

**DETECTION OF COVID-19 X-RAY AND CT IMAGES USING DEEP LEARNING AND IMAGE FUSION**

**A PROJECT REPORT**

***Submitted by***

|  |  |
| --- | --- |
| **KIRUTHIGA S** | **211717106060** |
| **NASIMA BANU A** | **211717106078** |
| **PRAVEENA R** | **211717106088** |

***in partial fulfilment for the award of the degree***

***of***

**BACHELOR OF ENGINEERING**

**in**

**ELECTRONICS AND COMMUNUCATION ENGINEERING**

**RAJALAKSHMI INSTITUE OF TECHNOLOGY, KUTHAMBAKKAM**

**ANNA UNIVERSITY: CHENNAI 600 025**

APRIL 2021

**ANNA UNIVERSITY CHENNAI: CHENNAI 600 025**

**BONA FIDE CERTIFICATE**

Certified that this project report “**DETECTION OF COVID-19 X-RAY AND CT IMAGES USING DEEP LEARNING AND IMAGE FUSION”** is a bona fide work of “**KIRUTHIGA S(211717106060) , NASIMA BANU A(211717106078) , PRAVEENA R(211717106088)**” who carried out the project under my guidance

|  |  |
| --- | --- |
| **SIGNATURE** | **SIGNATURE** |
| **Dr. R RAJESWARI M.E.,Ph.D.** | **Dr. V NAGARAJU M.E., Ph.D.** |
| **SUPERVISOR**  Professor | **HEAD OF THE DEPARTMENT**  Professor |
| Department of Electronics and  Communication Engineering | Department of Electronics and  Communication Engineering |
| Rajalakshmi Institute of Technology | Rajalakshmi Institute of Technology |
| Chennai-600124 | Chennai-600124 |

**CERTIFICATE**

|  |  |  |
| --- | --- | --- |
| **College Name** | **:** | **2117- Rajalakshmi Institute of**  **Technology** |
| **Branch & Semester** | **:** | **Electronics and Communication**  **Engineering & VIII Semester** |
| **Subject** | **:** | **EC8811 PROJECT WORK** |

**TITLE OF THE PROJECT:**

DETECTION OF COVID-19 X-RAY AND CT IMAGES USING DEEP LEARNING AND IMAGE FUSION

|  |  |
| --- | --- |
| **NAME OF THE STUDENTS:** |  |
| **KIRUTHIGA S** | **211717106060** |
| **NASIMA BANU A** | **211717106078** |
| **PRAVEENA R** | **211717106088** |

The report on the project work submitted by the above students in partial fulfilment for the award of the degree of Bachelor of Engineering in ELECTRONICS AND COMMUNICATION ENGINEERING of Anna University reported the work done by the above students and then evaluated.

**The University Viva- voce was held on \_\_\_\_\_\_\_\_\_\_\_\_\_.**

|  |  |
| --- | --- |
| **INTERNAL EXAMINAR** | **EXTERNAL EXAMINAR** |

**ACKNOWLEDGEMENT**

We wish to express our heartly and sincere gratitude to our chairman **Mr.S.MEGANATHAN**, B.E.., F.I.E., for his sincere endeavor in educating using his premier institution.

We wish to express our deep gratitude to our chairperson **Dr.(Mrs.) THANGAM MEGANATHAN**, M.A., M.Phil., Ph.D., for her enthusiastic motivation which helped us a lot in completing the project.

We wish to express our sincere gratitude to our vice chairman **Dr.M.HAREE SHANKAR**, MBBS., MD., for his constant motivation in educating us in his premier institution.

We express our thankfulness to **Dr.P.K.NAGARAJAN**, M.E, Ph.D., Principal**,**  **Dr.R.SUNDAR**, M.Tech, Ph.D., Director, and **Dr.M.VELAN**, M.Tech., Ph.D., Dean, Rajalakshmi Institute of Technology for their kind support and the facilities provided to complete our work on time.

With profound sense and regards, we acknowledge with great pleasure **Dr.V.NAGARAJU**, M.E., Ph.D., Professor & Head, Department of Electronics and Communication Engineering for their valuable suggestions and guidance for the development and completion of our project.

We express our sincere thanks to our supervisor **Dr.R.RAJESWARI**, M.E., Ph.D., Professor, Department of Electronics and Communication Engineering, for leading us on the project.

We extend our gratitude to **Dr.I.CHANDRA**, M.Tech., Ph.D., Professor, **Dr.M.Malathi**, M.E., Ph.D., Associate Professor, Project Coordinators for their timely organization of reviews and their support throughout the project work.

Finally, we express our deep sense of gratitude to our parents, all our faculty members, technical staff, and all our friends for their constant encouragement and moral support.

**ABSTRACT**

The 2019 novel coronavirus (COVID-19), with a beginning stage in China, has spread quickly among individuals living in different nations and is moving toward around 12,245,417 cases overall as per the insights of the European Center for Disease Prevention and Control. There are a predetermined number of COVID-19 test packs accessible in clinics because of the expanding cases every day. In this way, it is important to execute a programmed location framework as a snappy elective conclusion alternative to forestall COVID-19 spreading among individuals. In this examination, we take two modality images like CT or X-Rays or the combination of CT and X-Rays, and fuse (mixing of multiple pixels) them to produce a single image for the pre-processing stage. The output of the pre-processing stage is given as input to CNN & RNN. Now, both outputs are taken in F1 Precision recall (decision-making algorithm) and the result is produced with about 95-100% accuracy.

Keywords: Coronavirus (COVID – 19), CNN, Datasets, Deep Learning, Fusion.

**TABLE OF CONTENTS**

**CHAPTER.NO TITLE PAGE.NO**

**ABSTRACT v**

**LIST OF CONTENT vi**

**LIST OF TABLES viii**

**LIST OF FIGURES ix**

**LIST OF ABBREVIATIONS x**

1. **INTRODUCTION 1** 
   1. GENERAL
   2. SYSTEMS AND ITS DIFFICULTIES **2**
   3. OVERVIEW **2**
   4. OBJECTIVES **3**
   5. CONCLUSION **4**
2. **LITERATURE REVIEW 5**

2.1 INTRODUCTION **5**

2.2 RELATED WORKS **6**

2.3 CONCLUSION **9**

1. **SYSTEM OVERVIEW 11** 
   1. DATASET **11**
   2. PRE-PREPARING 12
   3. DEEP TRANSFER LEARNING 15
   4. TRIAL SETUP 18
   5. EXECUTION MATRICES 18
   6. CONCLUSION 19

1. **SYSTEM ANALYSIS 21** 
   1. COMPUTER VISION 21
   2. DEEP LEARNING 22
   3. OPENCV 22
   4. TENSORFLOW 23
   5. KERAS 24
   6. PYTORCH 25
2. **SYSTEM PLANNING 26**

5.1 INTRODUCTION 26

5.2 SYSTEM DESIGN 27

5.2.1 MOCILENETV2 27

5.2.2 IMAGE PREPROCESSING 28

USING PYTORCH

5.3 IMAGE CONTRAST ENHANCEMENT 28

5.4 MODULES 30

5.4.1 CNN MODULE 30

5.4.2 RNN MODULE 31

5.6 CONCLUSION 32

1. **IMPLEMENTATION AND RESULTS 33**

6.1 INTRODUCTION 33

6.2 IMPLENTATION 33

6.3 RESULT AND DISCUSSION 35

6.4 CONCLUSION 38

1. **CONCLUSION 41**
2. **REFERENCES 42**

**LIST OF TABLES**

**TABLE.NO TITLE PAGE.NO**

3.1 Two generally utilized discrete approximations 15

to the AMF

6.3.1 Expectation execution results got from various 39-40

pre-prepared CNN models for 5-overlap cross

approval techniques.

**LIST OF FIGURES**

**TABLE.NO TITLE PAGE.NO**

3.1.1 Agent chest X-beam pictures of typical. 12

3.1.2 Agent chest X-beam pictures of COVID-19 patients 13

3.2.1 Flowchart of AMF 15

3.3.1 Schematic portrayal of pre-prepared models for 18

the forecast of COVID-19 patients and typical.

3.4.1 Visual presentation of testing and preparing datasets for 19

5-crease cross-approval.

5.2.1 MobileNetV3 29

6.2.1 The presentation of three pre-prepared models 35

6.2.2 The presentation of three pre-prepared models 36

(misfortune esteems)

6.3.1 The disarray grid and ROC plots got utilizing 38

pre-prepared models

**LIST OF ABBREVIATIONS**

**ACC**  Accuracy

**AMF**  Adaptive Middle Filter

**API** Application Programming Units

**ARDS**  Acute respiratory distress syndrome

**BSD**  Berkely Software Distribution

**CNN** Convolutional Neural Network

**CPU** Central Processing Unit

**CT** Computed Tomography

**CUDA** Compute Unified Device Architecture

**DCNN**  Deep Convolutional Neural Network

**EFT**  Electronic fund transfer

**FAIR** Face Books Al Research

**FN**  False Negative

**FP**  False Positive

**GPU** Graphics Processing Unit

**IBM** Lean Body Mass

**MATLAB** Matrix Laboratory

**MERS-COV** Middle East respiratory syndrome

**MMX** Multimedia Extensions

**MRI**  Magnetic resonance imaging

**OPEN CV** Open Source Computer Vision Library

**OS** Operating System

**PCA**  Principal component analysis

**PCR** Polymerase chain reaction

**ResNet 50** Residual Neural Network 50

**RF**  Random Forest

**RNN** Recurrent Neural Network

**ROC** Registrars Of Companies

**SARS-COV-2** Severe acute respiratory syndrome coronavirus 2

**SSE** Senior Section Engineer

**STL** SteroLiTography / Standard Triangle Language

**SVM**  Support Vector Machine

**SYCL** Sickle

**TN**  True Negative

**TP**  True Positive

**TPU** Thermoplastic Polyurethane

**VB – NET** Visual Basic – Net

**CHAPTER 1**

**INTRODUCTION**

* 1. **GENERAL**

The 2019 novel coronavirus (COVID-19) pandemic showed up in Wuhan, China in December 2019 and has become a genuine general medical issue overall. The infection that caused COVID-19 pandemic illness was called extreme intense respiratory disorder coronavirus 2, additionally named SARS-CoV-2. Coronaviruses (CoV) are an enormous group of infections that cause maladies coming about because of colds, for example, the Middle East Respiratory Syndrome (MERS-CoV) and Severe Acute Respiratory Syndrome (SARS-CoV). Coronavirus ailment (COVID-19) is another species that was found in 2019 and has not been recently distinguished in people.

Coronaviruses are zoonotic because of pollution from creatures to people. There are contemplates that the SARS-CoV infection is sullied from musk felines to people, and the MERS-CoV infection is polluted from dromedary to people. COVID-19 infection is dared to be tainted from bats to people. Respiratory transmission of the ailment from one individual to an individual caused the quick spread of the pestilence.

While COVID-19 causes milder side effects in around 82 percent of cases, the others are extreme or basic. Coronavirus cases the all-out number is roughly 12,420,705 and 558,091 of them passed on and 7,246,372 were recouped. At present, the number of contaminated patients number is 4,616,242. While 99% of the number of contaminated patients endure the sickness somewhat, 1% of the rest has a genuine or basic condition.

**1.2 SYMPTOMS AND ITS DIFFICULTIES**

Indications of disease incorporate respiratory manifestations, fever, hack, and dyspnea. In more genuine cases, the disease can cause pneumonia, serious intense respiratory condition, septic stun, multi-organ disappointment, and passing. It has been resolved that men are more tainted than ladies and that there is no passing in kids between the ages of 0-9. Respiratory paces of cases with COVID-19 pneumonia have been demonstrated to be quicker contrasted with sound individuals.

Indeed, even in many created nations, the wellbeing framework has gotten to the meaningful part of breakdown because of the expanding interest for escalated care units all the while. Concentrated consideration units are loaded up with patients who deteriorate with COVID-19 pneumonia. The COVID-19 cases between the times of December 31st, 2019, and July tenth, 2020 is accounted for 12,245,417 affirmed cases everywhere throughout the world [10].

**1.3 OVERVIEW**

As indicated by the most recent rules distributed by the Chinese government, the conclusion of COVID-19 ought to be affirmed by quality sequencing for respiratory or blood tests as a key pointer for turn around record polymerase chain response (RT-PCR) or hospitalization. In the current general wellbeing crisis, the low affectability of RT-PCR implies that numerous COVID-19 patients won't be distinguished rapidly and may not get proper treatment. Likewise, given the exceptionally irresistible nature of the infection, they risk tainting a bigger populace.

Rather than the patients standing by to get positive infection tests, analyze now incorporate every individual who uncovers the conspicuous pneumonia example of chest examine COVID-19. Through this strategy, specialists will have the option to segregate and treat patients all the more rapidly. Regardless of whether demise doesn't happen in COVID-19, a few patients get by with lasting lung harm. As per the World Health Organization, COVID-19 likewise opens gaps in the lungs like SARS, giving them a "honeycomb-like appearance".

Registered Tomography (CT) sweep of the chest is one of the strategies used to analyze pneumonia. Man-made consciousness (AI) based robotized CT picture investigation apparatuses for the identification, evaluation, and observing of coronavirus and to recognize patients with coronavirus from sickness free have been created. In an investigation by Fei et al., they built up a profound learning-based framework for the programmed division of all lung and contamination locales utilizing chest CT. Xiaowei et al. meant to build up an early screening model to recognize COVID-19 pneumonia and Influenza-A viral pneumonia from sound cases utilizing aspiratory CT pictures and profound learning methods. In Shuai et al. study, given the COVID-19 radiographic changes from CT pictures, they have built up a profound learning strategy that can extricate the graphical highlights of COVID-19 to give clinical analysis preceding pathogenic testing and along these lines spare crucial time for the infection found.

**1.4 OBJECTIVE**

MERS-CoV and SARS-CoV are communicated as cousins of COVID-19. There are logical distributions utilizing chest X-beam pictures in the finding of MERS-CoV and SARS-CoV. The investigation of Ahmet Hamimi about MERS CoV indicated that there are highlights in the chest X-beam and CT that resemble the appearance of pneumonia. In the examination by Xuanyang et al., information mining procedures were utilized to recognize SARS and normal pneumonia dependent on X-beam pictures.

X-beam machines are utilized to filter the influenced body, for example, cracks, bone disengagements, lung contaminations, pneumonia, and tumors. CT filtering is a sort of cutting edge X-beam machine that looks at the delicate structure of the dynamic body part and more clear pictures of the inward delicate tissues and organs. Utilizing an X-beam is a quicker, simpler, less expensive, and less destructive technique than CT. Inability to expeditiously perceive and treat COVID-19 pneumonia may prompt an expansion in mortality.

**1.5 CONCLUSION**

In this investigation, we have proposed a programmed forecast of COVID-19 utilizing a profound convolution neural system based on pre-prepared exchange models and Chest X-beam pictures. For this reason, we have utilized ResNet50, InceptionV3, and Inception-ResNetV2 pre-prepared models to get a higher forecast precision for a little X-beam dataset. The oddity of this paper is summed up as follows: i) The proposed models have started to finish structure without manual component extraction and determination techniques. ii) We show that ResNet50 is a successful pre-prepared model among the other two pre-prepared models. iii) Chest X-beam pictures are the best apparatus for the recognition of COVID-19. iv) The pre-prepared models have been appeared to yield extremely high outcomes in the little dataset (50 COVID-19 versus 50 Normal).

**CHAPTER 2**

**LITERATURE REVIEW**

**2.1 INTRODUCTION**

Since the vaccine isn't yet evolved, the correct measure to decrease the scourge is to confine individuals who are decidedly influenced. Yet, the issue is making a speedy symptomatic to recognize positive patients from negative. In this situation, a few examinations were introduced permitting to recognize variations from the norm in Chest X-beam and CT pictures. In reality, Gozes et al. (2020) proposed a model permitted to separate coronavirus patients from sound patients. The proposed framework delivered a restriction guide of the lung anomaly just as estimations.

To be sure, it was part into two subsystems: Subsystem An: a 3D examination was utilized to identify knobs, and little opacities utilizing business off-the-rack programming from that point estimations and restriction were given. Subsystem B: the initial step is the lung Crop stage where the lung district of intrigue (ROI) was extricated utilizing a lung division module (U-net design). The subsequent advance is the identification of coronavirus variations from the norm utilizing profound convolutional neural system model ResNet50. The third step was the variation from the norm confinement step. On the off chance that another cut distinguished positive, the system enactment maps were removed utilizing the Grad-cam procedure. From that point, after the mix of the yield of subsystem An and subsystem B, the creators included a Corona score determined by a volumetric summation of the system initiation maps.

**2.2 RELATED WORKS**

A programmed and profound learning-based technique utilizing X-beam pictures to foresee Covid19 was proposed by Narin et al. (2020). The proposed technique

utilized three Deep Convolution Neural Network models. They have utilized a dataset containing 50 X-beam pictures of covid19 patients and 50 ordinary X-beam pictures and all the pictures were resized to 224 × 224. To beat the issue of the set number dataset, the creators utilized exchange learning models. The dataset was isolated into two sections: 80% for preparing and 20% for testing. The created DCNN depended on pre-prepared models (ResNet50, Inception\_V3, and Inceptio\_ResNet\_V2) permitted to recognize Covid19 from typical X-beam pictures. They utilized additionally an exchange learning procedure and the k-overlap strategy was utilized as a cross-approval technique with k = 5. The acquired outcomes demonstrated the pre-prepared model ResNet50 gave great (the estimation of precision is equivalent to 98%).

In Hemdan et al. (2020), a profound learning classifiers structure "COVIDX-Net" helping radiologists to consequently recognize Covid19 was proposed. The created structure permits grouping Covid19 X-beam pictures into positive and negative Covid19. Creators utilized seven DCNN structures (VGG19, DenseNet121, ResNetV2, InceptionV3, InceptionResNetV2, Xception, and MobileNetV2). They additionally utilized a dataset including 50 X-beam pictures split into two classes typical and Covid19 positive cases (25 X-beam pictures for each). The pictures were resized to 224 × 224 pixels. 80% of pictures were utilized for the preparation stage and 20% for testing. The acquired outcomes delineated that VGG19 and DenseNet201 designs have great exhibitions with an F1 score of 89% and 91% for typical and covid19.

To recognize Covid19 cases from other Pneumonia (Bacteria and infection) and ordinary cases, Farooq and Hafeez (2020) proposed a convolutional neural system (CNN) structure. They utilized the COVDIX dataset made by Wang and Wong (2020). The dataset contains 5941 chest radiography pictures gathered from 2839 patients. In this work, they utilized a bit of the COVIDX dataset, and it was separated into four sets: Covid19 (48 pictures), Bacterial (660), Viral (931), and Normal (1203 pictures). In the preparation step, which was acted in 3 stages, the Cyclical Learning Rate was utilized for assisting with choosing the ideal learning rate and that for each progression.

The acquired outcomes delineate that the proposed Covid-ResNes gave great recognizable proof exactness of 96.23% contrasted with Covid-Net 83.5%.

Bhandary et al. (2020) revealed a profound learning system to group lung irregularities like pneumonia utilizing chest X-beam pictures and malignancy utilizing lung CT pictures. The proposed model depended on a Modified AlexNet model (MAN). Consequently, they proposed two models: An) a MAN model joined with Support Vector Machine (SVM) used to recognize pneumonia pictures from typical pictures. For the outcomes, the proposed model indicated great outcomes (exactness 96.8%) contrasted with different models AlexNet, VGG16, VGG19, ResNet50, and MAN\_Softmax. B) For this assessment, the lung CT pictures were utilized. Creators combined MAN with Ensemble-Feature-Technique (EFT) to improve the exhibition of characterization. In the wake of separating highlights from pictures, the Principal Component Analysis (PCA) was executed. At long last, to group CT pictures into Malignant and Benign, the model was joined with SVM, k-Nearest Neighbors

(k-NN), and Random Forest (RF). The acquired outcomes delineated that MAN joined with SVM accomplished great exactness with and without EFT 97.27% and 86.47 separately.

In Zhang et al. (2020), the creators introduced a profound learning model permitted to distinguish Covid19 from sound individuals utilizing Chest X-beam pictures. The model depended on three segments: The first is the spine arrange which is made out of 18 layers remaining convolutional neural system. Its standard is to extricate the elevated level highlights from the chest X-beam picture. The subsequent one is the arrangement head planned to create an order score Pcs. It was fueled by the extricated highlights by the spine organization. The third part is the inconsistency discovery head permits producing a scalar oddity score Pano. In the wake of figuring the order score and scalar inconsistency score, the choice was made by an edge T. The acquired outcomes demonstrated that the affectability diminished as long as the estimation of limit T diminished (affectability of 96% for T = 0.15).

The work (Xu et al., 2020) detailed a strategy to recognize COVID-19 from Influenza-A viral pneumonia and sound pictures utilizing profound learning strategies. They utilized different CNN to characterize Computed Tomography (CT) pictures. The introduced procedure can be summed up as 4 stages: 1) the pictures were pre-handled to separate powerful aspiratory districts 2) a 3D CNN was utilized to section numerous up-and-comer picture solid shapes 3) a model of picture arrangement was utilized to recognize the pictures fix into Covid19, Influenza-An and ordinary 4) by utilizing the loud or Bayesian capacity a general investigation report for one CT test was determined. The VNET-IR-RPN model was utilized for the division while the ResNet-18 model and ResNet-18 with the area consideration component model were utilized for the order step. The trial results show that the ResNet-18 model with the area consideration system gave the general precision pace of 86.7%.

Another profound learning model, that permits portioning and evaluating contamination districts in CT outputs of COVID-19 patients, was accounted for by Shan et al. (2020). Creators utilized VB-Net Neural Network and a human-on the up

and up (HITL) approach in the request to assist radiologists with clarifying programmed comment of each case. At that point, they utilized assessment measurements to evaluate the adequacy of the model (volumes and level of disease in the entire lung). They partitioned the CT pictures into a lot of assortments. These CT pictures that were molded physically by the radiologists will take care of the division arrange for preparation. At that point, the division results were physically adjusted by radiologists and were considered as new information to take care of the model. This procedure was rehashed to iteratively fabricate the model.

El Asnaoui et al. (2020) introduced an examination of late DCNN models for a programmed twofold arrangement of pneumonia pictures dependent on fined tuned variants of VGG16 (Simonyan and Zisserman, 2014; Zhang et al., 2019), VGG19 (Simonyan and Zisserman, 2014; Zhang et al., 2019), DenseNet201 (Huang et al., 2017), Inception\_ResNet\_V2 (Szegedy et al., 2016), Inception\_V3 (Szegedy et al.,

2015), Resnet50 (He et al., 2016) and MobileNet\_V2 (Sandler et al., 2018). The proposed work has been tried utilizing chest X-beam and CT datasets.

**2.3 CONCLUSION**

The examination choice is intended for high affectability over the accuracy, to ensure that no pertinent investigations were left out. Right now, all works done in this field center around paired arrangement aside from not many investigations. For this reason, the fundamental objective of this work is going to introduce an examination of ongoing profound convolutional neural system structures for programmed multiclass grouping of X-beam and CT pictures between ordinary, microscopic organisms, and coronavirus to reply to the accompanying exploration questions (RQ): RQ1). Is there any DL method that particularly beats other DL strategies? RQ2). Would dl be able to use to early screen coronavirus from CT and X-beam pictures? RQ3). What is the analytic exactness that DL can be accomplished dependent on CT and X-beam pictures? RQ4). Would dl be able to aid the endeavors to precisely identify and follow the movement or goal of the coronavirus?

**CHAPTER 3**

**SYSTEM OVERVIEW**

**3.1. DATASET**

In this investigation, chest X-beam pictures of 50 COVID-19 patients have been acquired from the open-source GitHub store shared by Dr. Joseph Cohen. This archive is comprising chest X-beam/CT pictures of principal patients with intense respiratory pain disorder (ARDS), COVID-19, Middle East respiratory condition (MERS), pneumonia, serious intense respiratory condition (SARS). What's more, 50 ordinary chest X-beam pictures were chosen from the Kaggle archive called "Chest X-Ray Images (Pneumonia)".

Our trials have been founded on a made dataset with chest X-beam pictures of 50 typical and 50 COVID-19 patients (100 pictures altogether). All pictures in this dataset were resized to 224x224 pixel size. In Figure 1 and Figure 2, agent chest X-beam pictures of typical and COVID-19 patients are given, separately.

|  |  |
| --- | --- |
| C:\Users\narin1\Desktop\coronofold1\train\normal\normal25.jpeg | C:\Users\narin1\Desktop\coronofold1\train\normal\normal29.jpeg |



***Figure 3.1. Agent chest CT and X-beam pictures of typical.***

|  |  |
| --- | --- |
|  | ***C:\Users\narin1\Desktop\coronofold1\train\covid\covid50.jpeg*** |



***Figure 3.2. Agent chest CT and X-beam pictures of COVID-19 patients.***

**3.2. PRE-PREPARING**

The information picture is exposed to the pre-handling steps to make the picture reasonable for additional procedure. The pre-process is utilized to stack the information picture to the MATLAB condition and it will evacuate the clamor present

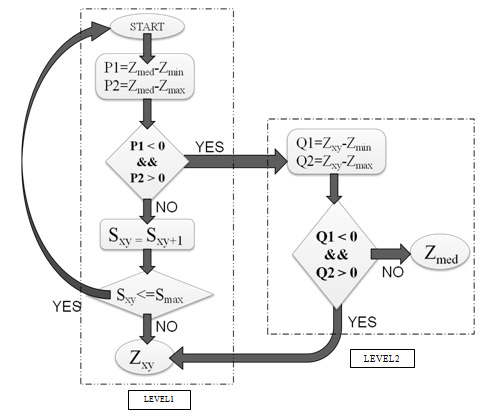
in the information picture. The picture is gone through the Adaptive Middle Filter (AMF) to bring down the commotion and to show signs of improvement picture. The AMF will likewise build the picture quality and the side of the pictures.

**AMF:** The standard middle channel performs well as long as the spatial commotion thickness of the salt and pepper clamor isn't huge. The channel execution debases when the spatial clamor fluctuation of the salt and pepper commotion builds (Chen and Whu, 1998). Further with the bigger picture and as the size of the piece builds, the subtleties and the edges become clouded (Maragos and Schafer, 1987). The standard middle channel doesn't consider the variety of picture attributes starting with one point then onto the next. The conduct of versatile channel changes dependent on the measurable attribute of the picture inside the channel district characterized by the rectangular window (Maragos and Schafer, 1987). The AMF varies from other versatile channels as the size of the rectangular window is made to change contingent upon:

The flowchart of versatile middle separating depends on two levels is appeared in Fig.2.2. The versatile middle separating calculation works in two levels, meant by level 1 and level 2. The utilization of AMF gives three significant purposes: To

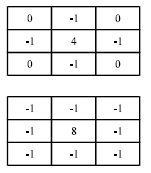
denoise pictures ruined by salt and pepper (drive) clamor; to give smoothing of non-hasty commotion and furthermore to diminish twisting brought about by extreme diminishing or thickening of item limits. The qualities and are considered factually by the calculation to be 'motivation like' clamor segments, regardless of

whether these are not the most reduced and most elevated conceivable pixel esteems in the picture.

******

***Fig.3.2.1: Flowchart of AMF***

The reason for LEVEL1 is to decide whether the middle channel yield is drive yield or not. On the off chance that level 1 finds a drive yield, at that point that would make it branch to level 2. Here, the calculation at that point builds the size of the window and rehashes level 1, and proceeds until it finds a middle worth that isn't motivation or the most extreme window size is reached, the calculation restores the estimation of .



***TABLE 3.1: Two generally utilized discrete approximations to the AMF***

Each time the calculation yields a worth, the window is moved to the following area in the picture. The calculation is then reinitialized and applied to the pixels in the new area. The middle worth can be refreshed iteratively utilizing just the new pixels, accordingly diminishing computational overhead.

Two regularly utilized little portions have appeared in Fig. 2.3. Since these portions are approximating a second subordinate estimation on the picture, they are delicate to clamor. To counter this, the picture is regularly smoothed before applying the Adaptive middle channel. This pre-preparing step diminishes the high recurrence commotion segments preceding the separation step.

**3.3. DEEP TRANSFER LEARNING**

Profound learning is a sub-part of the AI field, motivated by the structure of the cerebrum. Profound learning methods utilized as of late keep on demonstrating noteworthy execution in the field of clinical picture preparing, as in numerous fields. By applying profound learning methods to clinical information, it is attempted to draw important outcomes from clinical information.

Profound learning models have been utilized effectively in numerous zones, for example, grouping, division, and injury location of clinical information. Investigation of picture and sign information acquired with clinical imaging strategies, for example, Magnetic Resonance Imaging (MRI), Computed Tomography (CT), and X-beam with the assistance of profound learning models.

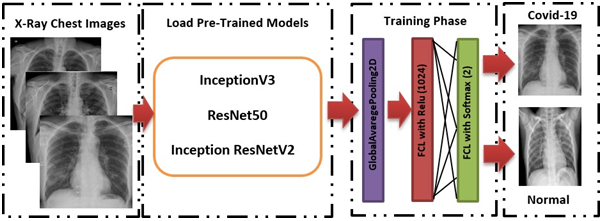
Because of these breaks down, discovery and analysis of sicknesses, for example, diabetes mellitus, mind tumor, skin malignant growth, and bosom disease are furnished with accommodation.

In the investigation of clinical information, probably the greatest trouble looked at by analysts is the set number of accessible datasets. Profound learning models frequently need a great deal of information. Naming this information by specialists is both exorbitant and tedious.

The greatest favorable position of utilizing the move learning strategy is that it permits the preparation of information with fewer datasets and requires fewer computation costs. With the exchange learning technique, which is broadly utilized in the field of profound learning, the data picked up by the pre-prepared model on an enormous dataset is moved to the model to be prepared.

In this investigation, we fabricated profound convolutional neural system (CNN) based ResNet50, InceptionV3, and Inception-ResNetV2 models for the characterization of COVID-19 Chest X-beam pictures to ordinary and COVID-19 classes. Furthermore, we applied the exchange learning strategy that was acknowledged by utilizing ImageNet information to beat inadequate information and preparing time.

The schematic portrayal of traditional CNN including pre-prepared ResNet50, InceptionV3, and Inception ResNetV2 models for the forecast of COVID-19 patients and ordinary were delineated in Figure 3.1. It is additionally accessible freely for open access at https://github.com/drcerenkaya/COVID-19-Detection.

******

***Figure 3.3.1 Schematic portrayal of pre-prepared models for the forecast of COVID-19 patients and typical.***

The lingering neural system (ResNet) model is an improved adaptation of the convolutional neural system (CNN). ResNet adds alternate routes between layers to take care of an issue. On account of this, it forestalls the twisting that happens as the system gets further and more perplexing. What's more, bottleneck squares are utilized to make preparing quicker in the ResNet model. ResNet50 is a 50-layer organization prepared on the ImageNet dataset. ImageNet is a picture database with more than 14 million pictures having a place with over 20 thousand classifications made for picture acknowledgment rivalries. InceptionV3 is a sort of convolutional neural system model.

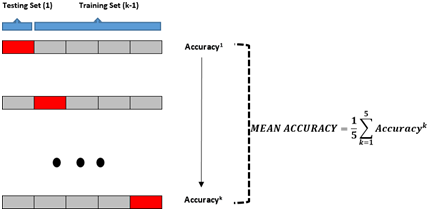
It comprises various convolution and most extreme pooling steps. In the last stage, it contains a completely associated neural system [30]. Similarly, as with the ResNet50 model, the system is prepared with the ImageNet dataset. The model comprises a profound convolutional organization utilizing the Inception of ResNetV2 engineering that was prepared on the ImageNet-2012 dataset. The contribution to the

model is a 299×299 picture, and the yield is a rundown of assessed class

probabilities.

**3.4. TRIAL SETUP**

The content programming language was utilized to prepare the proposed profound exchange, learning models. All trials were performed on a Google Colaboratory Linux worker with MATLAB. CNN models (ResNet50, InceptionV3, and Inception-ResNetV2) were pre-prepared with arbitrary introduction loads utilizing the Adam analyzer. The clump size, learning rate, and various ages were tentatively set to 2, 1e-5, and 30, individually for all trials. The dataset utilized was haphazardly part into two autonomous datasets with 80% and 20% for preparing and testing separately. As a cross-approval technique, k-crease was picked and results were gotten by 5 distinctive k esteems (k=1-5) as appeared in Figure 4.



***Figure 3.4.1. Visual presentation of testing and preparing datasets for 5-crease cross-approval.***

**3.5. EXECUTION METRICS**

Five standards were utilized for the exhibitions of profound exchange learning models. These are:

TP, FP, TN, and FN gave in Equation (1) – (5) speak to the number of True Positives,

Bogus Positive, True Negative, and False Negative, individually. Given a test dataset and model, TP is the extent of positive (COVID-19) that are effectively named as COVID-19 by the model; FP is the extent of negative (typical) that are mislabeled as positive (COVID-19); TN is the extent of negative (ordinary) that are accurately marked as should be expected and FN is the extent of positive (COVID-19) that are mislabeled as negative (typical) by the model.

**3.6 CONCLUSION**

Accuracy is a commonly used classification metric and indicates how well a classification algorithm can discriminate the classes in the test set. As shown in ( Eq 1 ), the accuracy can be defined as the proportion of the predicted correct labels (True Positive and True Negative ) to the total number (True Positive, True Negative, and False Positive and False Negative) of labels. In this study, accuracy refers to the overall accuracy of the model in distinguishing the two classes (covid, normal)...

Recall ( Eq 2) is often referred to as sensitivity (also called true positive rate) is the proportion to the Predicted True Positive and False Negative. Particular (Eq 3) is the proportion of predicted True Negative to the total number of predicted labels(True Negative and False Positive). Furthermore, F1 − score (Eq 5) refers to the harmonic mean of Precision and Recall while Specificity (also called true negative rate) measures the proportion of actual negatives that are correctly identified as such (Eq 4).

**CHAPTER 4**

**SYSTEM ANALYSIS**

**4.1 INTRODUCTION**

Computer vision is an interdisciplinary scientific field that deals with how computers can gain high-level understanding from digital images or videos. From the perspective of engineering, it seeks to understand and automate tasks that the human visual system can do, Computer vision tasks include methods for acquiring, processing, analyzing, and understanding digital images and extracting high dimensional data from the real world to produce numerical or symbolic information, e.g. in the forms of decisions, Understanding in this context means the transformation of visual images (the input of the retina)into descriptions of the world that make sense to thought processes and can elicit appropriate action.

This image understanding can be seen as the disentangling of symbolic information from image data using models constructed with the aid of geometry, physics, statistics, and learning theory. The scientific discipline of computer vision is concerned with the theory behind artificial systems that extract information from images. The image data can take many forms, such as video sequences, views from multiple cameras, multidimensional data from a 3D scanner or medical scanning device.

The technological discipline of computer vision seeks to apply its theories and models to the construction of computer vision systems. Computer vision is an interdisciplinary field that deals with how computers can be made to gain high-level understanding from digital images or videos. From the perspective of engineering, it seeks to automate tasks that the human visual system can do. It involves the development of a theoretical and algorithmic basis to achieve automatic visual

understanding. As a scientific discipline, computer vision is concerned with the theory behind artificial systems that extract information from images. The image data can take many forms, such as video sequences, views from multiple cameras, or multidimensional data from a medical scanner. As a technological discipline, computer vision seeks to apply its theories and models for the construction of computer vision systems.

**4.2 DEEP LEARNING**

**Deep learning** methods aim at learning feature hierarchies with features from higher levels of the hierarchy formed by the composition of lower-level features. Automatically learning features at multiple levels of abstraction allows a system to learn complex functions mapping the input to the output directly from data, without depending completely on human crafted features. Deep learning algorithms seek to exploit the unknown structure in the input distribution in order to discover good representations, often at multiple levels, with higher-level learned features defined in terms of lower-level features.

If we draw a graph showing how these concepts are built on top of each other, the graph is deep, with many layers. For this reason, we call this approach to AI deep learning. Deep learning excels on problem domains where the inputs (and even output) are analog. Meaning, they are not a few quantities in a tabular format but instead are images of pixel data, documents of text data, or files of audio data. Deep learning allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction.

**4.3 OPEN CV**

**OpenCV** (Open Source Computer Vision Library) is an open-source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code. The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high-resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc.

OpenCV has more than 47 thousand people in the user community and an estimated number of downloads exceeding 18 million. The library is used extensively in companies, research groups, and governmental bodies. Along with well-established companies like Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota that employ the library, there are many startups such as Applied Minds, Video Surf, and Zeitera, that make extensive use of OpenCV.

OpenCV’s deployed uses span the range from stitching street view images together, detecting intrusions in surveillance video in Israel, monitoring mine equipment in China, helping robots navigate and pick up objects atWillow Garage, detection of swimming pool drowning accidents in Europe, running interactive art in Spain and New York, checking runways for debris in Turkey, inspecting labels on products in factories around the world on to rapid face detection in Japan. It has C++, Python, Java, and MATLAB interfaces and supports Windows, Linux, Android, and Mac OS. OpenCV leans mostly towards real-time vision applications and takes advantage of MMX and SSE instructions when available. A full-featured CUDA and OpenCL interfaces are being actively developed right now. There are over 500 algorithms and about 10 times as many functions that compose or support those

algorithms. OpenCV is written natively in C++ and has a template interface that works seamlessly with STL containers.

**4.4** **TENSORFLOW**

Tensorflow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library and is also used for machine learning applications such as neural networks. It is used for both research and production at Google, TensorFlow is Google Brain's second-generation system. Version 1.0.0 was released on February 11, While the reference implementation runs on single devices, TensorFlow can run on multiple CPUs and GPUs (with optional CUDA and SYCL extensions for general-purpose computing on graphics processing units). Tensor Flow is available on 64-bit Linux, macOS, Windows, and mobile computing platforms including Android and iOS. Its flexible architecture allows for the easy deployment of computation across a variety of platforms (CPUs, GPUs, TPUs), and from desktops to clusters of servers to mobile and edge devices. The name Tensor Flow derives from the operations that such neural networks perform on multidimensional data arrays, which are referred to as tensors.

During the Google I/O Conference in June 2016, Jeff Dean stated that 1,500 repositories on GitHub mentioned TensorFlow, of which only 5 were from Google. Unlike other numerical libraries intended for use in Deep Learning like Theano, TensorFlow was designed for use both in research and development and in production systems, not least RankBrain in Google search and the fun DeepDream project. It can run on single CPU systems, GPUs as well as mobile devices, and large-scale distributed systems of hundreds of machines.

**4.5 KERAS**

Keras is an API designed for human beings, not machines. Keras follows best practices for reducing cognitive load: it offers consistent & simple APIs, minimizes the number of user actions required for common use cases, and it provides clear & actionable error messages. It also has extensive documentation and developer guides.

Keras contains numerous implementations of commonly used neural network building blocks such as layers, objectives, activation functions, optimizers, and a host of tools to make working with image and text data easier to simplify the coding necessary for writing deep neural network code. The code is hosted on GitHub, and community support forums include the GitHub issues page and a Slack channel. Keras is a minimalist Python library for deep learning that can run on top of Theano or Tensor Flow. It was developed to make implementing deep learning models as fast and easy as possible for research and development. It runs on python 2.7 or 3.5 and can seamlessly execute on GPUs and CPUs were given the underlying frameworks. It is released under the permissive MIT license.

Keras was developed and maintained by François Chollet, a Google engineer using four guiding principles: Modularity: A model can be understood as a sequence or a graph alone. All the concerns of a deep learning model are discrete components that can be combined in arbitrary ways. Minimalism: The library provides just enough to achieve an outcome, no-frills and maximizing readability. Extensibility: New components are intentionally easy to add and use within the framework, intended for researchers to trial and explore new ideas. Python: No separate model files with custom file formats. Everything is native Python. Keras is designed for minimalism and modularity allows you to very quickly define deep learning models and run them on top of a Theano or TensorFlow backend.

**4.6 PYTORCH**

Pytorchis an open-source machine learning library based on the Torch library, used for applications such as computer vision and natural language processing, primarily developed by Facebook’s AI Research lab (FAIR). It is free and open-source software released under the Modified BSD license. Although the Python interface is more polished and the primary focus of development, PyTorch also has a C++ interface. Tensor computing (like NumPy) with strong acceleration via graphics processing units (GPU). Deep neural networks built on a tape-based automatic differentiation system PyTorch defines a class called Tensor (torch. Tensor ) to store and operate on homogeneous multidimensional rectangular arrays of numbers. PyTorch Tensors are similar to NumPy Arrays, but can also be operated on a CUDA-capable Nvidia GPU. PyTorch supports various sub-types of Tensors.

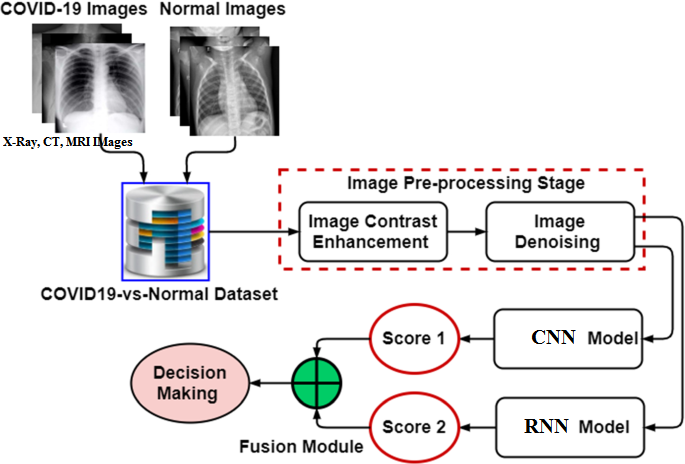
**CHAPTER 5**

**SYSTEM PLANNING**

**5.1 INTRODUCTION**

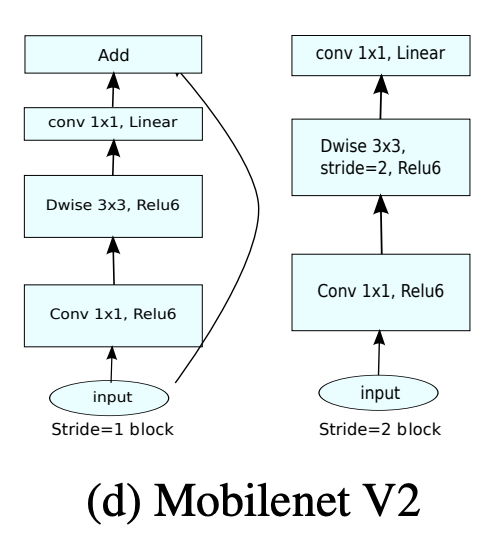
Data at Source The majority of the images were augmented by OpenCV. The set of images were already labeled “mask” and “no mask”. The images that were present were of different sizes and resolutions, probably extracted from different sources or machines (cameras) of different resolutions. Data preprocessing steps as mentioned below were applied to all the raw input images to convert them into clean versions, which could be fed to a neural network machine learning model.

**5.2 SYSTEM DESIGN**



**5.2.1 MOBILENETV2**

**It**is a convolutional neural network architecture that seeks to perform well on mobile devices. It is based on an inverted residual structure where the residual connections are between the bottleneck layers. The intermediate expansion layer uses lightweight depthwise convolutions to filter features as a source of non-linearity. As a whole, the architecture of MobileNetV2 contains the initial fully convolution layer with 32 filters, followed by 19 residual bottleneck layers.



***Fig 5.2.1 :MobileNetV2***

**5.2.2 IMAGE PREPROCESSING USING PYTORCH**

Images in a dataset do not usually have the same pixel intensity and dimensions. In this section, you will pre-process the dataset by standardizing the pixel

values. The next required process is transforming raw images into tensors so that the algorithm can process them.

**5.3 IMAGE CONTRAST ENHANCEMENT**

Contrast enhancement is a technique that uses the colors available on the monitor or output system to make image features stand out more clearly. To increase contrast, contrast manipulations include changing the range of values in an image.

The process of modifying digital images so that the effects are more appropriate for display or further image processing is known as image enhancement. You may, for example, delete noise, sharpen, or brighten a picture to make it easier to spot important details.

The contrast enhancement technique is used in image processing to bring out the detail that occurs within the grey level image's low dynamic range. It was necessary to perform operations such as contrast enhancement and noise reduction or removal to improve the quality of an image.

**5.4 IMAGE DENOISING**

One of the fundamental challenges in the field of image processing and computer vision is image denoising, where the underlying goal is to estimate the original image by suppressing noise from a noise-contaminated version of the image. Image noise may be caused by different intrinsic (i.e., sensor) and extrinsic (i.e., environment) conditions which are often not possible to avoid in practical situations.

Therefore, image denoising plays an important role in a wide range of applications such as image restoration, visual tracking, image registration, image segmentation, and image classification, where obtaining the original image content is crucial for

strong performance. While many algorithms have been proposed for the purpose of image denoising, the problem of image noise suppression remains an open challenge, especially in situations where the images are acquired under poor conditions where the noise level is very high.

The **effectiveness**of most **image processing** algorithms depends on a careful **parameter choice**. For instance, denoising methods commonly require a **denoising strength or** a **patch size** to be set. These parameters can be adjusted per image, but neglecting the local image characteristics leads to**sub-optimal results**.

Setting the filtering parameters adaptively has obvious benefits, the denoising strength can be higher in smooth areas where the risk of blurring out details is low, and in turn, it can be lower in highly textured areas where noise is less visible. Adaptiveness can also be easily achieved by mixing the output of different algorithms, each operating at best in a different part of an image.

**5.4 MODULES**

**5.4.1 CNN MODULE**

The convolutional neural network (CNN), also known as CNN, is a form of deep learning neural network. CNN's are a major step forward in image recognition. They're most widely used to analyze visual imagery and are often involved in image classification behind the scenes. In a nutshell, consider CNN to be a machine learning algorithm that can take an input image, assign importance (learnable weights and biases) to various aspects/objects in the image, and distinguish between them.

CNN works by extracting features from the images. Any CNN consists of the following:

1. The input layer which is a grayscale image
2. The Output layer which is a binary or multi-class labels
3. Hidden layers consisting of convolution layers, ReLU (rectified linear unit) layers, the pooling layers, and a fully connected Neural Network.

**5.4.2 RNN MODULE**

Convolutional Neural Network(CNN) has been widely used for image recognition with great success. However, there are several limitations of the current CNN-based image recognition paradigm. First, the receptive field of CNN is generally fixed, which limits its recognition capacity when the input image is very large. Second, it lacks the computational scalability for dealing with images of different sizes.

Third, it is quite different from a human visual system for image recognition, which involves both feedforward and recurrent preprocessing. This paper proposes a different paradigm of image recognition, which can take advantage of variable scales of the input images, has more computational scalabilities, and is more similar to image recognition by the human visual system. It is based on a recurrent neural network (RNN) defined on an image scale with an embedded base CNN, which is named Scale Recurrent Neural Network(SRNN). This RNN based approach makes it easier to deal with images with variable sizes and allows us to borrow existing RNN techniques, such as LSTM and GRU, to further enhance the recognition accuracy.

Our experiments show that the recognition accuracy of a base CNN can be significantly boosted using the proposed SRNN models. It also significantly outperforms the scale ensemble method, which integrates the results of performing

CNN to the input image at different scales, although the computational overhead of using SRNN is negligible.

**5.5 CONCLUSION**

The image fusion process is characterized as gathering all relevant data from multiple images and merging it into a smaller number of images, typically only one. Multisensor image fusion is a computer vision technique that combines relevant information from two or more images into a single image.

Image fusion encloses all data analysis strategies aiming at combining the information of several images obtained with the same platform or by different spectroscopic platforms. Image fusion can simply build a single multiset or multiway structure with all images involved or connecting the related individual images through regression models. In any case, data analysis performed on structures of fused images always overcomes the results coming from individual image analysis.

**CHAPTER 6**

**IMPLEMENTATION AND RESULTS**

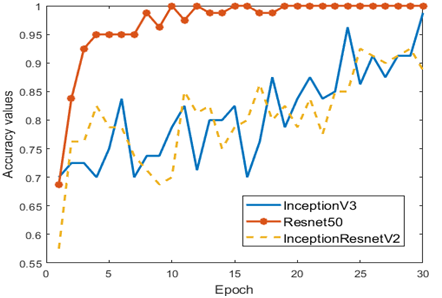
**6.1 INTRODUCTION**

In this examination, chest X-beam pictures have been utilized for the forecast of coronavirus malady patients (COVID-19). Well-known pre-prepared models, for example, ResNet50, InceptionV3, and Inception ResNetV2 have been prepared and tried on chest X-beam pictures. Preparing exactness and misfortune esteems for overlap 3 of the pre-prepared models are given in Figure 5 and Figure 6, individually.

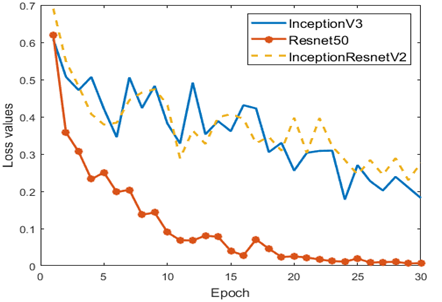
The preparation stage has been done up to the 30th age to abstain from overfitting for all pre-prepared models. It very well may be seen from Figure 5 that the most noteworthy preparing precision is gotten with the ResNet50 model. InceptionV3 and Inception-ResNetV2 models have comparative execution. Nonetheless, it is seen that ResNet50 shows a quick preparing process than different models. Although the pre-prepared models give high introductory qualities, the underlying qualities are beneath 70% because of the low number of information.

**6.2 IMPLEMENTATION**

The preparation misfortune estimations of ResNet50, InceptionV3, and Inception ResNetV2 have appeared in Figure 6. At the point when the misfortune figure is investigated, it is seen that the misfortune esteems decline in three pre-prepared models during the preparation stage. It tends to be said that the ResNet 50 model the two reductions misfortune esteems quicker and approaches zero.



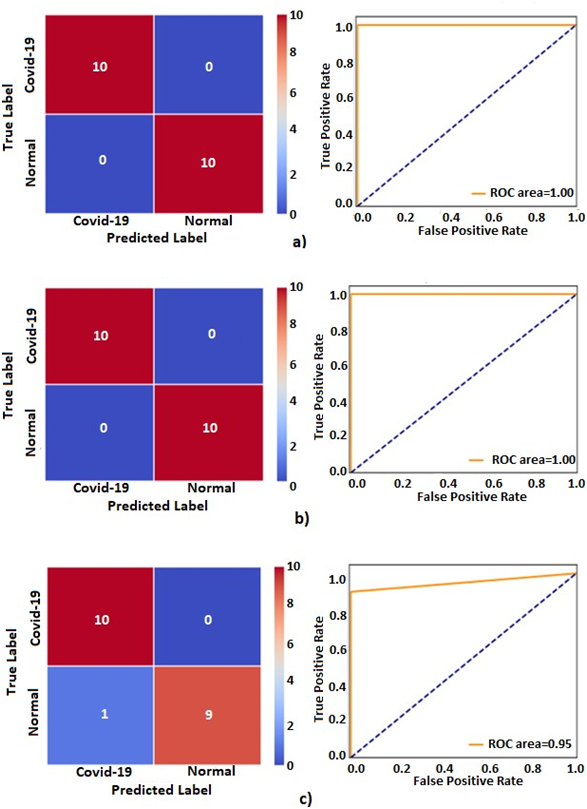
***Figure 6.2.1. The presentation of three pre-prepared models (Training precision for overlay 3)***

******

***Figure 6.2.2: The presentation of three pre-prepared models (Training misfortune esteems for overlay 3)***

In Figure 7, disarray networks of COVID-19 and ordinary test aftereffects of the models are given. Right off the bat, InceptionV3 pre-prepared model characterized 10 of the COVID-19 as True Positive for crease 3 and ordered 10 of the typical as True Negative. Furthermore, the ResNet50 model likewise characterized 10 of the COVID-19 as True Positive for overlap 3 and grouped 10 of the ordinary as True Negative. Ultimately, Inception ResNetV2 arranged 10 of the COVID-19 as True Positive for overlap 3 and grouped 9 of the ordinary as True Negative. Other than the disarray lattice, beneficiary working trademark bend (ROC) plots and territories for each model are given. InceptionV3 and ResNet50 pre-prepared models give off an impression of being high.

**6.3 RESULT AND DISCUSSION**

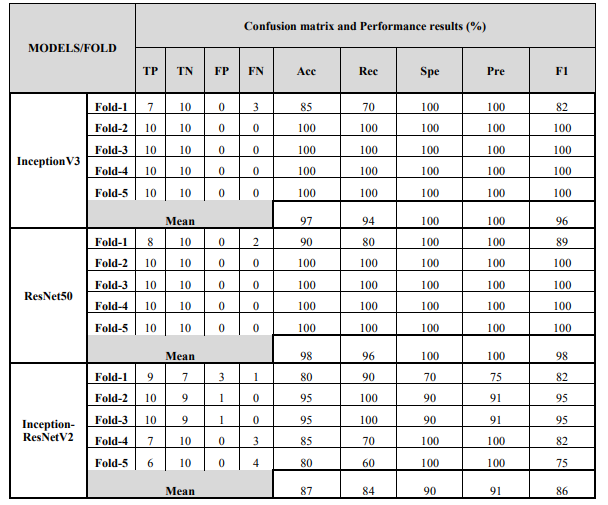


***Figure 6.3.1: The disarray grid and ROC plots got utilizing pre-prepared models for overlay 3 outcomes: an) InceptionV3, b) ResNet50, c) Inception-ResNetV2.***

In another nitty-gritty execution, examinations of three models utilizing the test information appear in Table 1. We have gotten the best execution as the precision of 98%, review of 96%, and particularity estimation of 100% for ResNet50 pre-prepared model. The most minimal exhibition esteems have been yielded an exactness of 87%, review of 84%, and explicitness estimation of 90% for Inception-ResNetV2. Thus, the ResNet50 model gives prevalence over the other two models both the preparing and testing stage.

There are not many examinations on writing because of the rise of COVID-19 infection ailment. A portion of these is as per the following: Prabira et al. [32] proposed identification of COVID-19 utilizing X-beam pictures dependent on profound component and SVM. They gathered X-beam pictures from GitHub, Kaggle, and the Open-I store. They extricated the profound component of CNN models and took care of to SVM classifier exclusively. They have gotten 95.38% of precision for ResNet50&SVM. Fei et al. [14], attempted to foresee COVID-19 patients utilizing the "VB-Net" neural system to portion COVID-19 contamination areas in CT filters. They dealt with the outcomes measurably.

***Table 6.3.1. Expectation execution results got from various pre-prepared CNN models for 5-overlap cross approval techniques. The shortened forms in Table 1 are True Positive (TP), True Negative (TN), False Positive (FP), False Negative (FN), Accuracy (ACC), Recall (REC), Specificity (SPE), Precision (PRE), F1-Score (F1).***



**6.4 CONCLUSION**

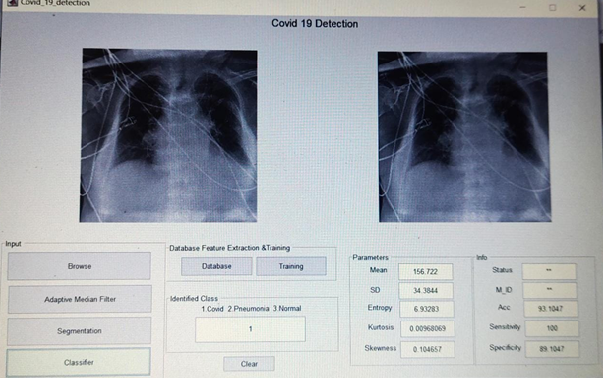
They acquired dice likeness coefficients of 91.6%±10.0%. Xiaowei et al. [15], proposed an early expectation model to order COVID-19 pneumonia from Influenza-

Viral pneumonia and solid cases utilizing aspiratory CT pictures utilizing profound learning methods. Their CNN model has yielded the most elevated by and large exactness was 86.7 % CT pictures. Shuai et al. [16], utilized CT pictures to anticipate COVID-19 cases. They likewise utilized the Inception move learning model to build up the calculation. They got a precision of 89.5% with the explicitness of 88.0% and affectability of 87.0%.

Notwithstanding these investigations in the writing, the principle focal points of our examination can be summed up as follows:

1. Chest X-beam pictures have been utilized in the investigation. X-beam pictures can be acquired from any emergency clinic effectively, rapidly, and without trouble.
2. Our strategy is a start-to-finish framework. In this way, it doesn't have any component extraction or choice.
3. Three distinctive pre-prepared normal models are looked at, for example, ResNet50, InceptionV3, and Inception-ResNetV2.
4. Although it is another subject and the quantity of information is constrained, the outcomes are very high.

The primary issue of our examination is the predetermined number of COVID-19 X-beam pictures utilized for the preparation of profound learning models. So as to overpower this issue, we have utilized profound exchange, learning models. On the off chance that we arrive at more information in the coming days, we are intending to improve working with various models.

******

***Figure 8: Output of the proposed system***

**CHAPTER 7**

**CONCLUSION**

**7.1 CONCLUSION**

An early forecast of COVID-19 patients is fundamental to forestall the spread of the sickness to others. In this investigation, we proposed a profound exchange learning-based methodology utilizing chest X-beam pictures acquired from COVID-19 patients and ordinary to anticipate COVID-19 patients naturally. Execution results show that the ResNet50 pre-prepared model yielded the most elevated exactness of 98% among the three models. In the light of our discoveries, it is accepted that it will help specialists to settle on choices in clinical practice because of the elite. So as to distinguish COVID-19 at a beginning phase, this examination gives knowledge on how profound exchange learning strategies can be utilized. In ensuing investigations, the characterization execution of various CNN models can be tried by expanding the number of pictures in the dataset.

**CHAPTER 8**

**REFERENCES**

1. Roosa, K., Lee, Y., Luo, R., Kirpich, A., Rothenberg, R., Hyman, J. M., Yan, P., and Chowell, G. Real-time forecasts of the COVID-19 epidemic in China from February 5th to February 24th, 2020. Infectious Disease Modelling, 5: 256-263, 2020.
2. Yan, L., Zhang, H. T., Xiao, Y., Wang, M., Guo, Y., Sun, C., Tang, X., Jing, L., Li, S., Zhang, M., Xiao, Y., Cao, H., Chen, Y., Ren, T., Jin, J., Wang, F., Xiao, Y., Huang, S., Tan, X., Huang, N., Jiao, B., Zhang, Y., Luo, A., Cao, Z., Xu, H., and Yuan, Y. Prediction of criticality in patients with severe Covid-19 infection using three clinical features: a machine learning-based prognostic model with clinical data in Wuhan. medRxiv 2020.02.27.20028027, 2020.
3. Stoecklin, S. B., Rolland, P., Silue, Y., Mailles, A., Campese, C., Simondon, A., Mechain, M., Meurice, L., Nguyen, M., Bassi C., Yamani, E., Behillil, S., Ismael,S., Nguyen, D., Malvy, D., Lescure, F. X., Georges, S., Lazarus, C., Tabaï, A., Stempfelet, M., Enouf, V., Coignard, B., Levy-Bruhl, D. and Team, I. First cases of coronavirus disease 2019 (COVID-19) in France: surveillance, investigations and control measures, January 2020. Eurosurveillance, 25(6):2000094, 2020.
4. Coronavirus. World Health Organization: https://www.who.int/health-topics/coronavirus, 2020.
5. Huang, C., Wang, Y., Li, X., Ren, L., Zhao, J., Hu, Y., Zhang, L., Fan, G., Xu, J., Gu, X., Cheng, Z., Yu, T., Xia, J., Wei, Y., Wu, W., Xie, X., Yin, W., Li, H., Liu, M., Xiao, Y., Gao, H., Guo, L., Xie, J., Wang, G., Jiang, R., Gao, Z., Jin Q., Wang, J., and Cao, B. Clinical features of patients infected with 2019 novel coronavirus in Wuhan, China. The Lancet, 395(10223): 497-506, 2020.
6. McKeever A. Here's what coronavirus does to the body. National Geographic: https://www.nationalgeographic.com/science/2020/02/here-is-what-coronavirus-does-to-the- body/, 2020.
7. Covid-19 coronavirus pandemic. worldometer:https://www.worldometers.info/ coronavirus/, 2020.
8. Mahase, E. Coronavirus: covid-19 has killed more people than SARS and MERS combined, despite lower case fatality rate. The BMJ, 368 : m641, doi: 10.1136/bmj.m641, 2020.
9. Wang, Y., Hu, M., Li, Q., Zhang, X. P., Zhai, G., and Yao, N. Abnormal respiratory patterns classifier may contribute to large-scale screening of people infected with COVID-19 in an accurate and unobtrusive manner. arXiv:2002.05534, 2020.
10. Covid-19 situation update worldwide, as of 10 July 2020. European Centre for Disease Prevention and Control: https://www.ecdc.europa.eu/en/geographical-distribution-2019-ncov- cases, 2020.
11. Ai, T., Yang, Z., Hou, H., Zhan, C., Chen, C., Lv, W., Tao, X., Sun, Z., and Xia, L. Correlation of Chest CT and RT-PCR Testing in Coronavirus Disease 2019 (COVID-19) in China: A Report of 1014 Cases. Radiology, 200642: 1-23, https://doi.org/10.1148/radiol.2020200642, 2020.
12. Zhang, W. Imaging changes of severe COVID-19 pneumonia in advanced stage. Intensive care medicine, 1-3, https://doi.org/10.1007/s00134-020-05990-y, 2020.
13. Gozes, O., Frid-Adar, M., Greenspan, H., Browning, P. D., Zhang, H., Ji, W., Bernheim, A., and Siegel, E. Rapid AI Development Cycle for the Coronavirus (COVID-19) Pandemic: Initial Results for Automated Detection & Patient Monitoring using Deep Learning CT Image Analysis. arXiv:2003.05037, 2020.
14. Shan, F., Gao, Y., Wang, J., Shi, W., Shi, N., Han, M., Xue, Z., and Shi, Y. Lung Infection Quantification of COVID-19 in CT Images with Deep Learning. arXiv:2003.04655, 2020.
15. Xu, X., Jiang, X., Ma, C., Du, P., Li, X., Lv, S., Yu, L., Chen, Y., Su, J., Lang, G., Li, Y., Zhao, H., Xu, K., Ruan, L., and Wu, W. Deep Learning System to Screen Coronavirus Disease 2019 Pneumonia. arXiv:2002.09334, 2020.
16. Wang, S., Kang, B., Ma, J., Zeng, X., Xiao, M., Guo, J., Cai, M., Yang, J., Li, Y., Meng, X., and Xu, B. A deep learning algorithm using CT images to screen for Corona Virus Disease (COVID-19). medRxiv 2020.02.14.20023028, 2020.
17. Hamimi, A. MERS-CoV: Middle East respiratory syndrome corona virus: Can radiology be of help? Initial single center experience. The Egyptian Journal of Radiology and Nuclear Medicine, 47(1): 95-106, 2016.
18. Xie, X., Li, X., Wan, S., and Gong, Y. Mining X-ray images of SARS patients. Data Mining: Theory, Methodology, Techniques, and Applications, Williams, Graham J., Simoff, Simeon J. (Eds.), pp. 282-294, ISBN: 3540325476 , Springer-Verlag, Berlin, Heidelberg, 2006.
19. Rachna C. Difference between X-ray and CT Scan. BioDifferences: https://biodifferences.com/difference-between-x-ray-and-ct-scan.html, 2017.
20. Cohen JP, Morrison P, and Dao L. Covid-19 image data collection. arXiv 2003.11597, 2020.
21. Mooney P. Chest x-ray images (pneumonia). Kaggle Repository: https://www.kaggle. com/paultimothymooney/chest-xray-pneumonia, 2018.
22. Yildirim, O., Talo, M., Ay, B., Baloglu, U. B., Aydin, G., and Acharya, U. R. Automated detection of diabetic subject using pre-trained 2D-CNN models with frequency spectrum images extracted from heart rate signals. Computers in Biology and Medicine, 113: 103387, 2019.
23. Saba, T., Mohamed, A. S., El-Affendi, M., Amin, J., and Sharif, M. Brain tumor detection using fusion of hand crafted and deep learning features. Cognitive Systems Research, 59: 221- 230, 2020.
24. Dorj, U. O., Lee, K. K., Choi, J. Y., and Lee, M. The skin cancer classification using deep convolutional neural network. Multimedia Tools and Applications, 77(8): 9909-9924, 2018.
25. Kassani, S. H., and Kassani, P. H. A comparative study of deep learning architectures on melanoma detection. Tissue and Cell, 58: 76-83, 2019.
26. Ribli, D., Horváth, A., Unger, Z., Pollner, P., and Csabai, I. Detecting and classifying lesions in mammograms with Deep Learning. Scientific Reports, 8(1): 1-7, 2018.
27. Celik, Y., Talo, M., Yildirim, O., Karabatak, M., and Acharya, U. R. Automated Invasive Ductal Carcinoma Detection Based Using Deep Transfer Learning with Whole-Slide Images. Pattern Recognition Letters, 133: 232-239, 2020.
28. Wu, Z., Shen, C., and Van Den Hengel, A. Wider or Deeper: Revisiting the ResNet Model for Visual Recognition. Pattern Recognition, 90: 119-133, 2019.
29. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C. and Fei-Fei, L. ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision, 115: 211-252, 2015.
30. Ahn, J. M., Kim, S., Ahn, K. S., Cho, S. H., Lee, K. B., and Kim, U. S. A deep learning model for the detection of both advanced and early glaucoma using fundus photography. PLoS ONE 13(11): e0207982, 2018.
31. Byra, M., Styczynski, G., Szmigielski, C., Kalinowski, P., Michalowski, L., Paluszkiewicz, R., Ziarkiewicz-Wróblewska, B., Zieniewicz, K., Sobieraj, P., and Nowicki, A. Transfer learning with deep convolutional neural network for liver steatosis assessment in ultrasound images. International

Journal of Computer Assisted Radiology and Surgery, 13(12): 1895-1903, 2018.

1. Sethy PK, Behera SK, Ratha PK, and Biswas P. Detection of coronavirus Disease (COVID-19) based on Deep Features and Support Vector Machine. International Journal of Mathematical, Engineering and Management Sciences, 5(4): 643-651, 2020.
2. R. Siegel, D. Naishadham, and A. Jemal, “Cancer statistics,” CA Cancer J. Clin., vol. 63, pp. 11–30, 2013.
3. Hao Han, Lihong Li, Senior Member, IEEE, Fangfang Han, Bowen Song, William Moore, and Zhengrong Liang,Fellow, IEEE journal of biomedical and health informatics, vol. 19, no. 2, march 2015
4. K. Awai, K. Murao, A. Ozawa, M. Komi, H. Hayakawa, S. Hori, and Y. Nishimura, “Pulmonary nodules at chestCT: Effect of computer-aided diagnosis on radiologists’ detection performance,” Radiology, vol. 230, no. 2, pp.347-d352, 2004.
5. Y.Lee,T.Hara,H.Fujita,S.Itoh,andT.Ishigaki,“Automateddetectionof pulmonary nodules in helical CT images basedon an improved templatematchingtechnique,”IEEETrans.Med.Imag.,vol.20,no.7,pp.595–604, Jul. 2001.
6. J. Dehmeshki, X. Ye, X. Lin, M. Valdivieso, and H. Amin, “Automated detection of lung nodules in CT imagesusing shape-based genetic algorithm,” Comput. Med. Imag. Graph., vol. 31, pp. 408–417, 2007.
7. A. Riccardi, T. S. Petkov, G. Ferri, M. Masotti, and R. Campanini, “Computer-aided detection of lung nodules via3-D fast radial transform, scale space representation, and Zernike MIP classification,” Med. Phys., vol. 38, no. 4,pp. 1962–1971, 2011.
8. M. Tan, R. Deklerck, B. Jansen, M. Bister, and J. Cornelis, “A novel computer-aided lung nodule detection systemfor CT images,”Med.Phys., vol. 38, no. 10, pp. 5630–5645, 2011.

**APPENDIX**

function varargout = Covid\_19\_detection(varargin)

gui\_Singleton = 1;

gui\_State = struct('gui\_Name', mfilename, ...

'gui\_Singleton', gui\_Singleton, ...

'gui\_OpeningFcn', @Covid\_19\_detection\_OpeningFcn, ...

'gui\_OutputFcn', @Covid\_19\_detection\_OutputFcn, ...

'gui\_LayoutFcn', [] , ...

'gui\_Callback', []);

if nargin && ischar(varargin{1})

gui\_State.gui\_Callback = str2func(varargin{1});

end

if nargout

[varargout{1:nargout}] = gui\_mainfcn(gui\_State, varargin{:});

else

gui\_mainfcn(gui\_State, varargin{:});

end

% End initialization code - DO NOT EDIT

% --- Executes just before Covid\_19\_detection is made visible.

function Covid\_19\_detection\_OpeningFcn(hObject, eventdata, handles, varargin)

% This function has no output args, see OutputFcn.

% hObject handle to figure

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% varargin command line arguments to Covid\_19\_detection (see VARARGIN)

% Choose default command line output for Covid\_19\_detection

handles.output = hObject;

set(handles.edit2,'String','\*\*');

set(handles.edit3,'String','\*\*');

set(handles.edit4,'String','\*\*');

set(handles.edit5,'String','\*\*');

set(handles.edit6,'String','\*\*');

set(handles.edit7,'String','\*\*');

set(handles.edit8,'String','\*\*');

set(handles.edit9,'String','\*\*');

set(handles.edit10,'String','\*\*');

set(handles.edit11,'String','\*\*');

set(handles.edit1,'String','\*\*');

% Update handles structure

guidata(hObject, handles);

% UIWAIT makes Covid\_19\_detection wait for user response (see UIRESUME)

% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.

function varargout = Covid\_19\_detection\_OutputFcn(hObject, eventdata, handles)

% varargout cell array for returning output args (see VARARGOUT);

% hObject handle to figure

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure

varargout{1} = handles.output;

% --- Executes on button press in Browse.

function Browse\_Callback(hObject, eventdata, handles)

% hObject handle to Browse (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

global a;global skw;

[filename, pathname] = uigetfile('\*.jpg', 'Pick a MRI Image');

if isequal(filename,0) || isequal(pathname,0)

warndlg('User pressed cancel')

else

filename=strcat(pathname,filename);

InputImage=imread(filename);

axes(handles.axes1);

imshow(InputImage);

[filename1, pathname1] = uigetfile('\*.jpg', 'Pick a CT Image');

handles.InputImage=InputImage;

a=InputImage;

me=mean2(a);%mean

sd=std2(a);%std dev

en=entropy(a);%enttropy

skw=skewness(a(:));%skewness

k=kurtosis(a(:));

set(handles.edit2,'String',me);

set(handles.edit3,'String',sd);

set(handles.edit4,'String',en);

set(handles.edit5,'String',k);

set(handles.edit6,'String',skw);

sen=Sensitivity\_image(a);

spe=specificity\_image(a);

set(handles.edit10,'String',sen);

set(handles.edit11,'String',spe);

end

% Update handles structure

guidata(hObject, handles);

% --- Executes on button press in AdaptiveMedianFilter.

function AdaptiveMedianFilter\_Callback(hObject, eventdata, handles)

% hObject handle to AdaptiveMedianFilter (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

InputImage=handles.InputImage;

GrayScaleImage=(InputImage);

NoisyImage=GrayScaleImage;

NoisyImage=double(GrayScaleImage);

[R C P]=size(NoisyImage);

OutImage=zeros(R,C);

Zmin=[];

Zmax=[];

Zmed=[];

for i=1:R

for j=1:C

if (i==1 & j==1)

% for right top corner[8,7,6]

elseif (i==1 & j==C)

% for bottom left corner[2,3,4]

elseif (i==R & j==1)

% for bottom right corner[8,1,2]

elseif (i==R & j==C)

%for top edge[8,7,6,5,4]

elseif (i==1)

% for right edge[2,1,8,7,6]

elseif (i==R)

% // for bottom edge[8,1,2,3,4]

elseif (j==C)

%// for left edge[2,3,4,5,6]

elseif (j==1)

else

SR1 = NoisyImage((i-1),(j-1));

SR2 = NoisyImage((i-1),(j));

SR3 = NoisyImage((i-1),(j+1));

SR4 = NoisyImage((i),(j-1));

SR5 = NoisyImage(i,j);

SR6 = NoisyImage((i),(j+1));

SR7 = NoisyImage((i+1),(j-1));

SR8 = NoisyImage((i+1),(j));

SR9 = NoisyImage((i+1)),((j+1));

TempPixel=[SR1,SR2,SR3,SR4,SR5,SR6,SR7,SR8,SR9];

Zxy=NoisyImage(i,j);

Zmin=min(TempPixel);

Zmax=max(TempPixel);

Zmed=median(TempPixel);

A1 = Zmed - Zmin;

A2 = Zmed - Zmax;

if A1 > 0 && A2 < 0

% go to level B

B1 = Zxy - Zmin;

B2 = Zxy - Zmax;

if B1 > 0 && B2 < 0

PreProcessedImage(i,j)= Zxy;

else

PreProcessedImage(i,j)= Zmed;

end

else

if ((R > 4 && R < R-5) && (C > 4 && C < C-5))

S1 = NoisyImage((i-1),(j-1));

S2 = NoisyImage((i-2),(j-2));

S3 = NoisyImage((i-1),(j));

S4 = NoisyImage((i-2),(j));

S5 = NoisyImage((i-1),(j+1));

S6 = NoisyImage((i-2),(j+2));

S7 = NoisyImage((i),(j-1));

S8 = NoisyImage((i),(j-2));

S9 = NoisyImage(i,j);

S10 = NoisyImage((i),(j+1));

S11 = NoisyImage((i),(j+2));

S12 = NoisyImage((i+1),(j-1));

S13 = NoisyImage((i+2),(j-2));

S14 = NoisyImage((i+1),(j));

S15 = NoisyImage((i+2),(j));

S16 = NoisyImage((i+1)),((j+1));

S17 = NoisyImage((i+2)),((j+2));

TempPixel2=[S1,S2,S3,S4,S5,S6,S7,S8,S9,S10,S11,S12,S13,S14,S15,S16,S17];

Zmed2=median(TempPixel2);

PreProcessedImage(i,j)= Zmed2;

else

PreProcessedImage(i,j)= Zmed;

end

end

end

end

end

PreProcessedImage3=[]

PreProcessedImage3(:,:,1)=PreProcessedImage;

PreProcessedImage3(:,:,2)=PreProcessedImage;

PreProcessedImage3(:,:,3)=PreProcessedImage;

PreProcessedImage=PreProcessedImage3;

PreProcessedImage=uint8(PreProcessedImage);

axes(handles.axes2);

imshow(PreProcessedImage,[]);

handles.PreProcessedImage=PreProcessedImage;

% Update handles structure

guidata(hObject, handles);

warndlg('Process completed');

% --- Executes on button press in GMMSegmentation.

function GMMSegmentation\_Callback(hObject, eventdata, handles)

% hObject handle to GMMSegmentation (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

PreProcessedImage= handles.PreProcessedImage;

Y=double(PreProcessedImage);

k=2; % k: number of regions

g=2; % g: number of GMM components

beta=1; % beta: unitary vs. pairwise

EM\_iter=10; % max num of iterations

MAP\_iter=10; % max num of iterations

% fprintf('Performing k-means segmentation\n');

[X,GMM,ShapeTexture]=image\_kmeans(Y,k,g);

[X,Y,GMM]=HMRF\_EM(X,Y,GMM,k,g,EM\_iter,MAP\_iter,beta);

Y=Y\*80;

Y=uint8(Y);

%OutImage=Y;

Y=rgb2gray(Y);

Y=double(Y);

% statsa = glcm(Y,0,ShapeTexture);d

% ExtractedFeatures1=statsa;

axes(handles.axes2);

imshow(Y,[]);

Y=uint8(Y);

% handles.ExtractedFeatures=ExtractedFeatures1;

% disp('exit');

% handles.gmm=1;

% Update handles structure

% guidata(hObject, handles);

warndlg('Process completed');

% run('NNcode.m');

% --- Executes on button press in Classifier.

function Classifier\_Callback(hObject, eventdata, handles)

% hObject handle to Classifier (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

global a;global skw;global Yq;

% gmm=1;

% .gmm=handles.gmm;

%

% .gmm=2;

% load ExtractedFeatures

%

% A=1:20;

% B=21:40;

% C=41:60;

%

% P = [A B C];

% Tc = [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3];

%

% k=2; % k: number of regions

% g=2; % g: number of GMM components

%

% beta=1; % beta: unitary vs. pairwise

% EM\_iter=10; % max num of iterations

% MAP\_iter=10; % max num of iterations

%

% disp('exit');

% file=handles.InputImage;

% [filename, pathname] = uigetfile('\*.jpg', 'Pick a MATLAB code file');

%diff=[];

% file=imread(filename);

% file=rgb2gray(file);

% file=adaptivemedian(file);

% [Xk,GMMk,ShapeTexture]=image\_kmeans(file,k,g);

% PreProcessedImage(:,:,1)=file;

% PreProcessedImage(:,:,2)=file;

% PreProcessedImage(:,:,3)=file;

% stats= gmmsegmentation(Xk,PreProcessedImage,GMMk,k,g,beta,EM\_iter,MAP\_iter,ShapeTexture);

%

% ShapeTexture=stats.ShapeTexture;

%

% for i=1:60

%

% statsa=ExtractedFeature{i};

% ShapeTexturea=statsa.ShapeTexture;

%

%

% diff1(i)=corr2(stats.autoc,statsa.autoc);

% diff2(i)=corr2(stats.contr,statsa.contr);

% diff3(i)=corr2(stats.corrm,statsa.corrm);

% diff4(i)=corr2(stats.cprom,statsa.cprom);

% diff5(i)=corr2(stats.cshad,statsa.cshad);

% diff6(i)=corr2(stats.dissi,statsa.dissi);

% diff7(i)=corr2(stats.energ,statsa.energ);

% diff8(i)=corr2(stats.entro,statsa.entro);

% diff9(i)=corr2(stats.homom,statsa.homom);

% diff10(i)=corr2(stats.homop,statsa.homop);

% diff11(i)=corr2(stats.maxpr,statsa.maxpr);

% diff12(i)=corr2(stats.sosvh,statsa.sosvh);

% diff13(i)=corr2(stats.savgh,statsa.savgh);

% diff14(i)=corr2(stats.svarh,statsa.svarh);

% diff15(i)=corr2(stats.senth,statsa.senth);

% diff16(i)=corr2(stats.dvarh,statsa.dvarh);

% diff17(i)=corr2(stats.denth,statsa.denth);

% diff18(i)=corr2(stats.inf1h,statsa.inf1h);

% diff19(i)=corr2(stats.inf2h,statsa.inf2h);

% diff19(i)=corr2(stats.indnc,statsa.indnc);

% diff19(i)=corr2(stats.idmnc,statsa.idmnc);

% diff20(i)=corr2(ShapeTexture,ShapeTexturea);

%

%

%

%

%

%

% end

%

% [val1 index1]=max(diff1);

% [val2 index2]=max(diff2);

% [val3 index3]=max(diff3);

% [val4 index4]=max(diff4);

% [val5 index5]=max(diff5);

% [val6 index6]=max(diff6);

% [val7 index7]=max(diff7);

% [val8 index8]=max(diff8);

% [val9 index9]=max(diff9);

% [val10 index10]=max(diff10);

% [val11 index11]=max(diff11);

% [val12 index12]=max(diff12);

% [val13 index13]=max(diff13);

% [val14 index14]=max(diff14);

% [val15 index15]=max(diff15);

% [val16 index16]=max(diff16);

% [val17 index17]=max(diff17);

% [val18 index18]=max(diff18);

% [val19 index19]=max(diff19);

% [val20 index20]=max(diff20);

%

%

%

%

% T = ind2vec(Tc);

%

% spread = 1;

%

% net = newpnn(P,T,spread);

%

% A = sim(net,P);

% Ac = vec2ind(A);

% pl(1) = index20;

% p1(2) = index1;

% p1(3) = index2;

% p1(4) = index3;

% p1(5) = index4;

% p1(6) = index5;

% p1(7) = index6;

% p1(8) = index7;

% p1(9) = index8;

% p1(10) = index9;

% p1(11) = index10;

% p1(12) = index11;

% p1(13) = index12;

% p1(14) = index13;

% p1(15) = index14;

% p1(16) = index15;

% p1(17)= index16;

% p1(18) = index17;

% p1(19) = index18;

% p1(20) = index19;

%

run('Findit.p');

% % pl = index20;

% a = sim(net,pl);

% ac = vec2ind(a);

% disp(ac);

ac=Yq;

if(ac==1)

set(handles.edit1,'String','1');

elseif(ac==2)

set(handles.edit1,'String','2');

elseif(ac==3)

set(handles.edit1,'String','3');

else

set(handles.edit1,'String','Error');

end

acc=accuracy\_image(a);

set(handles.edit9,'String',acc);

warndlg('Process completed');

% --- Executes on button press in loaddatabase.

function loaddatabase\_Callback(hObject, eventdata, handles)

% hObject handle to loaddatabase (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

clc;

;

% [filename, pathname] = uigetfile('\*.jpg', 'Pick a MATLAB code file');

% load diff;

k=2; % k: number of regions

g=2; % g: number of GMM components

beta=1; % beta: unitary vs. pairwise

EM\_iter=10; % max num of iterations

MAP\_iter=10; % max num of iterations

helpdlg('In case of error please rerun the same program on system with 8gb ram to avoid empty clusters');

len=1;

len1=21;

len2=41;

h = waitbar(0,'Please wait...');

for num=1:20

%num=15;

waitbar(num/20,h)

filename1=strcat('Beningn',num2str(num),'.jpg');

filename2=strcat('Malign',num2str(num),'.jpg');

filename3=strcat('Malign',num2str(num),'.jpg');

a=imread(filename1);

b=imread(filename2);

c=imread(filename3);

a=rgb2gray(a);

b=rgb2gray(b);

c=rgb2gray(c);

a=adaptivemedian(a);

b=adaptivemedian(b);

c=adaptivemedian(c);

[Xka,GMMka,ShapeTexturea]=image\_kmeans(a,k,g);

[Xkb,GMMkb,ShapeTextureb]=image\_kmeans(b,k,g);

[Xkc,GMMkc,ShapeTexturec]=image\_kmeans(c,k,g);

% a=wlt4(a);

% b=wlt4(b);

%

% c=wlt4(c);

%[Xk,GMMk,ShapeTexture]=image\_kmeans(file,k,g);

PreProcessedImagea(:,:,1)=a;

PreProcessedImagea(:,:,2)=a;

PreProcessedImagea(:,:,3)=a;

PreProcessedImageb(:,:,1)=b;

PreProcessedImageb(:,:,2)=b;

PreProcessedImageb(:,:,3)=b;

PreProcessedImagec(:,:,1)=c;

PreProcessedImagec(:,:,2)=c;

PreProcessedImagec(:,:,3)=c;

statsa= gmmsegmentation(Xka,PreProcessedImagea,GMMka,k,g,beta,EM\_iter,MAP\_iter,ShapeTexturea);

statsb= gmmsegmentation(Xkb,PreProcessedImageb,GMMkb,k,g,beta,EM\_iter,MAP\_iter,ShapeTextureb);

statsc= gmmsegmentation(Xkc,PreProcessedImagec,GMMkc,k,g,beta,EM\_iter,MAP\_iter,ShapeTexturec);

diff{len}=statsa;

diff{len1}=statsb;

diff{len2}=statsc;

len=len+1;

len1=len1+1;

len2=len2+1;

end

save extractedfeatures diff

close(h);

[val index]=max(diff);

disp('exit');

warndlg('Process completed');

function edit1\_Callback(hObject, eventdata, handles)

% hObject handle to edit1 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text

% str2double(get(hObject,'String')) returns contents of edit1 as a double

% --- Executes during object creation, after setting all properties.

function edit1\_CreateFcn(hObject, eventdata, handles)

% hObject handle to edit1 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.

% See ISPC and COMPUTER.

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

set(hObject,'BackgroundColor','white');

end

% --- Executes on button press in pushbutton7.

function pushbutton7\_Callback(hObject, eventdata, handles)

% hObject handle to pushbutton7 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

a=ones(256,256);

axes(handles.axes1);

imshow(a);

axes(handles.axes2);

imshow(a);

clear;

% --- Executes on button press in pushbutton8.

function pushbutton8\_Callback(hObject, eventdata, handles)

% hObject handle to pushbutton8 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

A=1:20;

B=21:40;

C=41:60;

P = [A B C];

Tc = [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3];

T = ind2vec(Tc);

spread = 1;

net = newpnn(P,T,spread);

warndlg('Training Completed Sucessfully');

% --- Executes on button press in pushbutton9.

function pushbutton9\_Callback(hObject, eventdata, handles)

% hObject handle to pushbutton9 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

msgbox('LOAD THE IMAGES....');

pause(2)

msgbox('INITIALIZATION....');

pause(2)

msgbox('CREATE THE DATABASE.....');

pause(2)

for i=1:5

j=num2str(i);

imgname=strcat(j,'.jpg');

TrainData=imread(imgname);

TrainData=imresize(TrainData,[250 250]);

TrainData = im2double(TrainData);

save database TrainData

end

msgbox('DATABASE SAVED SUCCESSFULLY....')

pause(2)

function edit7\_Callback(hObject, eventdata, handles)

% hObject handle to edit7 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit7 as text

% str2double(get(hObject,'String')) returns contents of edit7 as a double

% --- Executes during object creation, after setting all properties.

function edit7\_CreateFcn(hObject, eventdata, handles)

% hObject handle to edit7 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.

% See ISPC and COMPUTER.

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

set(hObject,'BackgroundColor','white');

end

function edit8\_Callback(hObject, eventdata, handles)

% hObject handle to edit8 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit8 as text

% str2double(get(hObject,'String')) returns contents of edit8 as a double

% --- Executes during object creation, after setting all properties.

function edit8\_CreateFcn(hObject, eventdata, handles)

% hObject handle to edit8 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.

% See ISPC and COMPUTER.

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

set(hObject,'BackgroundColor','white');

end

function edit9\_Callback(hObject, eventdata, handles)

% hObject handle to edit9 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit9 as text

% str2double(get(hObject,'String')) returns contents of edit9 as a double

% --- Executes during object creation, after setting all properties.

function edit9\_CreateFcn(hObject, eventdata, handles)

% hObject handle to edit9 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.

% See ISPC and COMPUTER.

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

set(hObject,'BackgroundColor','white');

end

function edit10\_Callback(hObject, eventdata, handles)

% hObject handle to edit10 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit10 as text

% str2double(get(hObject,'String')) returns contents of edit10 as a double

% --- Executes during object creation, after setting all properties.

function edit10\_CreateFcn(hObject, eventdata, handles)

% hObject handle to edit10 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.

% See ISPC and COMPUTER.

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

set(hObject,'BackgroundColor','white');

end

function edit11\_Callback(hObject, eventdata, handles)

% hObject handle to edit11 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit11 as text

% str2double(get(hObject,'String')) returns contents of edit11 as a double

% --- Executes during object creation, after setting all properties.

function edit11\_CreateFcn(hObject, eventdata, handles)

% hObject handle to edit11 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.

% See ISPC and COMPUTER.

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

set(hObject,'BackgroundColor','white');

end

function edit2\_Callback(hObject, eventdata, handles)

% hObject handle to edit2 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text

% str2double(get(hObject,'String')) returns contents of edit2 as a double

% --- Executes during object creation, after setting all properties.

function edit2\_CreateFcn(hObject, eventdata, handles)

% hObject handle to edit2 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.

% See ISPC and COMPUTER.

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

set(hObject,'BackgroundColor','white');

end

function edit3\_Callback(hObject, eventdata, handles)

% hObject handle to edit3 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit3 as text

% str2double(get(hObject,'String')) returns contents of edit3 as a double

% --- Executes during object creation, after setting all properties.

function edit3\_CreateFcn(hObject, eventdata, handles)

% hObject handle to edit3 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.

% See ISPC and COMPUTER.

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

set(hObject,'BackgroundColor','white');

end

function edit4\_Callback(hObject, eventdata, handles)

% hObject handle to edit4 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit4 as text

% str2double(get(hObject,'String')) returns contents of edit4 as a double

% --- Executes during object creation, after setting all properties.

function edit4\_CreateFcn(hObject, eventdata, handles)

% hObject handle to edit4 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.

% See ISPC and COMPUTER.

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

set(hObject,'BackgroundColor','white');

end

function edit5\_Callback(hObject, eventdata, handles)

% hObject handle to edit5 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit5 as text

% str2double(get(hObject,'String')) returns contents of edit5 as a double

% --- Executes during object creation, after setting all properties.

function edit5\_CreateFcn(hObject, eventdata, handles)

% hObject handle to edit5 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.

% See ISPC and COMPUTER.

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

set(hObject,'BackgroundColor','white');

end

function edit6\_Callback(hObject, eventdata, handles)

% hObject handle to edit6 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit6 as text

% str2double(get(hObject,'String')) returns contents of edit6 as a double

% --- Executes during object creation, after setting all properties.

function edit6\_CreateFcn(hObject, eventdata, handles)

% hObject handle to edit6 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.

% See ISPC and COMPUTER.

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

set(hObject,'BackgroundColor','white');

end