resultant image.

Imagenet_stats :

Mean: [0.485, 0.456, 0.406]

Standard_deviation: [0.229, 0.224, 0.225]

The three values given above depict the mean and standard deviation for each RGB channel of every CT image.

CHAPTER 6 PROPOSED ARCHITECTURE

Instead of developing a new architecture by scratch, we confide in the knowledge of already available CNN architectures that have proved great results across a broad range of classification tasks. We use ResNet50 [3], a residual neural network variation with 50 layers along with DenseNet121 [4], VGG16 [5], and AlexNet [6].

6.1 ResNet50

ResNet, short for Residual Network, is a form of neural network developed by Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun in their article "Deep Residual Learning for Image Recognition" published in 2015. ResNet models were highly successful, as evidenced by the following:

- With a top-5 mistake rate of 3.57 percent, won first place in the ILSVRC 2015 classification competition (An ensemble model)
- ImageNet detection, ImageNet localization, Coco detection, and Coco segmentation took first place in the ILSVRC and COCO 2015 competitions.
- ResNet-101 replaces VGG-16 layers in Faster R-CNN. They saw a 28 percent improvement in relative terms.
- Easily trained networks with 100 and 1000 layers are also available.

We usually stack some more layers in Deep Neural Networks to tackle a complicated issue, which improves accuracy and performance. The idea behind adding more layers is that these layers would learn increasingly complicated characteristics as time goes on. In the instance of picture recognition, the first layer may learn to recognise edges, the second layer might learn to identify textures, and the third layer would learn to recognise objects, and so on. However, it has been

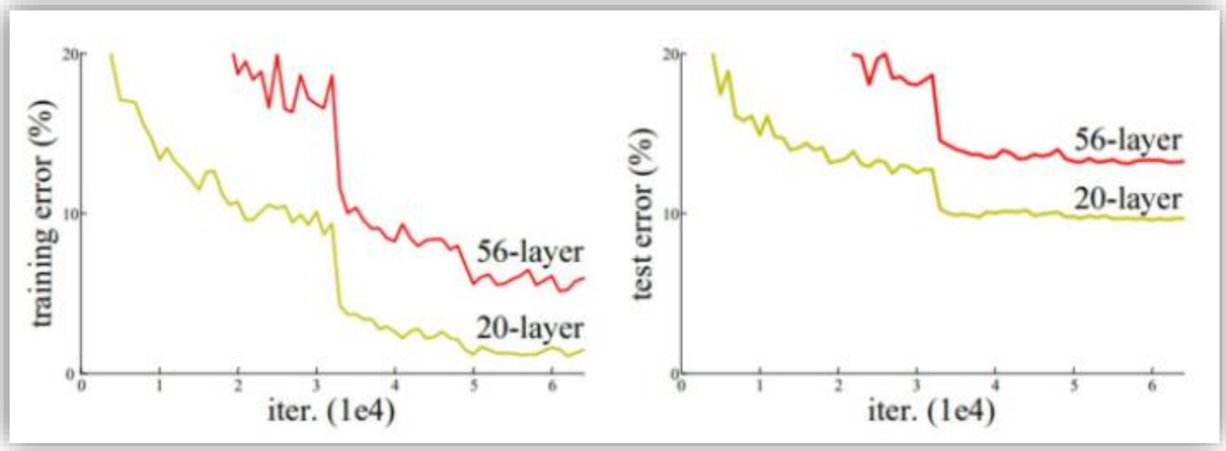


Figure 9 Error percent graph of ResNets

discovered that the conventional Convolutional neural network model has a maximum depth threshold. A graphic depicting error percent on training and testing data for a 20 layer Network and a 56 layer Network is shown below.

We can observe that the error percent for a 56-layer network is higher than a 20-layer network in both training and testing data. This implies that as a network's performance declines as additional layers are added on top of it. This might be attributed to the optimization function, network setup, and, most significantly, the vanishing gradient issue. You could believe it's because of overfitting, however the error percent of the 56-layer network is the worst on both training and testing data, which doesn't happen when the model is overfitted.

6.1.1 Residual Block

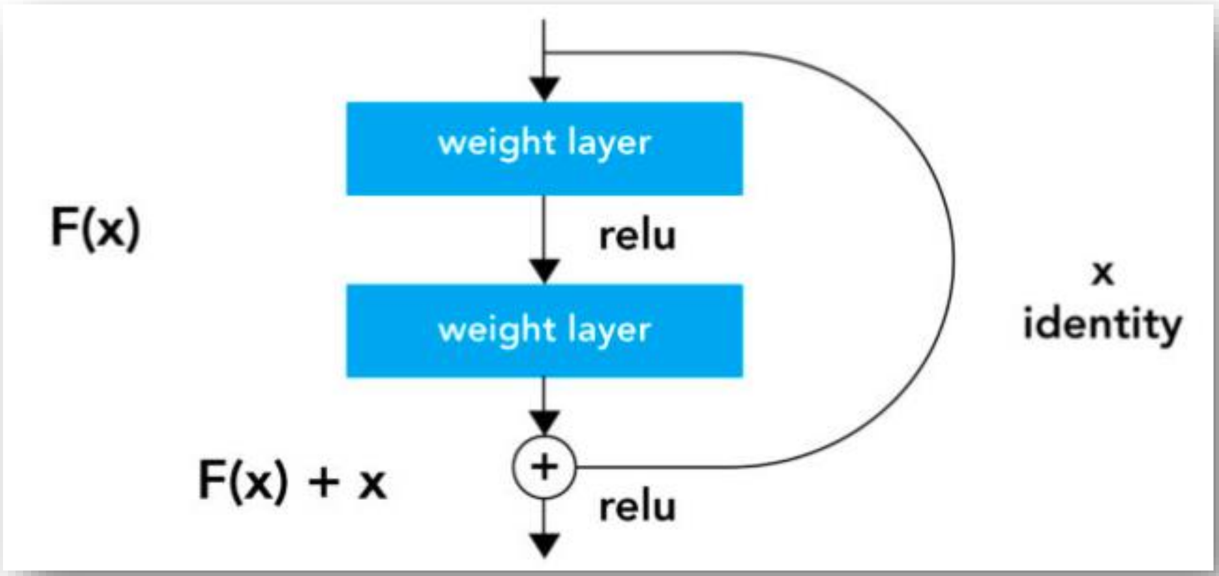


Figure 10 Structure of Residual Block

The first thing we notice is that there is a direct link that bypasses certain levels (which may change depending on the model) in between. The core of residual blocks is a link known as the'skip connection.' The output of the layer is no longer the same due to this skip connection. Without this skip link, the input 'x' is multiplied by the layer's weights, then a bias term is added.

The activation function, f(), is then applied to this term, and the result is H(x).

$$H(x) = f(\ w*x + b\ ) \tag{6.1}$$

or

$$H(x)=f(x) \tag{6.2}$$

The output has altered since the introduction of the skip connection.

$$H(x)=f(x)+x \tag{6.3}$$

When the dimensions of the input and output differ, as can happen with convolutional and pooling layers, there appears to be a little issue with this technique. When the dimensions of f(x) differ from x, we can use one of two approaches:

- To expand its dimensions, the skip connection is padded with extra zero entries.
- To match the dimension, the projection technique is employed, which involves adding 11 convolutional layers to the input. In this example, the result is:

$$H(x)=f(x)+w1.x \tag{6.4}$$

In this case, we add an additional parameter w1, but in the first technique, no new parameter is added.

ResNet's skip connections alleviate the problem of disappearing gradient in deep neural networks by enabling the gradient to flow along an additional shortcut channel. These connections also aid the model by helping it to learn the identity functions, ensuring that the upper layer performs at least as well as the lower layer, if not better.

6.1.2 ResNet50 Architecture

Each layer follows a similar pattern. They conduct 3x3 convolution with a fixed feature map dimension (F) of [64, 128, 256, 512] and bypass the input after every 2 convolutions. Furthermore, the width (W) and height (H) of the layer remain consistent throughout.

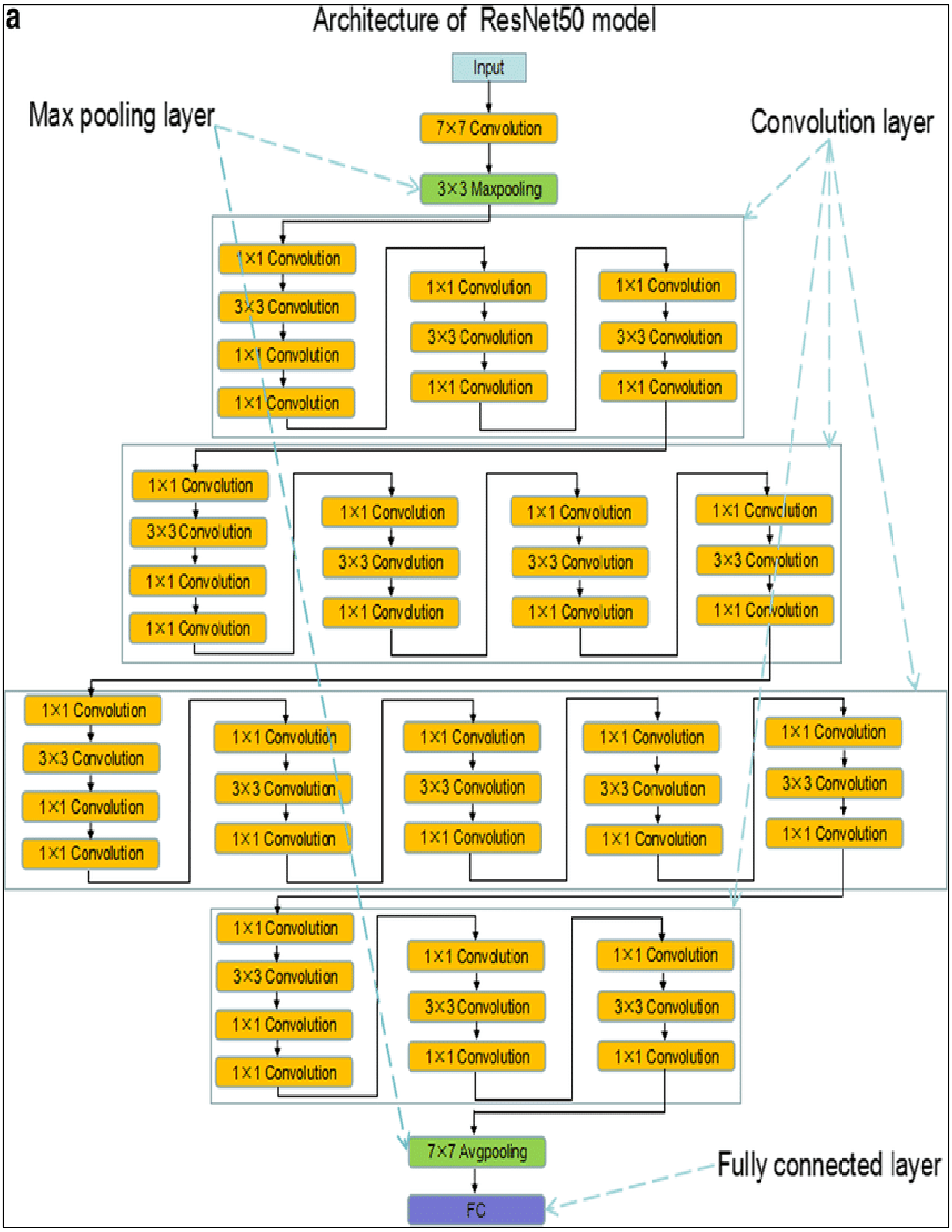


Figure 11 ResNet50 Architecture

6.2 DenseNet121

One of the most recent developments in neural networks for visual object detection is DenseNet. DenseNet is quite similar to ResNet, however there are a few key distinctions. DenseNet concatenates (.) the output of the previous layer with the output of the future layer, whereas ResNet employs an additive technique (+) that combines the previous layer (identity) with the future layer.

DenseNet was created particularly to address the vanishing gradient's effect on high-level neural networks' accuracy. Simply said, the information disappears before it reaches its destination due to the longer journey between the input and output layers.

6.2.1 DenseNet Structure

DenseNet is classified as a traditional network.

A 5-layer dense block with a k = 4 growth rate and the conventional ResNet structure is shown in this picture.

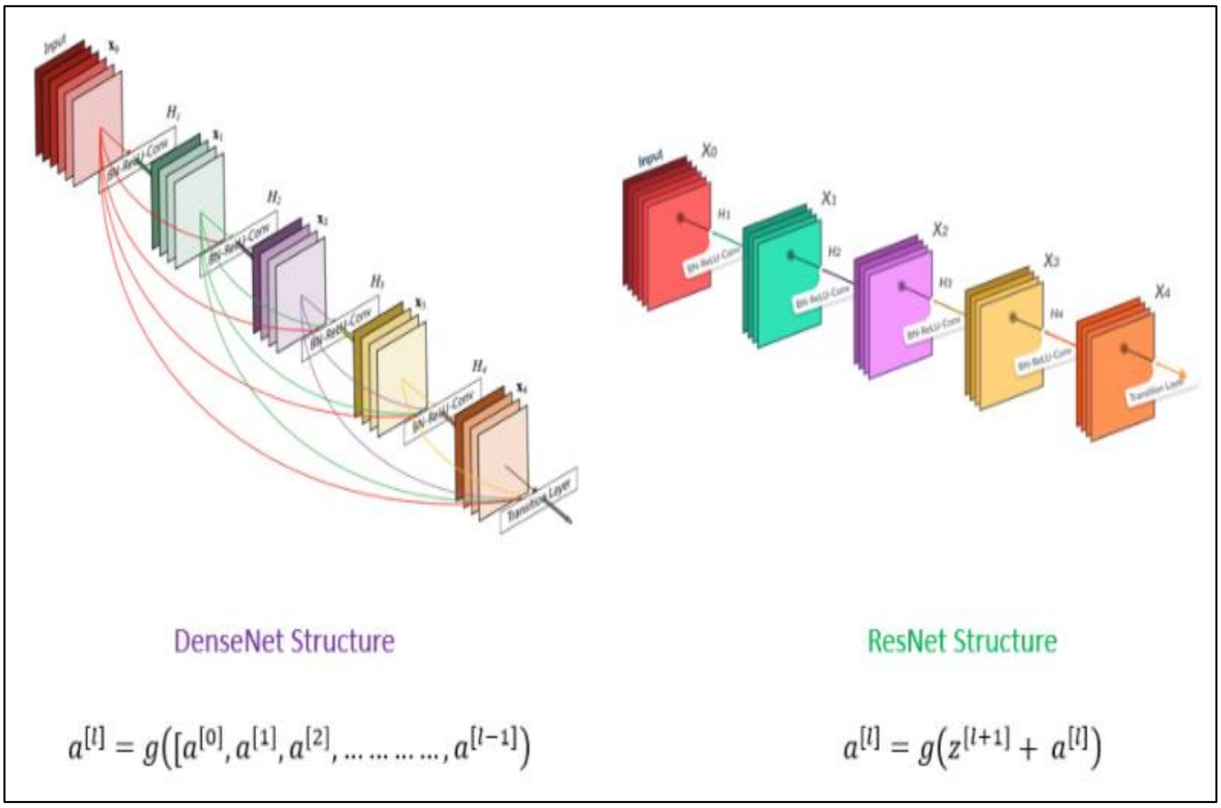


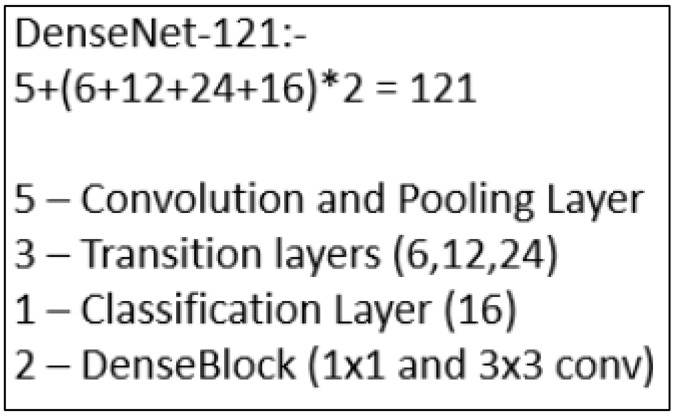
Figure 12 5-layer DenseBlock and Conventional ResNet

Sources: DenseNet Structure - G. Huang, Z. Liu and L. van der Maaten, “Densely Connected Convolutional Networks,” 2018; Resnet Structure.

By employing the composite function operation, an output of the previous layer becomes an input of the second layer. The convolution layer, pooling layer, batch normalisation, and non-linear activation layer make up this composite process.

The network has L(L+1)/2 direct connections as a result of these linkages. The number of levels in the architecture is denoted by the letter L.

DenseNet comes in a variety of variants, such as DenseNet-121, DenseNet-160, DenseNet-201, and so on. The numbers indicate how many layers there are in the neural network. This is how you get the number 121:



DenseNet-121:-
5+(6+12+24+16)*2 = 121

5 – Convolution and Pooling Layer
3 – Transition layers (6,12,24)
1 – Classification Layer (16)
2 – DenseBlock (1x1 and 3x3 conv)

Figure 13 Calculation of DenseNet Layers

6.2.2 Dense Blocks and Layers

The above equation can only group layers if the feature map dimensions are the same, whether adding or concatenating. What if the dimensions aren't the same? The DenseNet is split into DenseBlocks, each of which has a distinct set of filters but the same dimensions. The Transition Layer uses downsampling to do batch normalisation; it's an important stage in CNN.

Let's have a look at the contents of the DenseBlock and transition layer.

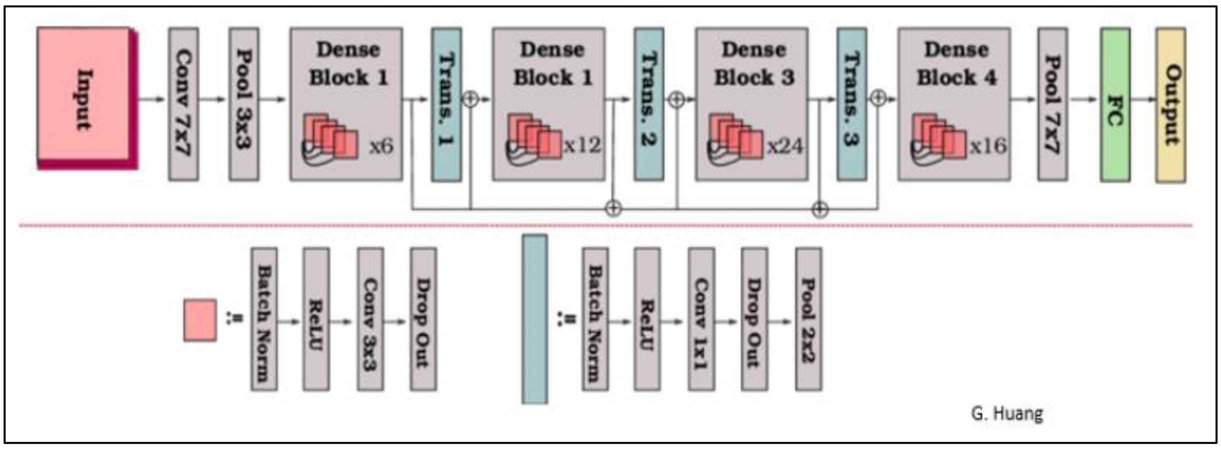


Figure 14 Short representation of DenseNet121 Architecture

Source: G. Huang, Z. Liu and L. van der Maaten, “Densely Connected Convolutional Networks,” 2018.

This is a representation of the entire architecture in abstract form.

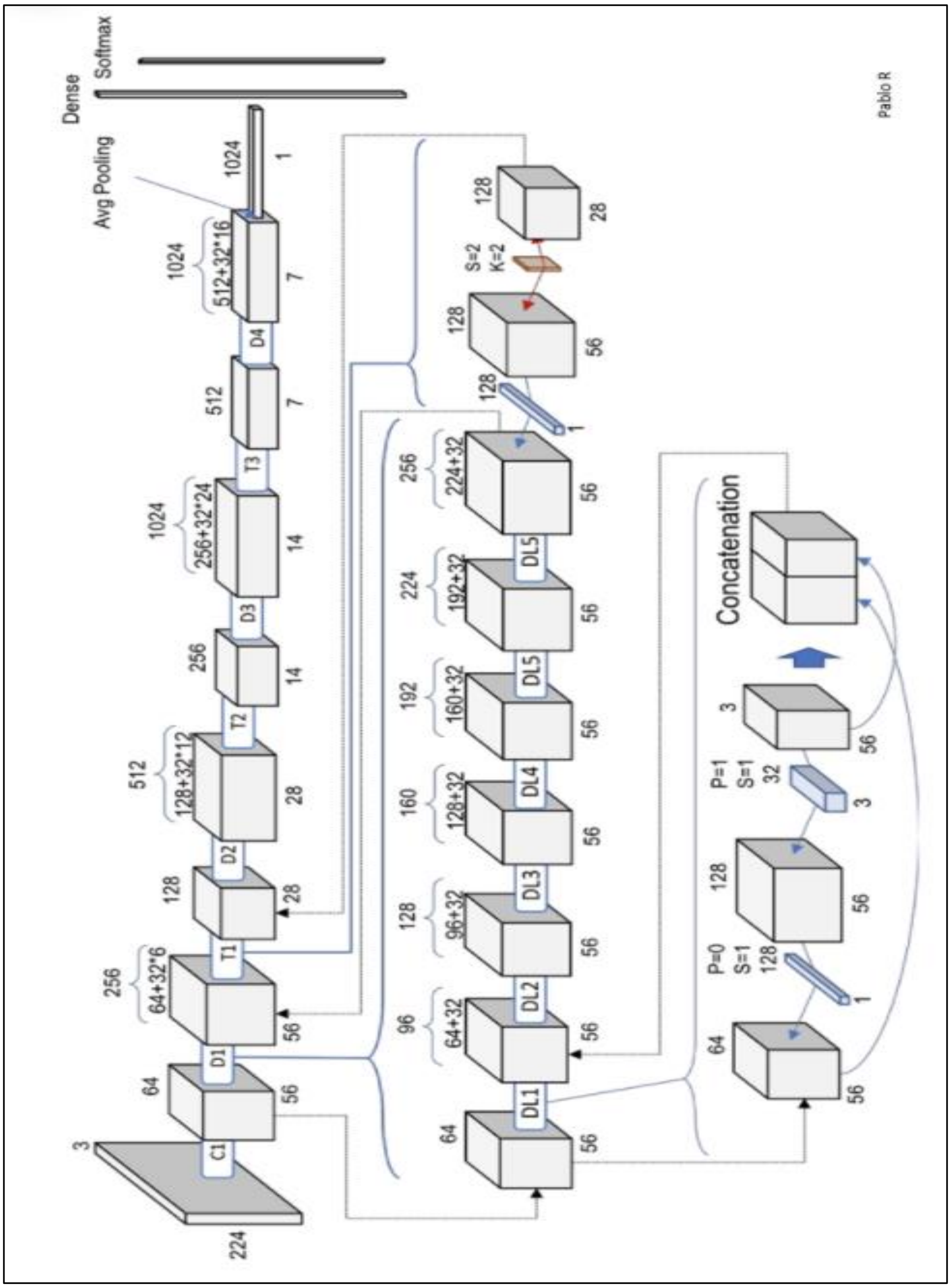


Figure 15 DenseNet121 architecture in abstract from

6.3 VGG16

In their publication "Very Deep Convolutional Networks for Large-Scale Image Recognition," K. Simonyan and A. Zisserman from the University of Oxford proposed the VGG16 convolutional neural network model. In ImageNet, a dataset of over 14 million pictures belonging to 1000 classes, the model achieves 92.7 percent top-5 test accuracy. It was a well-known model that was submitted to the ILSVRC-2014. It outperforms AlexNet by sequentially replacing big kernel-size filters (11 and 5 in the first and second convolutional layers, respectively) with numerous 33 kernel-size filters. VGG16 had been training for weeks on NVIDIA Titan Black GPUs.

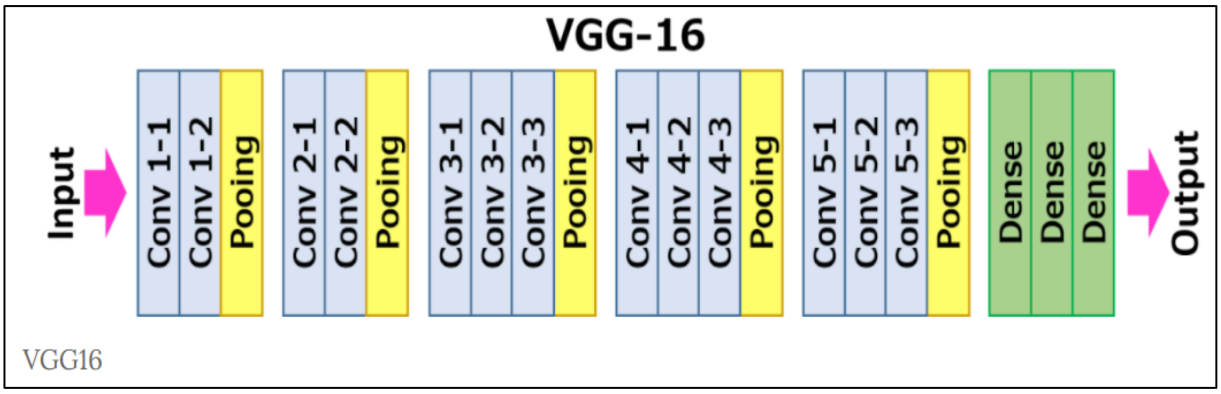


Figure 16 Short representation of VGG16 architecture

6.3.1 VGG Architecture

The input to the cov1 layer is a 224 by 224 RGB picture with a fixed size. The image is processed through a stack of convolutional (conv.) layers with an extremely tiny receptive field: 33 (the smallest size to capture the notions of left/right, up/down, and centre). It also uses 11 convolution filters in one of the setups, which may be thought of as a linear modification of the input channels (followed by non-linearity). The convolution stride is set to 1 pixel, and the spatial padding of conv. layer input is set to 1 pixel for 3X3 conv. layers so that the spatial resolution is retained after convolution. Five max-pooling layers, which follow part of the conv. layers, do spatial pooling (not all the conv. layers are followed by max-pooling). Max-pooling is done with stride 2 across a 22 pixel frame. Following a stack of convolutional layers (of varying depth in various designs), three Fully-Connected (FC) layers are added: the first two have 4096 channels apiece, while the third performs 1000-way ILSVRC classification and therefore has 1000 channels (one for each class). The soft-max layer is the last layer. In all networks, the completely linked levels are configured in the same way. The rectification (ReLU) non-linearity is present in all buried layers.

It should also be highlighted that, with the exception of one, none of the networks feature Local Response Normalization (LRN), which does not enhance performance on the ILSVRC dataset but increases memory consumption and computation time.

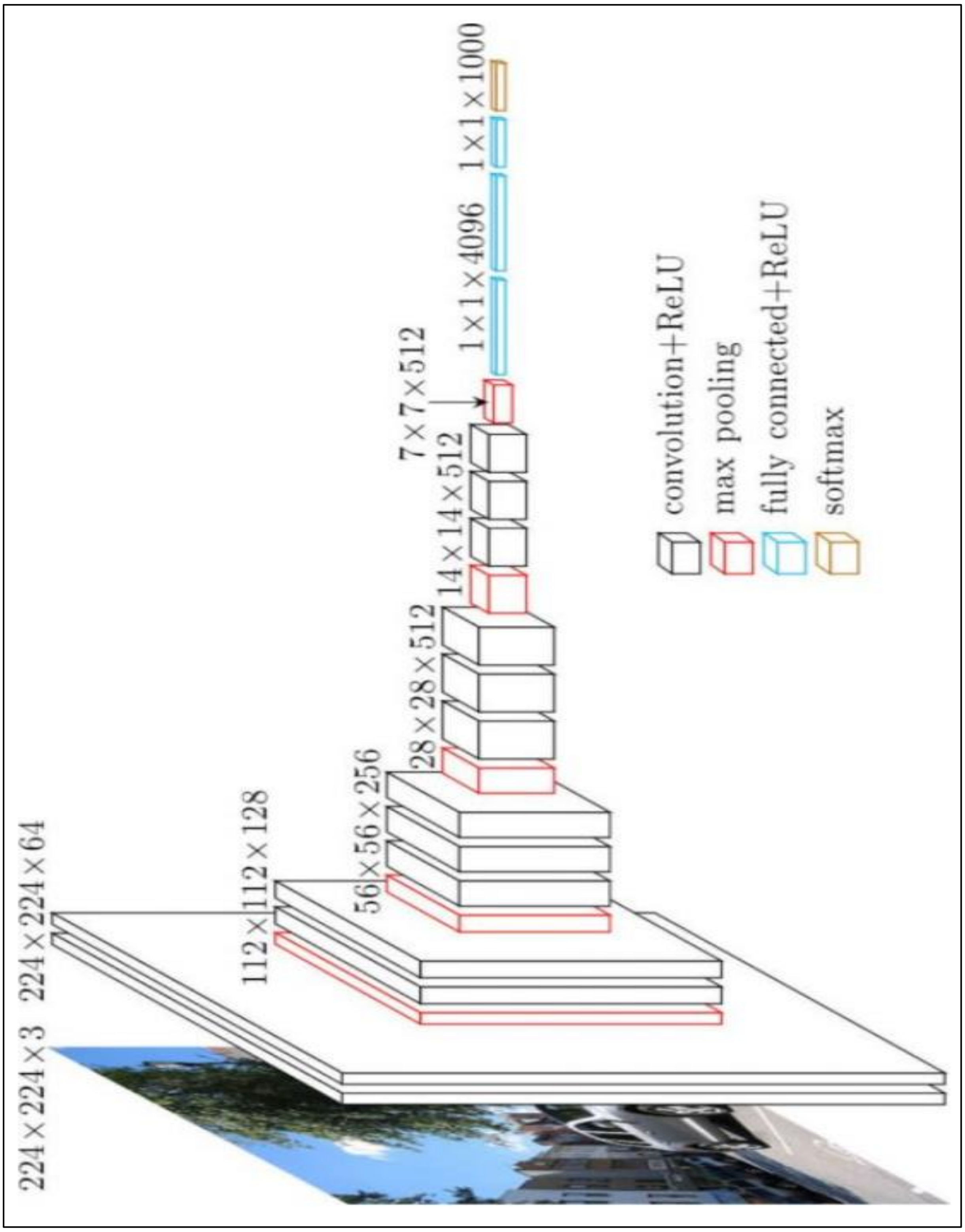


Figure 17 VGG16 architecture in abstract form

VGGNet, however, has two significant flaws:

- Training is very slow.
- The network architectural weights (in terms of disk/bandwidth) are fairly significant.

VGG16 is over 533MB in size because to its depth and amount of completely linked nodes. As a result, installing VGG is a time-consuming process. Many deep learning image classification challenges employ VGG16; nevertheless, smaller network designs are frequently preferred (such as SqueezeNet, GoogLeNet, etc.). However, because it is simple to execute, it is an excellent learning tool.

6.4 AlexNet

AlexNet is a convolutional neural network that has had a significant influence on machine learning, particularly in the application of deep learning to machine vision. It notably won the 2012 ImageNet LSVRC-2012 competition by a huge margin (15.3 percent mistake rates vs 26.2 percent error rates in second place). The network's design was quite similar to that of Yann LeCun et al's LeNet, but it was deeper, with more filters per layer and layered convolutional layers. Convolutions, max pooling, dropout, data augmentation, ReLU activations, and SGD with momentum were all part of it. After each convolutional and fully-connected layer, it added ReLU activations.

- To provide non-linearity, the Relu activation function is employed instead of Tanh. It increases the speed by 6 times while maintaining the same precision.
- To cope with overfitting, use dropout rather than regularisation. With a dropout rate of 0.5, however, the training time is doubled.
- Overlap pooling is a technique for reducing the size of a network. The top-1 and top-5 error rates are reduced by 0.4 percent and 0.3 percent, respectively.

6.4.1 AlexNet Architecture

The AlexNet has eight layers with weights, as shown in Figure 1, with the first five being convolutional and the final three being fully connected. The last fully-connected layer's output is sent into a 1000-way softmax, which generates a distribution across the 1000 class labels. The network aims to maximise the multinomial logistic regression goal, which is the average of the log-probability of the right label under the prediction distribution across all training examples. Only those kernel mappings in the preceding layer that are on the same GPU are linked to the kernels of the second, fourth, and fifth convolutional layers.

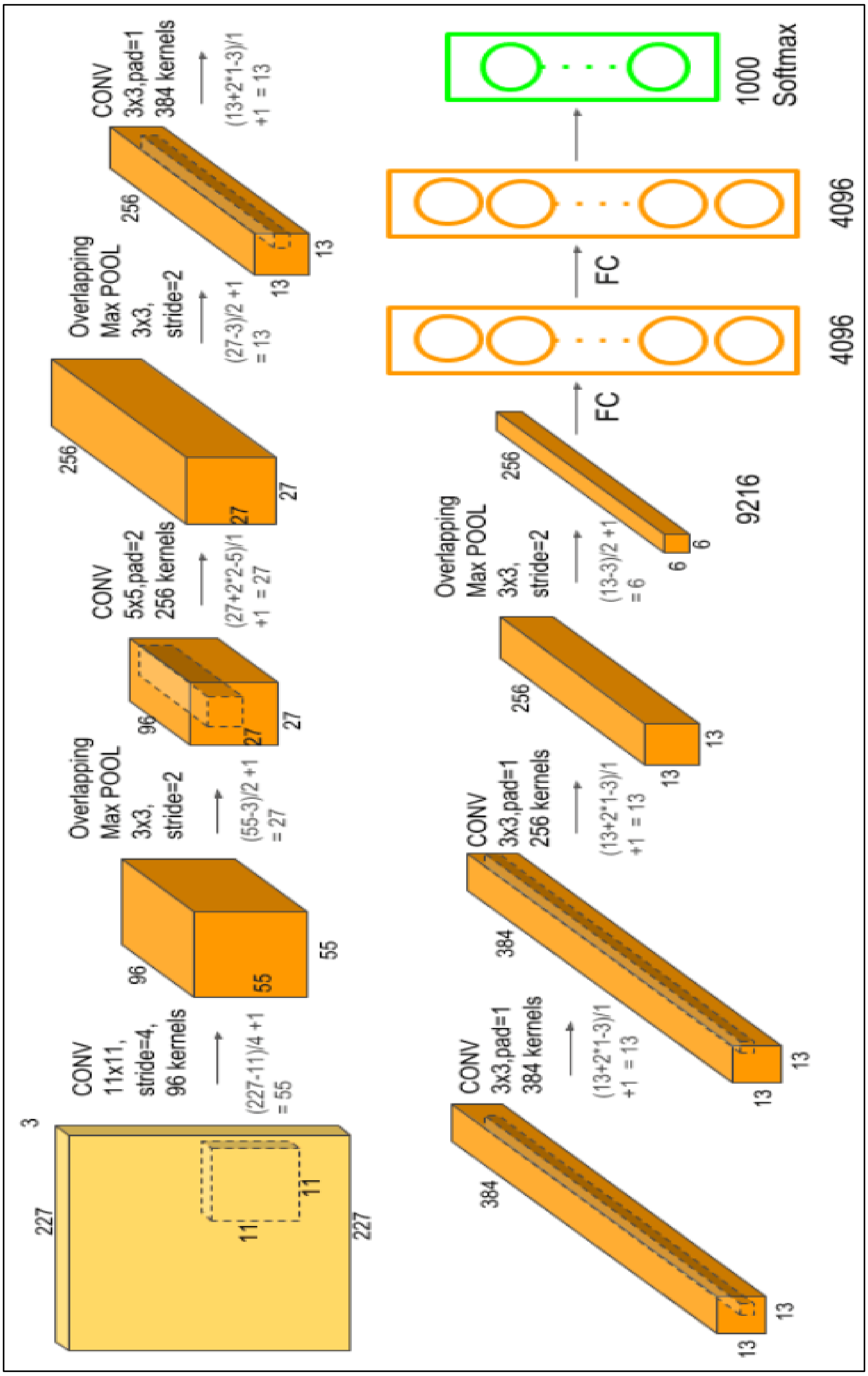

227
3
227
11
11
CONV 11x11, stride=4, 96 kernels
(227-11)/4 +1 = 55
96
55
55
Overlapping Max POOL 3x3, stride=2
(55-3)/2 +1 = 27
96
27
27
CONV 5x5,pad=2 256 kernels
(27+2*2-5)/1 +1 = 27
256
27
27
Overlapping Max POOL 3x3, stride=2
(27-3)/2 +1 = 13
256
13
13
CONV 3x3,pad=1 384 kernels
(13+2*1-3)/1 +1 = 13
384
13
13
CONV 3x3,pad=1 384 kernels
(13+2*1-3)/1 +1 = 13
384
13
13
CONV 3x3,pad=1 256 kernels
(13+2*1-3)/1 +1 = 13
256
13
13
Overlapping Max POOL 3x3, stride=2
(13-3)/2 +1 = 6
256
6
6
9216
FC
4096
FC
4096
1000 Softmax

Figure 18 AlexNet architecture as explained in the *[6]*

All kernel mappings in the second layer are linked to the kernels of the third convolutional layer. All neurons in the preceding layer are linked to the neurons in the fully-connected levels.

AlexNet is made up of five convolutional layers and three fully linked layers. After a highly convolutional and completely linked layer, Relu is applied. Before the first and second completely connected years, dropout is used. In a forward pass, the network contains 62.3 million parameters and requires 1.1 billion computing units. Convolution layers, which account for 6% of all parameters yet require 95% of the work, may also be seen.

AlexNet uses 90 epochs that were trained on two Nvidia Geforce GTX 580 GPUs concurrently for six days, which is why their network is split into two streams. The learning rate is 0.01, the momentum is 0.9, and the weight decay is 0.0005. Once the accuracy reaches a plateau, the learning rate is divided by ten. During the training, the learning rate is reduced three times.

CHAPTER 7 TRANSFER LEARNING AND HOW TO USE IT

Data science tasks such as training machine learning models may be difficult. The training algorithms may not function properly, training timeframes may be excessive, or training data may be insufficient. One of the strategies for making training easier is transfer learning. Transmit learning allows data scientists to transfer insights learned from one machine learning job to another, much as people may transfer their knowledge from one field to another. They can reduce machine learning model training time and rely on fewer data points as a result of this.

In order to implement knowledge transfer between tasks we humans are quite perfect. This is why we recognise and use our pertinent knowledge from past learning experiences when we meet a new difficulty or task. This simplifies and quickly completes our work. For example, if you know how to ride a bike and if a motorcycle you never did is requested to ride. In this instance, our biking experience comes in and handles responsibilities such as balancing, steering, etc. Compared to the complete beginner, this facilitates things. Such leanings make us far more flawless in actual life and allow us to gain more experience.

The term Transfer Learning in the field of machine learning was created in line with the same methodology. This strategy entails applying knowledge obtained in some tasks and solving the problem in the corresponding objective task. The creation of algorithms that promote transfer learning is an area of continuous attention in the learning field although most master training is geared to meet a single job.

Many deep neural networks with images share a unique phenomenon: a deep learning model attempts to learn a small degree of properties such as edges, colours, fluctuations in intensities and so on in early layers of the network. Such attributes do not appear specialised to a certain data set or task, since we either analyse the image to recognise a lion or cars for whatever sort of image. These low-level characteristics have to be detected in both circumstances. All these features are provided irrespective of the actual cost or image data package. In one task lion detection can therefore be used to learn these properties in other tasks, such as human detection. This is transmission learning.

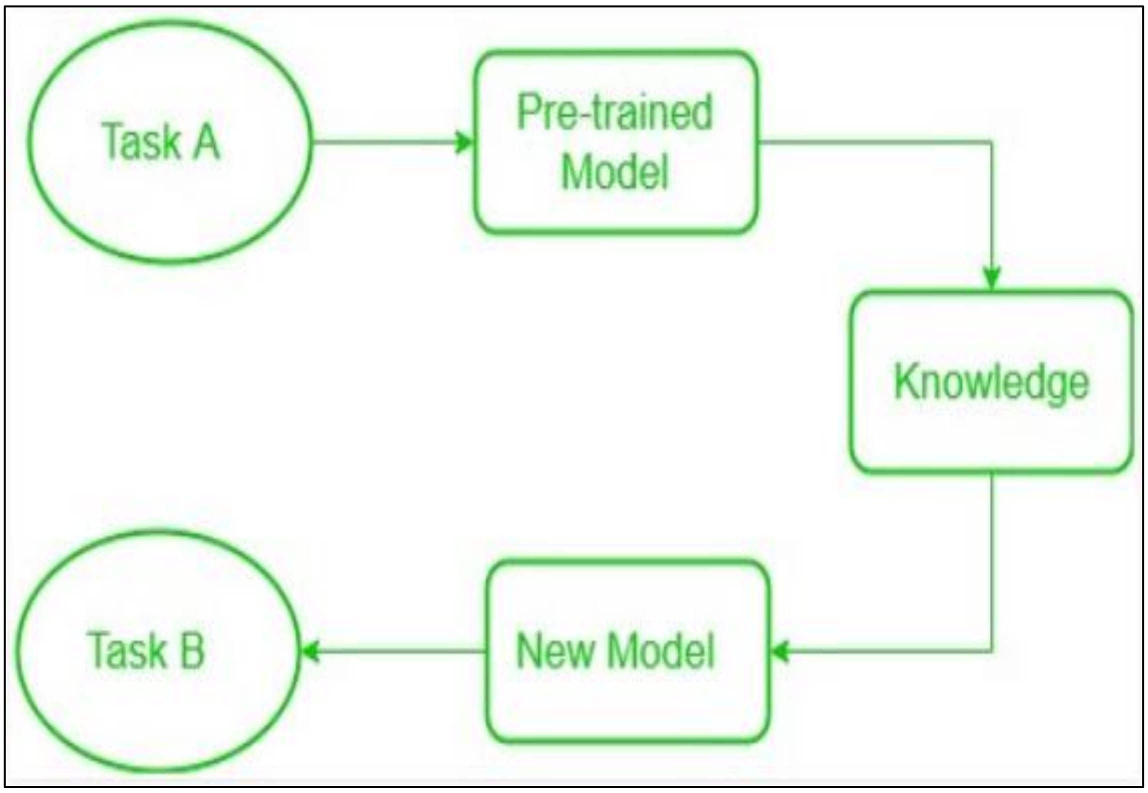


Figure 19 Base Idea of Transfer Learning

We come upon a phenomenon termed the freezing of layers while dealing with transfer learning. A layer may be a CNN layer, hidden layer, a layer block or any subset of all layer, if no longer available for training. It may be claimed that a layer is fixed. Thus, the weights of freezing layers during the training will not be updated. While the non-frozen layers are trained regularly.

We choose a pre-trained model as our basis for translation learning when solving an issue. Now, two ways of using knowledge from the pretrained model are feasible. First is to freeze some layers of pre-trained models and train our new dataset in other levels for the new task. Secondly, a new model is constructed, but also some layers features in the pre-trained model are taken and used in a new model. In both circumstances, the learning features are taken and the rest of the model trained. That means that the sole feature which can be identical in both tasks is removed from the pre-trained model, and by training the remainder of the model is turned into a new dataset.

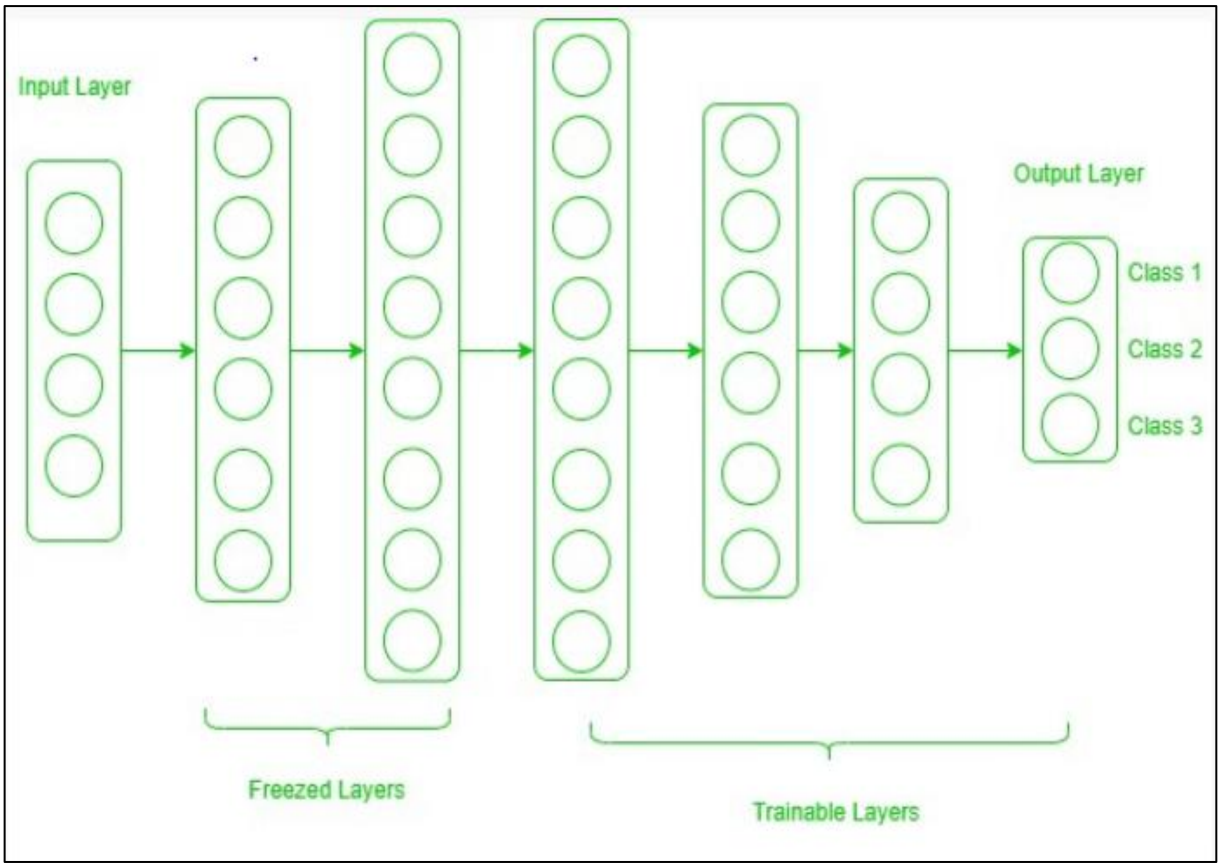


Figure 20 State of neural network layers in Transfer Learning

Now you can question how you can figure out which layers to freeze and which layers to train. The solution is simple, the more features you want from a pre-trained model to take over, the more layers you must freeze. For example, if the pre-trained model detects some floral species and certain additional species must be detected. In such a circumstance, there are many similar features to the pre-trained model in a new dataset with new species. We are therefore freezing fewer layers to utilise most of its information in a new model. If we want to use this knowledge to detect cars, then if the dataset is totally different, it's not good to freeze a lot of layers because freezing large numbers of layers will not only give low-level features, they can also provide high-level features like nose, eyes, etc, which are useless to new data (car detection). We therefore simply copy the low-level capabilities of the dataset and create a new data collection for the whole network.

Consider all circumstances where the target task's size and data collection are different from the base network.

- **Small and similar target data set to dataset basis network:** Since the target data set is tiny, we can improve the pretrained data set network. However, this can lead to an overfitting problem. Also, the number of classes in the target job may change somewhat. In this scenario, we remove from the end of the layers, perhaps one or two, additional layers that are fully connected, which satisfy a number of new classes. Now the remainder of the model is freezing and we're only training new layers.

- **The dataset target is broad and similar to the dataset base training:** There will be no danger of overpassing in cases where the data collection is huge and it is able to hold a pre-trained model. Here, a new fully integrated layer, with the appropriate number of classes, is likewise eliminated and a new totally connected layer. Now, the whole model has a new dataset training. This ensures that the model keeps the model identical on a new large dataset.

- **The target data set is small and differs from the dataset of the basic network:** As the objective dataset is different, it will not be useful to use high level characteristics of the pretrained model. In this scenario, remove most layers in a pre-trained model from the end and add new levels in a new dataset with the satisfactory amount of classes. This allows us to utilise low level characteristics from the pretrained model and to train the other layers for a new dataset. It is often useful, once you add a new layer, to train the whole network.

- **The target dataset is broad and different from the dataset in the base network:** The easiest technique to remove the layers from a pretrained network is to add layers which fulfil the classes. Then, the whole network is formed without frozen layers. Since the goal network is vast and various

Transfer learning is a powerful and quick technique to get a handle on a subject. It directs you in the right way; most of the time, transfer learning yields the best results.

these are some most popular architectures available. Residual Networks provide a decent combination of several parameters and performance and have faster training [3]. Another advantage of employing the ResNet architecture is the possibility to load input images of variable sizes than those with which they are usually trained. This is a crucial part of the training methods used to train a high-performance network with a small number of epochs utilizing the FastAI approaches. [18]. DenseNets offer many compelling advantages such as removing the problem of vanishing gradient [4] and improved propagation of features, feature reuse, and a significant reduction in the number of parameters. VGG-16 was one of the top-performing architectures in the 2014 ILSVRC competition. It finished second in the classification challenge, with a top-5 classification error of 7.32 percent (only behind GoogLeNet with a classification error of 6.66 percent). It also won the localization job, with a localization error of 25.32 percent.

Using only supervised learning, Alexnet was able to achieve record-breaking performance on a highly challenging dataset. It's worth noting that removing a single convolutional layer reduces the performance of this network [6].

Each model's weights are pre-trained using the ImageNet dataset [19]. The original dimensions of these images vary from 119×104×3 to 416×512×3 [11]. These images are rescaled to 128×128×3, 150×150×3, and 224×224×3 and used in various phases of training (Subsection A). We normalize the images using the mean and standard deviation of the images present in the ImageNet dataset for each RGB channel. We also leverage the pre-trained weights of a network that has previously been trained using the ImageNet dataset [19].

The trained head of the model is substituted with another head, including a sequence of adaptive average/max pooling, batch normalization, drop out, and linear layers for transfer learning, as described in [18].

CHAPTER 8 TRAINING THE NETWORK WITH PROGRESSIVE RE-SIZING AND L2-REGULARIZATION

Progressive resizing is a technique that involves fine-tuning the network using smaller images at first, then gradually increasing the input image size as the training advances. This is possible because the characteristics learned by the successive CNN layers are not affected by the size of the input image. In addition, the same resized image with multiple pixel resolutions retains the global features. The training is divided into three stages, each of which corresponds to images of varying input dimensions.

8.1 TRAINING STAGES

1st stage: We resize the input images to 128×128×3 pixels. Training is done only to the newly joined head of the network while retaining the ImageNet weights for the rest of the body for six epochs with a learning rate (2e-2).

2nd stage: After the results from the first stage the head of the model is again fine-tuned with images rescaled to 150×150×3 pixels for six epochs and with a learning rate of (5e-4).

3rd stage: We further finetune the whole network with input images of size 224×224×3 for ten epochs in the final stage. In this case, a slice of learning rates is used, where the training of the earliest layer is performed with a learning rate of (1e-5), and the final layer is trained with a learning rate of (1e-4). The layers between the first and the final are trained with the learning rates, which are equally spaced between the two values.

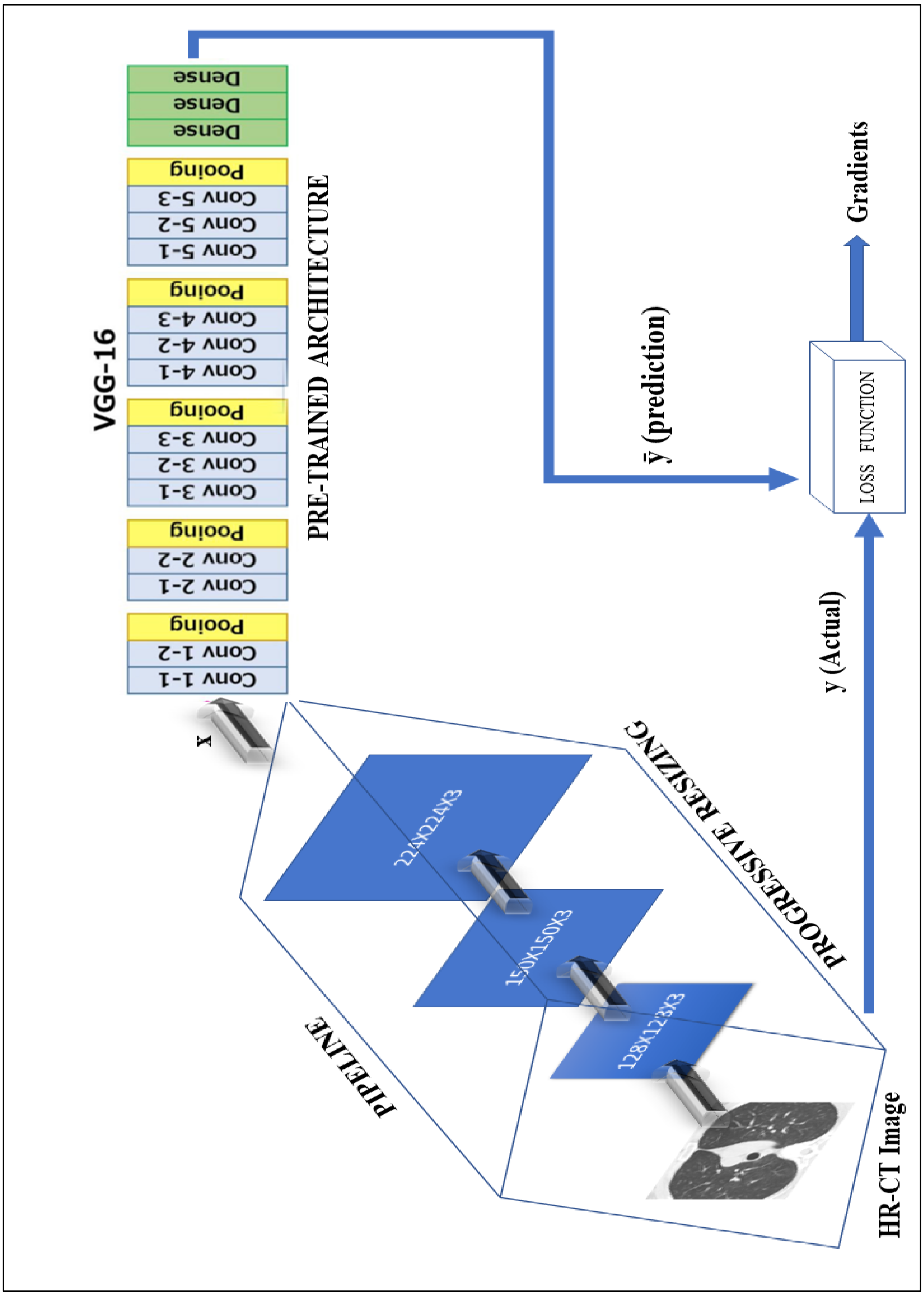


Figure 21 Proposed method Flow Diagram

Multiple stage training of the model with input image of variable sizes along with progressive resizing leads to achieving better results and diagnosis This is also an example of how learning is transferred from one image size to another [20]. It is noted that we ensure the learning rates are depreciated as we progress through the training. This ensures that the weights do not change significantly from one step to the next. In training, a batch size of 128 with Adam optimizer is utilized. The FastAI [18] framework was used for all data preprocessing, data augmentation, and training.

8.2 LEARNING RATE SELECTION

We pass the loaded data, specify the model, and specify error rate, accuracy, F1-score [21], and Area Under Curve(AUC) receiver operating characteristic (ROC) [22] as a list for the metrics parameters in the CNN learner function available in FastAI library. Instead of manually modifying learning rates at each stage, we employ Leslie Smith's Cyclical Learning Rate technique described in [23] to assist in selecting optimal learning rates. The best learning rate should be chosen as the learning rate value where the Loss versus Learning rate curve is the steepest. For an image size of 128X128X3 on plotting the loss versus learning rate curve,

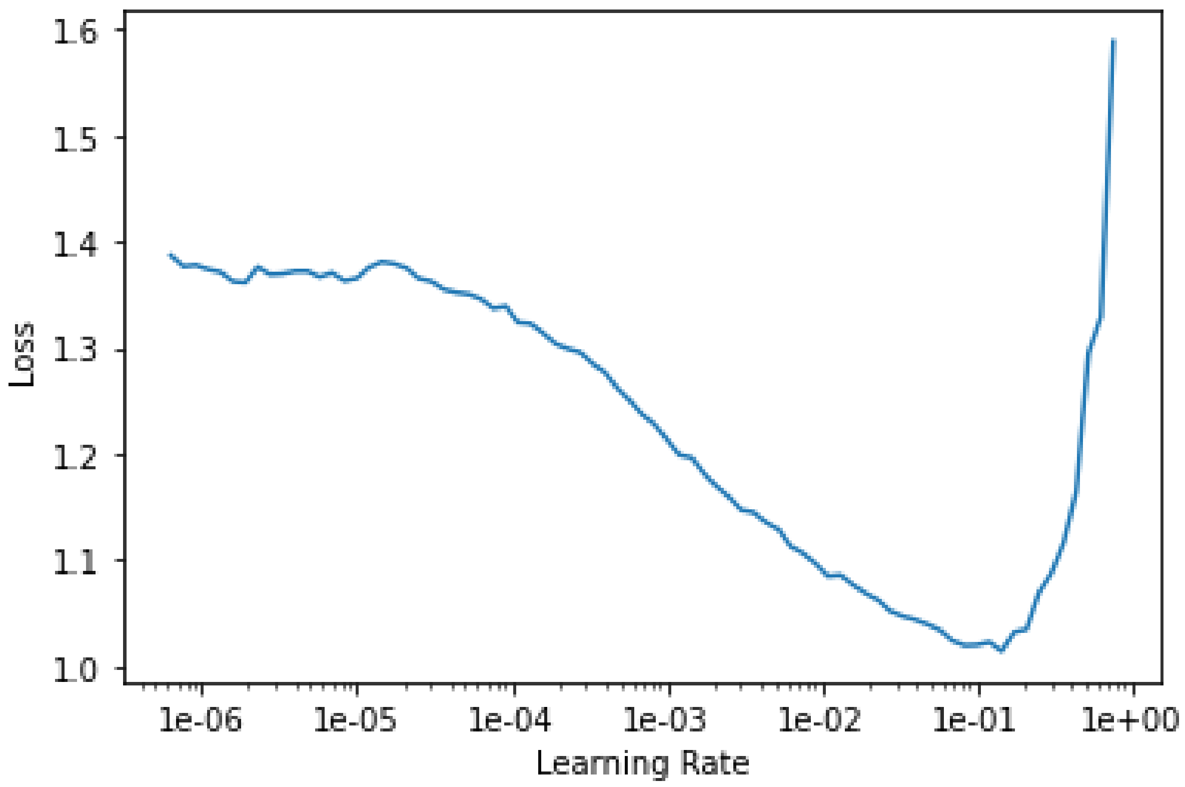


Figure 22 Plot of Cyclical Learning rate finder graph for image size 128×128×3

we can see that the derivate of the curve is least in the slice of (1e-3) to (1e-1) as shown in Figure 4.

In every pre-trained architecture on plotting the above curve for different image sizes, we find pretty close learning rate values hence; we have used the following learning rates in Table I for various image sizes.

TABLE I. LEARNING RATES FOR VARIOUS INPUT SIZES

Input Image size	Learning rate
128×128×3	2e-2
150×150×3	5e-4
224×224×3	Slice(1e-5,1e-4)

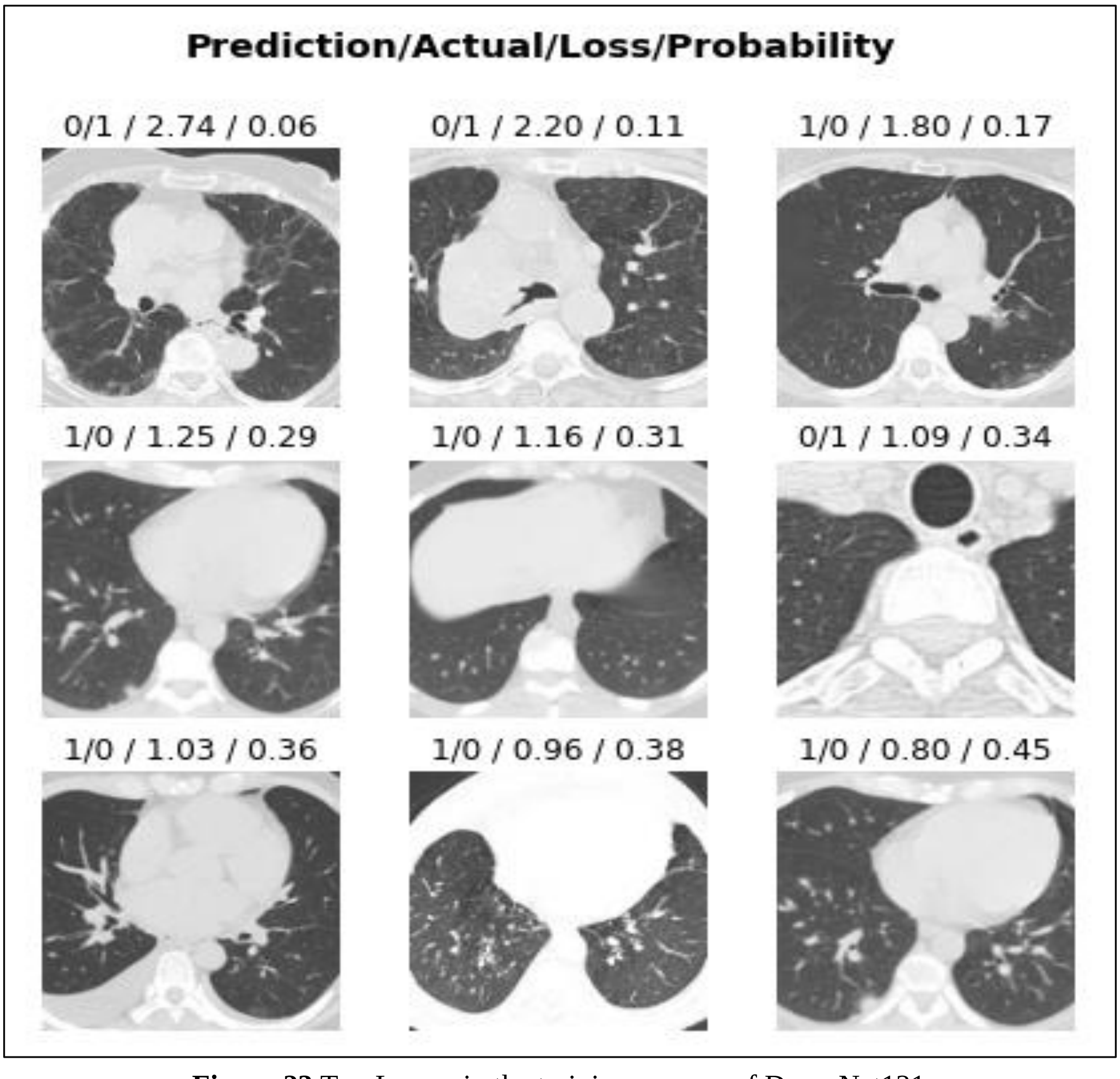


Figure 23 Top Losses in the training process of DenseNet121

The parameter slice allows the use of a discriminative learning rate. We apply a lower learning rate to the first layer and a higher learning rate to the last layer to reduce the training time. A split of 80:20 (train: validation) is used in training. Learning aims to take advantage of a pre-trained model's ability to recognize specific patterns and adapt to our dataset.

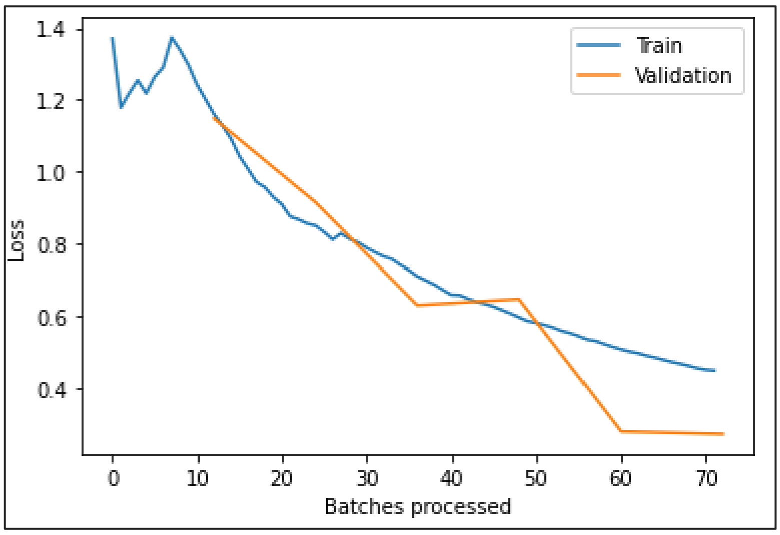
The top losses in the training process generated by DenseNet121 are shown in Figure 9.

8.3 FURTHER MODEL OPTIMIZATION WITH L2 REGULARIZATION

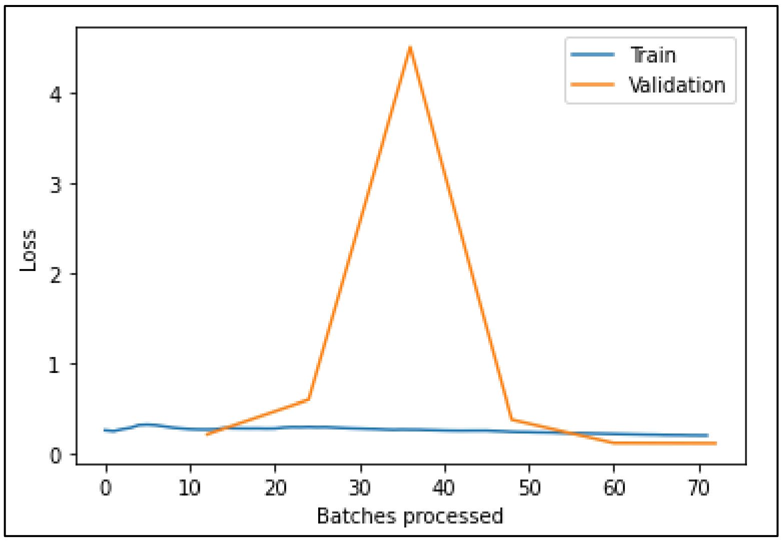
L2 regularization [24] is one of the Deep Learning regularization techniques. It keeps our model from becoming overly complex by penalizing complexity. It does so by summing up the squares of all parameters in the loss function. However, this can result in such an enormous loss that the best model would set all the parameters to 0. To prevent this, we multiply the sum of squares by another small number **wd** called weight decay as shown in equation 1.

$$\textbf{Loss} = \textbf{cross-entropy}\ (\bar{y}, y) + \textbf{wd} * \sum \textbf{parameter}^{2} \qquad (8.1)$$

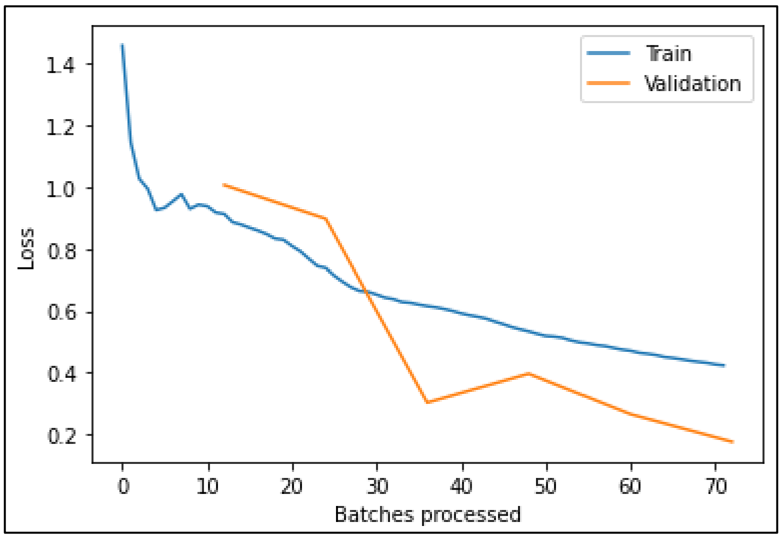
We use a value of 0.1 for weight decay. The loss plot of all the pre-trained architectures after the third stage are shown in figure . The training and validation losses can be seen converging.



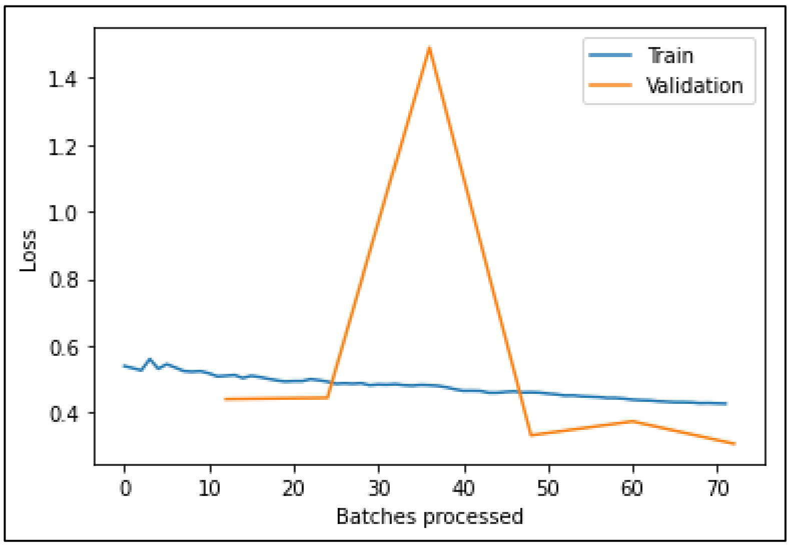
ResNet50



VGG16



DenseNet121



AlexNet

CHAPTER 9 PERFORMANCE EVALUATION

We use a set of different performance evaluation measures to evaluate our models. For each model, the number of predicted cases are divided as true positives (TP) (row:1, column:1), true negatives (TN) (row:0, column:0), false negatives (FN) (row:1, column:0) and false positives (FP) (row:0, column:1) in the confusion matrices in Figure 7. The metrics in 2, 3, and 4 are then calculated below on the validation set with the help of the confusion matrix.

A **confusion matrix** is a tabular representation of your prediction model's performance. The number of predictions produced by the model where it categorised the classes correctly or erroneously is represented by each entry in a confusion matrix.

Anyone who is familiar with the confusion matrix is aware that it is frequently used to explain a binary classification problem. This explanation, on the other hand, is not one of them. We'll look at how a confusion matrix works with multi-class machine learning models today. However, to put things in perspective, we'll start with some background information and a binary classification.

There are just two classes to categorise in a binary classification task, preferably a positive and a negative class.

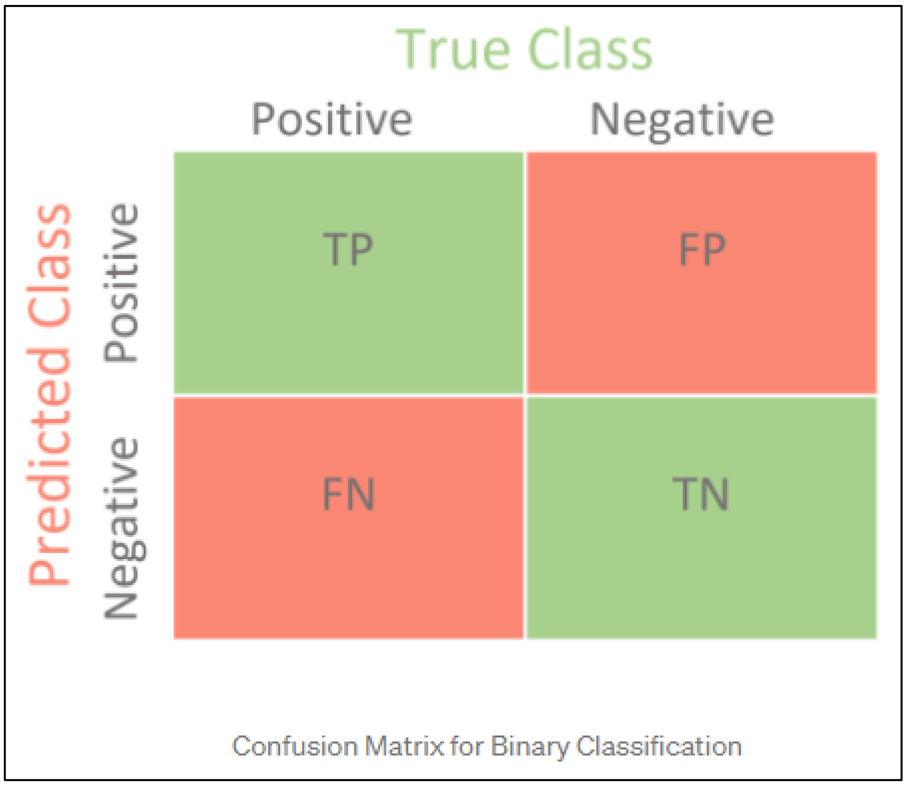


Figure 24 Confusion matrix for a Binary Classification model

True Positive (TP): It's the number of times the classifier has successfully predicted the positive class as positive.

True Negative (TN): It's the number of times the classifier has successfully predicted the negative class as negative.

False Positive (FP): This refers to the number of times the classifier has predicted the negative class as positive.

False Negative (FN): It's the number of times the classifier gets the positive class wrong and predicts the negative class.

9.1 EVALUATION METRICS

9.1.1 Error Rate: It tells you what percentage of your predictions were wrong. It's also referred to as a Classification Error. You can figure it out by using:

$$\text{Error rate} = \frac{\text{FP} + \text{FN}}{\text{TP} + \text{TN} + \text{FN} + \text{FP}} \qquad (9.1)$$

9.1.2 Accuracy: It shows you the model's overall accuracy, which is the percentage of total samples correctly identified by the classifier. Use the following formula to calculate accuracy:

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FN+FP}} \qquad (9.2)$$

9.1.3 F1-Score: The weighted average of Precision and Recall is the F1 Score. As a result, this score considers both false positives and false negatives. Although it is not as intuitive as accuracy, F1 is frequently more useful than accuracy, especially if the class distribution is unequal.

$$F_1 - \text{ score } = \frac{\text{2TP}}{\text{2TP+FP+FN}} \qquad (9.3)$$

9.1.4 Area Under receiver-operating-characteristic curve (AUC-ROC)

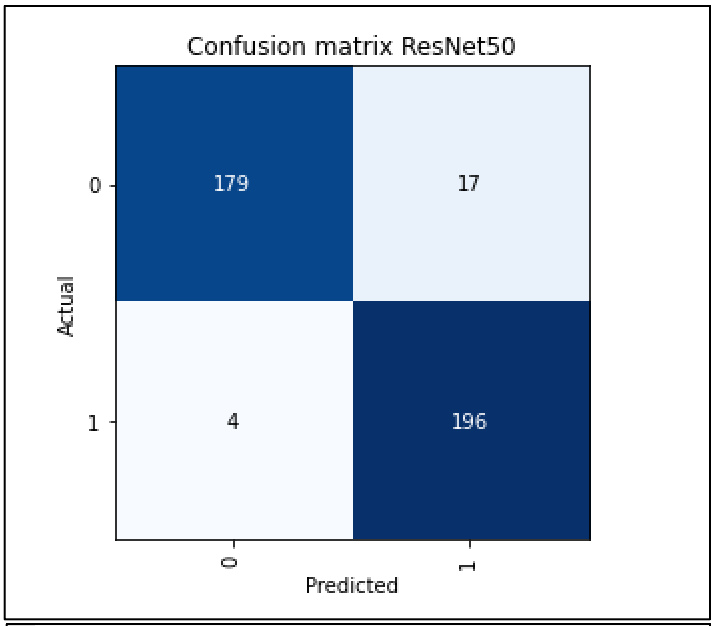
The receiver-operating-characteristic curve (ROC curve) [22] depicts a classification model's performance across all classification levels. The curve compares the false positive rate (1-specificity) to the true positive rate (recall). The area under the receiver operating characteristic curve (ROC-AUC) is a composite measure of performance across all conceivable classification thresholds. A model with a high ROC-AUC is superior at identifying true negative HRCT scans as negatives and true positive scans as positives.

9.2 RESULTS ANALYSIS

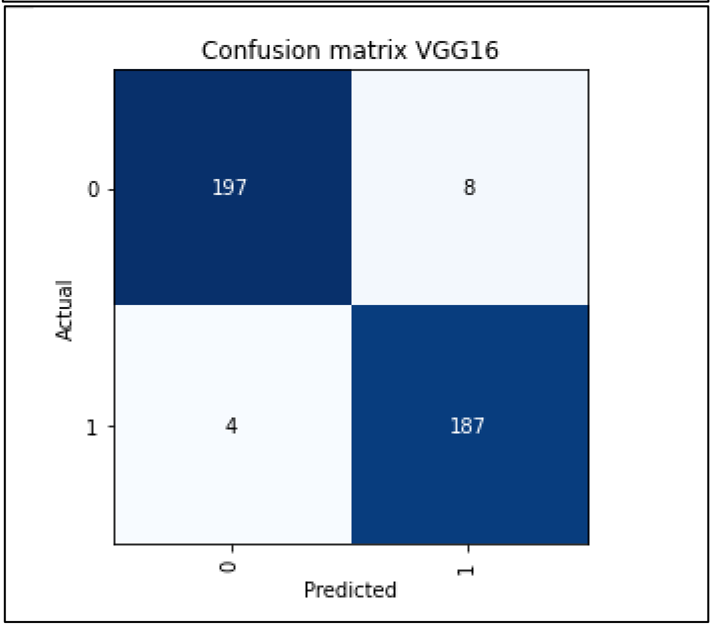
On the HR-CT dataset [12], Table 2 features the values of evaluation metrics attained by various pre-trained models. The best outcomes are bolded, and all values are provided in decimals. When appropriate, we also compare our findings to those previously published in the literature on the same dataset in Table 3. With a validation accuracy of 97.4% and an F1-score of 0.975 DenseNet121 [4] gets the best overall performance in practically all evaluation metrics. Other Pre-trained architectures used in this paper are ResNet50 [3], VGG16 [5], and AlexNet [6]. Four Different architectures are used here to test the robustness of Progressive resizing. CNN's can discover broad-level patterns like curves or edges using smaller images, which is why progressive image resizing is so effective. Furthermore, because 128 and 150-pixel pictures use less GPU memory than 224-pixel pictures, mini-batch sizes may be larger, and epochs may be completed much quicker, saving time.ResNet50 gives a validation accuracy of 94.6%. These values are the results of training the images AFTER 3RD STAGE. The accuracy of VGG16 is also close to DenseNet121, comes out to be 96.9%. AlexNet remains the least accurate model with a decent accuracy of 89.3%. One thing is to note that all evaluation methods improve in every architecture after each training stage, which shows that the progressive resizing of the image is essential in the improvement of deep neural networks training

TABLE II Performance of various models with progressive resizing

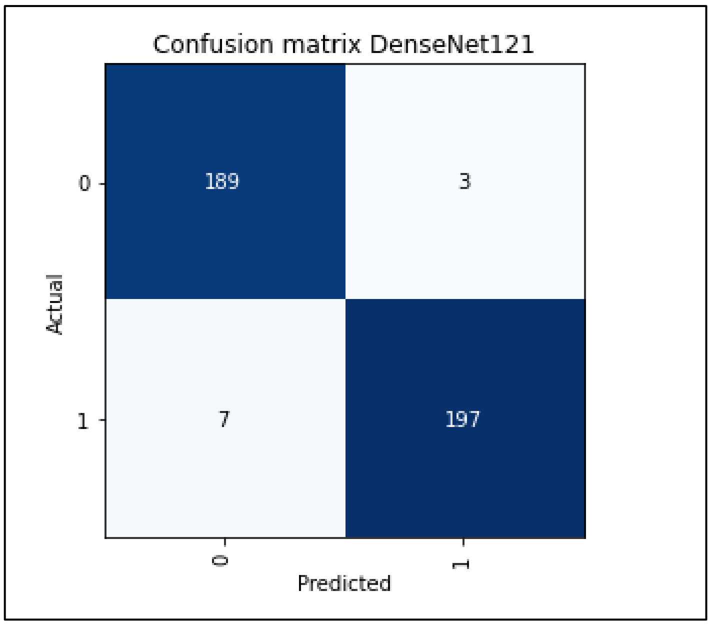
Model	*Input Image Size*	*Epochs*	*Error rate*	*Accuracy*	*F1 Score*	*AUC ROC*
ResNet50	128X128X3	6	0.095	0.904	0.907	0.960
	150X150X3	6	0.078	0.921	0.925	0.979
	224X224X3	10	0.053	0.946	0.949	0.990
DenseNet 121	128X128X3	6	0.068	0.931	0.936	0.978
	150X150X3	6	0.053	0.946	0.947	0.990
	224X224X3	10	**0.025**	**0.974**	**0.975**	**0.998**
VGG16	128X128X3	6	0.070	0.929	0.931	0.976
	150X150X3	6	0.032	0.967	0.968	0.989
	224X224X3	10	0.030	0.969	0.968	0.993
AlexNet	128X128X3	6	0.174	0.825	0.829	0.910
	150X150X3	6	0.121	0.878	0.874	0.958
	224X224X3	10	0.106	0.893	0.901	0.964



Confusion matrix ResNet50
179
17
4
196
0
1
Actual
Predicted



Confusion matrix VGG16
197
8
4
187
0
1
Actual
Predicted



Confusion matrix DenseNet121
189
3
7
197
0
1
Actual
Predicted

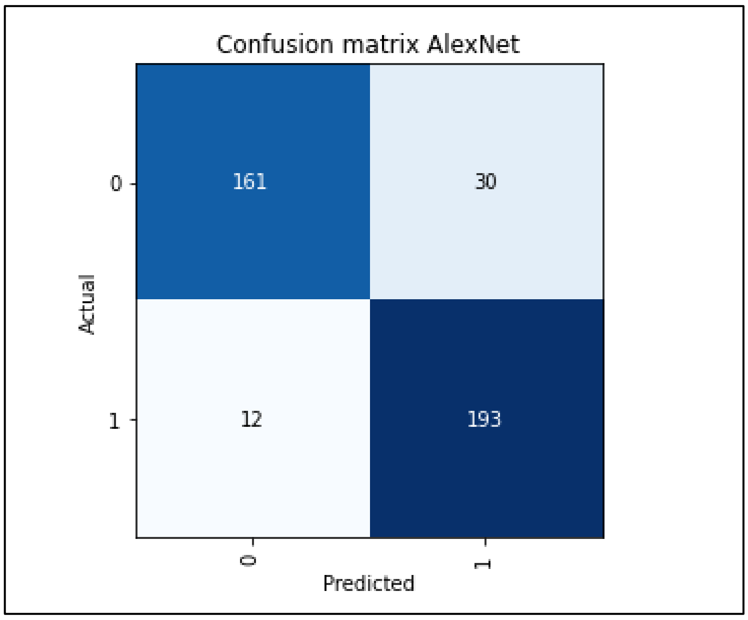


Figure 25 Confusion matrices of all architectures after the third stage of training

9.3 COMPARISON WITH RECENT RESEARCH WORK ON THE SAME DATASET

Panwar et al. [25] presented a deep transfer learning system that uses gradient weighted class activation mapping (Grad-CAM). It achieved 95 percent accuracy and a value of 0.943 for F1-score. This method uses only one type of CNN to perform the detection, resulting in lower scores than our approach. The authors of [10] updated the COVID-Net architecture and learning process for its usage with CT images. A collaborative learning strategy was developed to optimize the diagnosis of COVID-19 instances and to address data dissimilarity in the CT scan datasets employed. Experiments on two CT image datasets reveal that the suggested joint learning strategy is thriving, with 90.83 percent accuracy and 85.89 percent sensitivity. Jaiswal et al. [9] tested the detection of COVID-19 from CT lung scans using a DenseNet201 architecture. Because of the 201-layer depth, the usage of a DenseNet-201 enables the extraction of complicated features for classification while laying off the vanishing gradient problem. The DenseNet-201 base is merged with an artificial neural network consisting of two hidden layers of 64 and 128 nodes with ReLU activation functions, and an output layer of 2-node softmax. The SARS-CoV-2 dataset was used in this study. The DenseNet-201 accuracy is said to be 96.2 percent in the paper. However, our strategy employs a DenseNet version with only 121 layers, but still marginally outperforms their accuracy due to progressive resizing. Additionally, DenseNet121 significantly reduces training time and GPU consumption.

TABLE III . RECENT WORK ON COVID DETECTION AND COMPARISON WITH THE PROPOSED METHOD

Model	Accuracy	F1-Score
DenseNet201 [9]	96.2	96.2
Modified VGG19 [25]	95.0	94.3
COVID CT-NET [26]	-	90.0±0.1
Contrastive Learning [10]	90.8±0.9	90.0±1.3
Progressive Re-sizing	97.4	97.5

CHAPTER 10 CONCLUSION AND FUTURE WORK

10.1 CONCLUSION

During the pandemic and, more specifically, in the future, this COVID-19 detection model has the potential to have a significant effect on clinical workflows, such as in diagnostic purposes in every healthcare system. Vaccination and distancing are the most important things right now, and deep learning methods are the best option. The approach in this paper shows the effectiveness of progressive resizing and pre-trained architectures, to give an accuracy of 97.4 percent on a specifically larger dataset with excellent values of F1-score and AUC-ROC.The proposed approach has been accepted in the upcoming **2021 IEEE 2ND GCAT, BANGALORE CONFERENCE (IEEE CONFERENCE ID: 52182) From 01st to 03rd OCTOBER 2021.**

10.2 FUTURE WORK

The Future work lies in using a larger dataset to the same technique since 1252 images are not adequate for a very accurate classification.Reducing the training time with larger dataset is another challenge.Training the network with mixed-precision may help in this case. Our future goal of this project is to apply this method to a dataset having millions of images and to improve the evaluation parameters.Our main goal is to make the approach more reliable and efficient.

References

[1] "Worldometer," [Online]. Available: https://www.worldometers.info/coronavirus/.

[2] T. Ai, Z. Yang, H. Hou, C. Zhan, C. Chen, W. Lv, Q. Tao, Z. Sun and L. Xia, "Correlation of Chest CT and RT-PCR Testing for Coronavirus Disease 2019 (COVID-19) in China: A Report of 1014 Cases," *Radiology,* vol. 296, pp. E32-E40, 2020.

[3] K. He, X. Zhang, S. Ren and J. Sun, *Deep Residual Learning for Image Recognition,* 2015.

[4] G. Huang, Z. Liu, L. van der Maaten and K. Q. Weinberger, *Densely Connected Convolutional Networks,* 2018.

[5] K. Simonyan and A. Zisserman, *Very Deep Convolutional Networks for Large-Scale Image Recognition,* 2015.

[6] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Commun. ACM,* vol. 60, p. 84–90, 5 2017.

[7] T. Wang, Y. Zhao, L. Zhu, G. Liu, Z. Ma and J. Zheng, "Lung CT image aided detection COVID-19 based on Alexnet network," in *2020 5th International Conference on Communication, Image and Signal Processing (CCISP),* 2020.

[8] V. Positano, A. K. Mishra, S. K. Das, P. Roy and S. Bandyopadhyay, "Identifying COVID19 from Chest CT Images: A Deep Convolutional Neural Networks Based Approach," *Journal of Healthcare Engineering,* vol. 2020, p. 8843664, 2020.

[9] V. Chahar, A. Jaiswal, N. Gianchandani, D. Singh and M. Kaur, "Classification of the COVID-19 infected patients using DenseNet201 based deep transfer learning," *Journal of biomolecular Structure & Dynamics,* 7 2020.

[10] Z. Wang, Q. Liu and Q. Dou, "Contrastive Cross-Site Learning With Redesigned Net for COVID-19 CT Classification," *IEEE Journal of Biomedical and Health Informatics,* vol. 24, p. 2806–2813, 10 2020.

[11] E. Soares, P. Angelov, S. Biaso, M. H. Froes and D. K. Abe, "SARS-CoV-2 CT-scan dataset: A large dataset of real patients CT scans for SARS-CoV-2 identification," *medRxiv,* 2020.

[12] P. Eduardo, "SARS-COV-2 Ct-Scan Dataset," [Online]. Available: https://www.kaggle.com/plameneduardo/sarscov2-ctscan-dataset.

[13] A. K. Tripathi, K. Sharma, M. Bala, A. Kumar, V. G. Menon and A. K. Bashir, "A Parallel Military-Dog-Based Algorithm for Clustering Big Data in Cognitive Industrial Internet of Things," *IEEE Transactions on Industrial Informatics,* vol. 17, pp. 2134-2142, 2021.

[14] V. Tanwar and K. Sharma, "Multi-Model Fake News Detection based on Concatenation of Visual Latent Features," in *2020 International Conference on Communication and Signal Processing (ICCSP)*, 2020.

[15] A. Tomar and P. Chanak, "A Game Theory based Fault Tolerance Routing Scheme for Wireless Sensor Networks," in *2020 IEEE International Students' Conference on Electrical,Electronics and Computer Science (SCEECS)*, 2020.

[16] "FastAI List of transforms for data augmentation in CV," FastAI, [Online]. Available: https://fastai1.fast.ai/vision.transform.html.

[17] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition,* 2009.

[18] J. Howard and S. Gugger, "Fastai: A Layered API for Deep Learning," *Information*, vol. 11, p. 108, 2 2020.

[19] "ImageNet data set," [Online]. Available: https://www.image-net.org.

[20] K. Weiss, T. M. Khoshgoftaar and D. Wang, "A survey of transfer learning," *Journal of Big Data,* vol. 3, p. 9, 2016.

[21] C. Goutte and E. Gaussier, "A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation," in *Advances in Information Retrieval*, Berlin, 2005.

[22] A. P. Bradley, "The Use of the Area under the ROC Curve in the Evaluation of Machine Learning Algorithms," *Pattern Recogn.*, vol. 30, p. 1145–1159, 7 1997.

[23] L. N. Smith, "Cyclical Learning Rates for Training Neural Networks," in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV),* 2017.

[24] A. Y. Ng, "Feature Selection, L1 vs. L2 Regularization, and Rotational Invariance," in *Proceedings of the Twenty-First International Conference on Machine Learning,* New York, NY, USA, 2004.

[25] H. Panwar, P. K. Gupta, M. K. Siddiqui, R. Morales-Menendez, P. Bhardwaj and V. Singh, "A deep learning and grad-CAM based color visualization approach for fast detection of COVID-19 cases using chest X-ray and CT-Scan images," *Chaos, Solitons & Fractals,* vol. 140, p. 110190, 2020.

[26] S. Yazdani, S. Minaee, R. Kafieh, N. Saeedizadeh and M. Sonka, *COVID CT-Net: Predicting Covid-19 From Chest CT Images Using Attentional Convolutional Network*, 2020.