PÉCSI TUDOMÁNYEGYETEM

UNIVERSITY OF PÉCS

DIPLOMA THESIS

Joseph Twal

2022

Saturday, May 7, 2022

University of Pécs

Faculty of Engineering and Information Technology

Computer Science Engineering Master

DIPLOMA THESIS

Predicting Covid-19 Cases In
Chest X-RAY Scans Using ML & DL
And
Lung Segmentation Using
Morphological Image Processing Methods

Author: Joseph Twal
Supervisor: Dr. Sári Zoltan

Pécs

2022

Saturday, May 7, 2022

UNIVERSITY OF PÉCS
FACULTY OF ENGINEERING AND INFORMATION TECHNOLOGY

Number:

………………../2022

Computer Science Engineering Master

DIPLOMA THESIS

....

For student

The title and the topic of the diploma thesis, which must be submitted before the final exam, are the following:

Title: Predicting Covid-19 cases in chest X-RAY scans using ML & DL and Lung Segmentation using Morphological process imaging methods.

Tasks:

- Classify and predict Covid-19 cases and determine either having Covid-19, healthy or other Viral Pneumonia symptoms on the lungs.
- Separate Lungs using advance image processing morphological methods.

Responsible department: Engineering And Information Technology

External supervisor (if any): NA
Institution: NA

Supervisor: Dr. Sári Zoltán
Institution: University Of Pécs - Faculty of Engineering and Information Technology

Pécs, date: 2022/03/03

Prof. Dr. Péter Iványi
Head of Department

DECLARATION

I declare, that this diploma thesis is the result of my own work. All references and external works have been identified and cited. I have not used any other external help.

The results of my diploma thesis can be used by the university for its own purposes free of charge.

Pécs, 2022/03/03

....
signature of the student

DEDICATION

I dedicate my dissertation work to my family. A special feeling of gratitude to my loving parents, Ibrahim and Samira whose words of encouragement and push for tenacity ring in my ears. My brother Daniel and my sister Maria have never left my side and you have been my best cheerleaders. You are very special and dear to my heart.

I also dedicate this dissertation to my many friends who have supported me throughout the process. I will always appreciate all of you and be grateful and thankful for helping me develop both my personal and career life.

I dedicate this work and give special thanks to my Professors, doctors and supervisor Dr. Sári Zoltán for being there for me throughout the entire master program.

Table of Contents

DECLARATION ____ iii
DEDICATION ____ iv
ABSTRACT ____ vii
1. Introduction ____ - 1 -
1.1 Background ____ - 1 -
1.2 Treatment for COVID-19 ____ - 3 -
1.3 Scope of work, Literature review, and previous studies ____ - 5 -
1.3.1 Scope of work and Literature review ____ - 5 -
1.3.2 Previous studies ____ - 6 -
2 Tools & Database ____ - 8 -
2.1 Tools ____ - 8 -
2.1.1 Programming language [24] ____ - 8 -
2.1.2 Convolutional Neural Network (CNN) [28] ____ - 10 -
2.1.3 Visual Geometry Group (VGG-16) [29] ____ - 11 -
2.1.4 Support Vector Machine (SVM) [30] ____ - 13 -
2.1.5 K-nearest neighbors (KNN) [31] ____ - 16 -
2.1.6 Morphological Operations [32] ____ - 18 -
2.2 Database [33-34] ____ - 19 -
3 Methodology, Challenges & Results ____ - 20 -
3.1 Methodology ____ - 20 -
3.1.1 COVID classification & prediction ____ - 20 -
3.1.2 Lung Segmentation ____ - 25 -
3.2 Challenges ____ - 26 -
3.3 Results ____ - 27 -
3.3.1 Results of Classification & Prediction ____ - 27 -
3.3.2 Lung Segmentation Results ____ - 33 -
3.3.3 COVID-19 Application ____ - 34 -
4 Conclusion ____ - 35 -
5 References ____ - 36 -
6 Appendix A ____ - 39 -

Figure 1-1 Covid sample - 2 -
Figure 1-2 Covid sample - 2 -
Figure 1-3 Covid sample - 2 -
Figure 1-4 norma sample - 2 -
Figure 1-5 norma sample - 2 -
Figure 1-6 norma sample - 2 -
Figure 2-1 CNN - 10 -
Figure 2-2 VGG16 architecture - 12 -
Figure 2-3 SVM 2D example - 13 -
Figure 2-4 non-linear data - 14 -
Figure 2-5 3D plane conversion - 14 -
Figure 2-6 3D HYPERPLANE - 15 -
Figure 2-7 convert back to 2D - 15 -
Figure 2-8 knn data set - 17 -
Figure 2-9 nearest neighbors - 17 -
Figure 3-1 proposed methodology - 20 -
Figure 3-2 VGG16 LAYERS - 22 -
Figure 3-3 SEGMENTATION WORKFLOW - 25 -
Figure 3-4 CONFUSION MATRIX EXAMPLE - 27 -
Figure 3-5 MULTI CLASSIFICATION - 29 -
Figure 3-6 KNN CM - 31 -
Figure 3-7 3 SVM LINEAR CM - 32 -
Figure 3-8 SVM (RBF) CM - 32 -
Figure 3-9 segmentation process and result - 33 -
Figure 3-10 GUI OUTPUT EXAMPLE - 34 -
Figure 6-1 CM OUTPUT FROM GUI - 39 -
Figure 6-2 GUI STARTING STEPS - 41 -
Figure 6-3 GUI OUTPUT - 41 -

Table 1-1 Vaccine types - 4 -
Table 1-2 Summary of COVID-19 prediction algorithms. - 7 -
Table 3-1 ML measures comparison - 31 -
Table 6-1 MODEL PARAMETERS - 40 -

ABSTRACT

The World Health Organization (WHO) has declared an epidemic emergency around the globe because of new variant of SARS-CoV-2 virus causes a respiratory disease in humans, known as COVID-19. The confirmatory diagnostic of this disease occurs through the real-time reverse transcription and polymerase chain reaction test (RT-qPCR). However, the time of obtaining the results limits the application of the enormous test. Thus, chest X-ray images are analyzed to help and speed the process of the diagnoses of the disease. However, during an outbreak of a disease that causes respiratory problems, radiologists may be overwhelmed with analyzing medical images.

In this study, Feature extraction method based on CNN were used to extract important features and use it as an input for machine learning classification techniques to determine and predict COVID, NON-COVID, and Viral Pneumonia symptoms on the lungs. The proposed methods were using VGG-16 as a feature extractor model and two machine-learning algorithms is used K-Nearest Neighbor (KNN) and Support Vector Machine (SVM). The results were very astonishing in terms of accuracy and F1 scores for the proposed models and it is reliable to make a good indication and prediction of the disease. The best result was using VGG-16 as extractor and SVM with an RBF kernel to achieve accuracy of 98% while KNN achieved 97%.

The second focus of the study is to use and apply Advance image processing methods and techniques to be able to segment the lungs into right and left side lungs. Multiple morphological methods are used in the study to achieve the best possible outcomes and to make the separation as accurate as it can be.

1. Introduction

1.1 Background

Coronaviruses are a large family of viruses that cause illness ranging from the common cold to more severe diseases, such as Middle East Respiratory Syndrome (MERS-CoV) and Severe Acute Respiratory Syndrome (SARS-CoV) [1]. This novel coronavirus (COVID-2019) [2–4] is a new strain not previously identified in humans. A common clinical feature of severe COVID-19 infection is pneumonia [5–8].

The most common symptoms of the disease can vary and include dyspnea, high fever, runny nose, and cough, other symptoms may include diarrhea, muscle pain, sore throat, sputum production, abdominal pain and loss of taste and smell [10-16]. Even though the majority of cases end in mild symptoms, some progress to pneumonia and multi-organ failure [10, 15].

COVID-19 is typically spread during close contact and via respiratory droplets produced when people sneeze or cough. Respiratory droplets may be formed during breathing but it is not well thought out airborne. It may also spread through fomite transmission. For example, touching a fomite (contaminated surface) and then touching the body's mucous membranes, such as the mouth, nose, or eyes, could potentially introduce the pathogen into the body. This is why proper and frequent hand washing is so important. It is most contagious when people are symptomatic, although spread may be possible before symptoms appear [10, 11]. COVID-19 can live on surfaces up to 72 hours [15]. Time from exposure to onset of symptoms is generally between two and fourteen days, with an average of five days [17]. The standard method of diagnosis is by reverse transcription polymerase chain reaction (rRT-PCR) from a nasopharyngeal swab. These cases can most commonly be diagnosed using chest X-ray imaging analysis for the abnormalities [9].

Recommended measures to prevent infection include frequent hand washing, social distancing (maintaining physical distance from others, especially from those with symptoms), covering coughs and sneezes with a tissue or inner elbow, and keeping unwashed hands away from the face. The use of masks is recommended by some national health authorities for those who suspect they have the virus and their caregivers, but not for the general public, although simple cloth masks may be used by those who desire them. There is no vaccine or specific antiviral treatment for COVID-19. Management involves treatment of symptoms, isolation, supportive care and experimental measures [19].

COVID-19 attacks the epithelial cells that line our respiratory tract, X-rays can be used to analyze the lungs of a patient. Doctors often use X-ray images to diagnose pneumonia, lung irritation, swellings, and/or distended lymph nodes. X-RAY Images with the help of the machine learning can easily help radiologist and assist them to make an early prediction of COVID cases without using any external examination toolkit, just by X-RAY and machine learning we can achieve high accuracy of prediction by making our program learn from the images and extract features and use these to distinguish COVID from normal lungs or other Viral Pneumonia diseases on the lungs.

Some sample of X-RAY images provided below between the three categories of the lungs (COVID, Normal, and Viral Pneumonia) *(figure 1-1 – 1-9)*

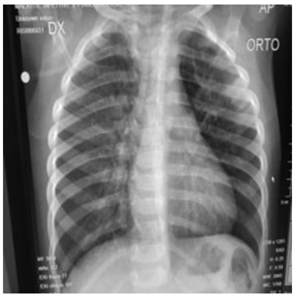


Figure 1-1 Covid sample

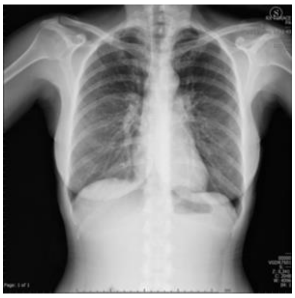


Figure 1-2 Covid sample

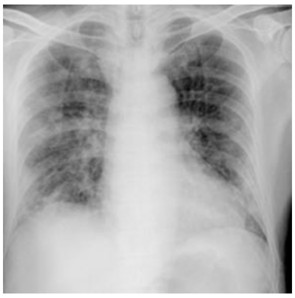


Figure 1-3 Covid sample

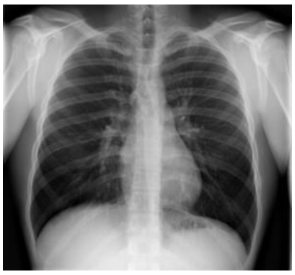


Figure 1-4 norma sample

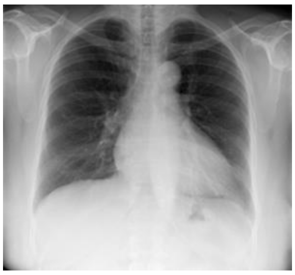


Figure 1-5 norma sample

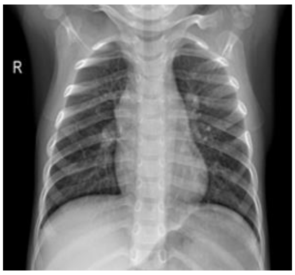


Figure 1-6 norma sample

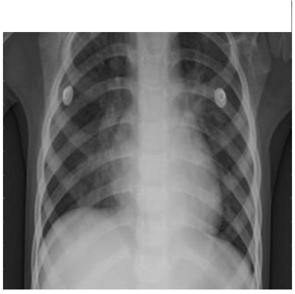


Figure 1-7 Viral sample

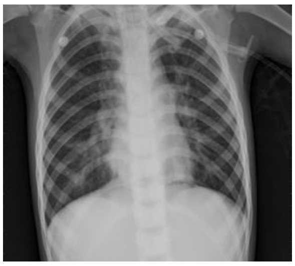


Figure 1-8 Viral sample

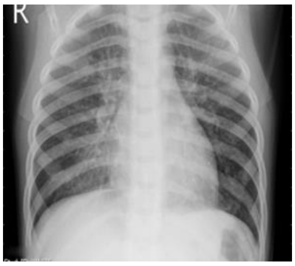


Figure 1-9 Viral sample

1.2 Treatment for COVID-19

The FDA has approved the **antiviral drug** Veklury (remdesivir) for adults and certain pediatric patients with COVID-19.

During public health emergencies, the FDA may authorize the use of unapproved drugs or unapproved uses of approved drugs under certain conditions. This is called an **Emergency Use Authorization (EUA).** These products are not a substitute for vaccination against COVID-19.

For example, the FDA has issued EUAs for several **monoclonal antibody treatments** for COVID-19 for the treatment, and in some cases prevention (prophylaxis), of COVID-19 in adults and pediatric patients. Monoclonal antibodies are laboratory-made molecules that act as substitute antibodies. They can help your immune system recognize and respond more effectively to the virus, making it more difficult for the virus to reproduce and cause harm.

The FDA continues to work with developers, researchers, manufacturers, the National Institutes of Health, and other partners to help expedite the development and availability of therapeutic drugs and biological products to prevent or treat COVID-19.

Researchers are studying drugs that are already approved for other health conditions as possible treatments for COVID-19. Additionally, the FDA created the **Coronavirus Treatment Acceleration Program (CTAP)** to use every available means to assess new treatments and move them to patients as quickly as possible. [20]

Researches, doctors, and almost everyone who is working or contributing to the pharmaceuticals realm are working hard to make the epidemic fade away as soon as possible, therefor, they are focusing their mind and research to develop the vaccine against COVID and it is astonishing of they have achieved, multiple vaccines were developed and highly effective against CVOID while reducing the symptoms on the patients, the following table *(TABLE 1- 1 VACCINE TYPES)* contains the three major vaccines and the differences between them.

[21]	**Ages Recommended**	**Primary Series**	**Booster Dose**	**When Fully Vaccinated**
Pfizer-BioNTech	5+ years old	2 doses, Given 3 weeks (21 days) apart	Everyone ages 18 years and older should get a booster dose of either Pfizer-BioNTech or Moderna (COVID-19 vaccines) 5 months after the last dose in their primary series. Teens 12-17 years old should get a Pfizer-BioNTech COVID-19 Vaccine booster 5 months after the last dose in their primary series.	2 weeks after 2nd dose
Moderna	18+ years old	2 doses, Given 4 weeks (28 days) apart	Everyone ages 18 years and older should get a booster dose of either Pfizer-BioNTech or Moderna (COVID-19 vaccines) 5 months after the last dose in their primary series.	2 weeks after 2nd dose
Johnson**&** **Johnson's Janssen**	18+ years old	1 dose	Everyone ages 18 years and older should get a booster dose of either Pfizer-BioNTech or Moderna (mRNA COVID-19 vaccines) at least 2 months after the first dose of J&J/Janssen COVID-19 vaccine	2 weeks after 1st dose

TABLE 1-1 VACCINE TYPES

1.3 Scope of work, Literature review, and previous studies

1.3.1 Scope of work and Literature review

Deep learning has shown a dramatic increase in the medical applications in general and specifically in medical image based diagnosis. Deep learning models performed prominently in computer vision problems related to medical image analysis. The ANNs outperformed other conventional models and methods of image analysis. Due to the very promising results provided by CNNs in medical image analysis and classification, they are considered as de facto standard in this domain. CNN has been used for a variety of classification tasks related to medical diagnosis such as lung disease, detection of malarial parasite in images of thin blood smear, breast cancer detection, wireless endoscopy images, interstitial lung disease, CAD-based diagnosis in chest radiography, diagnosis of skin cancer by classification, and automatic diagnosis of various chest diseases using chest X-ray image classification. Since the emergence of COVID-19 in December 2019, numerous researchers are engaged with the experimentation and research activities related to diagnosis, treatment, and management of COVID-19 [22].

The Aim of this study and the main scope of work here is to implement an application and use the correct tools, algorithms, and deep learning methods to speed up the process for radiologist and make a very fast prediction of COVID in patients to reduce the time and the effort and make it as accurate as possible. After taking all the consideration and pay a good attention to all aspect of the problem, the solution is built on using pre-defined models of CNN; the model is VGG-16 and it used as a feature extractor from the X-RAY images and feed these images as an input to SVM to make a classification and a prediction on these images.

Second minor scope of this study is to make a simple application to make a segmentation of the lungs into two parts (left and right), by use advance image processing methods and morphological operations, the segmentation will be easier to accomplish and the aim of the segmentation is to focus only on the side that is effected by COVID and eliminating other areas and unnecessary areas on the X-RAY images.

1.3.2 Previous studies

A great number of researchers have been working on finding solution to mitigate the epidemic and the problems related to such applications being developed to make a prediction for such difficult disease, the table below *(TABLE 1-2 Summary of covid-19)* summarize most of the studies and work done by researchers and to make prediction algorithms to test the data (Radiological (chest X-rays and C.T. images), RT-PCR, and Clinical data) along with the highest prediction results of the selected previous studies. Only the best-obtained results of different ML/DL techniques on C.T., X-ray images, RT-PCR, and clinical blood test data were mentioned. [23]

Test Type	ML/DL Techniques	Prediction Results	County	Cited by No of Papers
CT Images	CNN, COVNet	AUC 0.96	China	553
CT Images	CNNs, ResNet-101 & Xception	AUC of 0.99, Sensitivity 98.02%, Specificity 99.51%	Iran	120
CT Images	3-D DNN, DeCoVNet	0.9	China	205
CT Images	Inception Transfer learning model	Accuracy of 89.5% with Specificity of 0.88 and Sensitivity of 0.87	China	376
CT Images	DRE-Net	A.U.C. of 0.99 Sensitivity of 0.93	China	198
CT Images	2D and 3D deep learning (Resnet-50-2D)	AUC of 0.99 Sensitive 92.2% Specificity 92.2%	China	306
CT Images	Classification Stage 1 SVM, Stage 2 GLCM, GLSZ MDWT	Accuracy of 99.68%	Turkey	103
CT Images	Multilayer perceptron and LSTM	AUC of 0.954	China	48
CT Images	Combined model 3D UNet++ and RestNet-50	AUC of 0.991 Sensitivity of 0.974 and specificity of 0.922	China	99
CT Images	3D-DNN, COVID-19Net	AUC 0.86 Sensitivity of 79.35% and specificity of 71.43%	China	95
CT Images	CNN, Multi-task learning, self-supervised learning, DenseNet-169	Accuracy of 0.89 and AUC 0.90	China	175
CT Images	Decision tree, Proposed COVIDiag model	Accuracy of 91.4% sensitivity of 93.24%, and specificity of 90.32%	Iran	7
X-RAY Images	CNN, CoroNet	Overall Accuracy 89.6%	India	159
X-RAY Images	CNN, VGG16	Average accuracy 0.97%	Italy	69
X-RAY Images	DarkCovidNet	Accuracy of 98.08%	Turkey	417
X-RAY Images	CNN, MobileNet	Accuracy of 96.78% Sensitive 98.66% Specificity 96.46%	Greece	480

X-RAY Images	D.N.N., VGG-19, ResNet-50, COVID-Net	Accuracy of 93.3%	Canada	558
X-RAY Images	CNN, RestNet50 + SVM	Accuracy of 95.33%	Egypt	251
X-RAY Images	COV19-ResNet,	Accuracy of 97.61%	Turkey	1
X-RAY Images	ResNet50, DenseNet201, Inception-v3, and Xception	AUC 0.996,	USA	2
X-RAY Images	CNN, Bayesian ResNet50V2	Accuracy of 89.92%	UK	104
X-RAY Images	DCNN, CheXNet + DenseNet-201	Accuracy of 99.7%	Qatar	161
X-RAY Images	CNN, Classification Grad-CAM	AUC 95.13%, sensitivity of 90%, specificity of 87.84%	China	161
X-RAY Images	COVID-ResNet	Accuracy of 96.23%	USA	125
X-RAY Images	D-CNN, DeTraC	Accuracy of 95.12%	Egypt	161
X-RAY Images	D.N.N., Deep COVID Explainer	Positive Predictive Value 96.12% and recall of 94.3%	Germany	34
X-RAY Images	VGG-16, VGG-19	Accuracy of 87.49%	Australia	5
X-RAY Images	CNN, AlexNet, GoogleNet, SqueezeNet	Overall accuracy 99%	Saudi Arabia	4
X-RAY Images	K.N.N., ANN, D.T., SVM	Overall accuracy of 93.41%	India	8
RT-PCR	Support Vector Machine,RF, NN ,LR	AUC 0.847, Sensitivity 0.67, Specificity 0.85	Brazil	31
RT-PCR	K.N.N. Decision Tree,	Accuracy of 80%	China	147
Clinical Blood Test	Random Forests	Accuracy of 0.9512,	China	27
Clinical Blood Test	(XGBoost)	A.U.C. of 0.97, Sensitivity 81.9%, and specificity of 97.9%	Switzerland	5

TABLE 1-2 SUMMARY OF COVID-19 PREDICTION ALGORITHMS.

2 Tools & Database

2.1 Tools

In this section, we describe the tools, background and a summary of the theoretical aspect of the methods being used to complete and accomplish the work of the study.

2.1.1 Programming language [24]

Python is the programming language used in the study; Python is one of the most popular programming languages used by developers today. Guido Van Rossum created it in 1991 and ever since its inception has been one of the most widely used languages along with C++, Java, etc.

In our endeavor to identify what is the best programming language for AI and neural network, Python has taken a big lead.

Python is an Interpreted language which means that it does not need to be compiled into machine language instruction before execution and can be used by the developer directly to run the program. This makes it comprehensive enough for the language to be interpreted by an emulator or a virtual machine on top of the native machine language, which is what the hardware understands.

It is a High-Level Programing language and can be used for complicated scenarios. High-level languages deal with variables, arrays, objects, complex arithmetic or Boolean expressions, and other abstract computer science concepts to make it more comprehensive thereby exponentially increasing its usability.

Python is also a General-purpose programming language, which means it can be used across domains and technologies.

Python also features dynamic type system and automatic memory management supporting a wide variety of programming paradigms including object-oriented, imperative, functional and procedural to name a few.

Python is available for all Operating Systems also has an open-source offering titled CPython, which is garnering widespread popularity as well.

Python offers the least code among others and is in fact 1/5 the number compared to other OOP languages. No wonder it is one of the most popular in the market today.

- Python has Prebuilt Libraries like Numpy for scientific computation, Scipy for advanced computing and Pybrain for machine learning (Python Machine Learning) making it one of the best languages For AI.

- Python developers around the world provide comprehensive support and assistance via forums and tutorials making the job of the coder easier than any other popular languages.

- Python is platform Independent and is hence one of the most flexible and popular choices for use across different platforms and technologies with the least tweaks in basic coding.

- Python is the most flexible of all others with options to choose between OOPs approach and scripting. You can also use IDE itself to check for most codes and is a boon for developers struggling with different algorithms.

Libraries used in python:

- **NumPy** is used as a container for generic data comprising of an N-dimensional array object, tools for integrating C/C++ code, Fourier transform, random number capabilities, and other functions.
- **Pandas**, an open source library that provides users with easy-to-use data structures and analytic tools for Python.
- **Matplotlib** is another service, which is a 2D plotting library creating publication quality figures. You can use matplotlib to up to six graphical users interface toolkits, web application servers, and Python scripts.
- **Scikit-learn** is probably the most useful library for machine learning in Python. The sklearn library contains many efficient tools for machine learning and statistical modeling including classification, regression, and clustering and dimensionality reduction.[25]
- **Keras** is a powerful and easy-to-use free open source Python library for developing and evaluating deep learning models. It wraps the efficient numerical computation libraries Theano and TensorFlow and allows you to define and train neural network models in just a few lines of code.[26]
- **TensorFlow** is an open source library for fast numerical computing. It was created and is maintained by Google and released under the Apache 2.0 open source license. The API is nominally for the Python programming language, although there is access to the underlying C++ API. 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.[27]

❖ **OpenCV-Python** is an appropriate tool for fast prototyping of computer vision problems.

2.1.2 Convolutional Neural Network (CNN) [28]

A convolutional neural network (CNN) is a type of artificial neural network used in image recognition and processing that is specifically designed to process pixel data.

CNNs are powerful image processing, artificial intelligence (AI) that use deep learning to perform both generative and descriptive tasks, often using machine vison that includes image and video recognition, along with recommender systems and natural language processing (NLP).

A neural network is a system of hardware and/or software patterned after the operation of neurons in the human brain. Traditional neural networks are not ideal for image processing and must be fed images in reduced-resolution pieces. CNN have their “neurons” arranged more like those of the frontal lobe, the area responsible for processing visual stimuli in humans and other animals. The layers of neurons are arranged in such a way as to cover the entire visual field avoiding the piecemeal image processing problem of traditional neural networks.

A CNN uses a system much like a multilayer perceptron that has been designed for reduced processing requirements. The layers of a CNN consist of an input layer, an output layer and a hidden layer that includes multiple convolutional layers, pooling layers, fully connected layers and normalization layers. The removal of limitations and increase in efficiency for image processing results in a system that is far more effective, simpler to trains limited for image processing and natural language processing. Refer to *figure 2-1 CNN* as an example of CNN

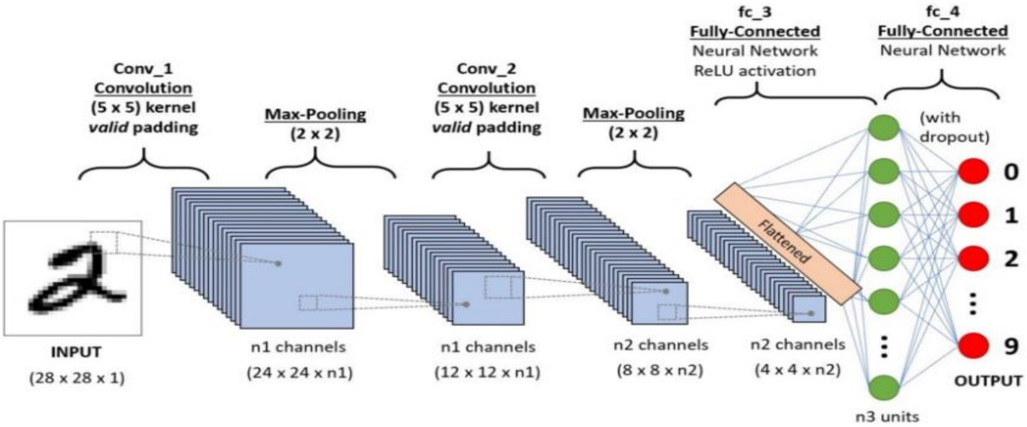


FIGURE 2-1 CNN

2.1.3 Visual Geometry Group (VGG-16) [29]

VGG16 is a simple and widely used Convolutional Neural Network (CNN) Architecture used for ImageNet, a large visual database project used in visual object recognition software research. The VGG16 Architecture was developed and introduced by Karen Simonyan and Andrew Zisserman from the University of Oxford, in the year 2014, through their article “Very Deep Convolutional Networks for Large-Scale Image Recognition.” ‘VGG’ is the abbreviation for Visual Geometry Group, which is a group of researchers at the University of Oxford who developed this architecture, and ‘16’ implies that this architecture has 16 layers.

The VGG16 model achieved 92.7% top-5 test accuracy in ImageNet, which is a dataset of over 14 million images belonging to 1000 classes. It was one of the famous models submitted to ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in the year 2014. It made improvements over AlexNet architecture by replacing large kernel-sized filters (11 and 5 in the first and second convolutional layer, respectively) with multiple three × three kernel-sized filters one after another. VGG16 was trained for weeks using NVIDIA Titan Black GPUs.

VGG16 is used in many deep learning image classification techniques and is popular due to its ease of implementation. VGG16 is extensively used in learning applications due to the advantage that it has.

VGG16 is a CNN Architecture, which was used to win the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2014. It is still one of the best vision architecture to date.

- **VGG16 architecture**

  - During training, the input to the convnets is a fixed-size 224 x 224 RGB image. Subtracting the mean RGB value computed on the training set from each pixel is the only pre-processing done here. The image is passed through a stack of convolutional (conv.) layers, where filters with a very small receptive field: 3 × 3 (which is the smallest size to capture the notion of left/right, up/down, center and has the same effective receptive field as one 7 x 7), is used. It is deeper, has more non-linearities, and has fewer parameters. In one of the configurations, 1 × 1 convolution filters, which can be seen as a linear transformation of the input channels (followed by non-linearity), are also utilized. The convolution stride and the spatial padding of conv. layer input is fixed to 1 pixel for 3 x 3 convolutional layers, which ensures that the spatial resolution is preserved after convolution. Five max-pooling layers, which follow some of the convolutional layers, helps in spatial pooling. Max-pooling is performed over a 2×2 pixel window, with stride 2.

- There are three Fully-Connected (FC) layers that follow a stack of convolutional layers (these have different depths in different architectures): the first two have 4096 channels each, the third performs 1000-way ILSVRC classification and thus contains 1000 channels (one for each class). The final layer is the soft-max layer. The configuration of the fully connected layers is the same in all networks.
- The 16 layer VGG architecture was the best performing, and it achieved a top-5 error rate of 7.3% (92.7% accuracy) in ILSVRC — 2014, as mentioned above. VGG16 had significantly outperformed the previous generation of models ILSVRC — 2012 and ILSVRC — 2013 competitions.
- The VGG16 architecture is Summarized in *figure 2-2 VGG16 architecture*, shown below:

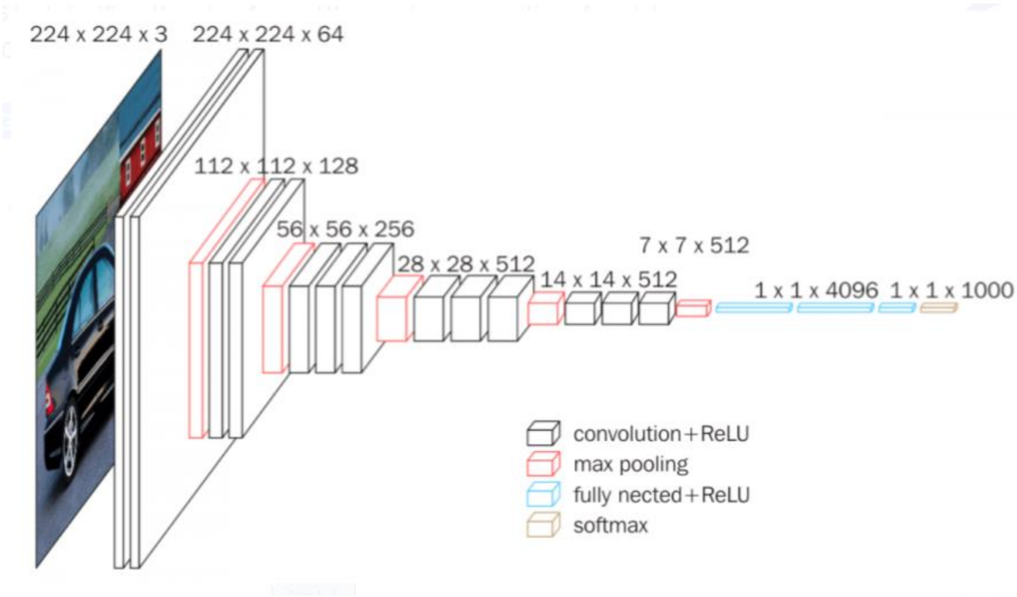


FIGURE 2-2 VGG16 ARCHITECTURE

2.1.4 Support Vector Machine (SVM) [30]

A support vector machine (SVM) is a supervised machine-learning model that uses classification algorithms for two-group classification problems. After giving SVM model sets of labeled training data for each category, they are able to categorize new data.

Compared to newer algorithms like neural networks, they have two main advantages: higher speed and better performance with a limited number of samples (in the thousands). This makes the algorithm very suitable for text classification problems, where it is common to have access to a dataset of at most a couple of thousands of tagged samples.

A support vector machine takes data points and outputs the hyperplane (which in two dimensions it is simply a line) that best separates the tags. This line is the **decision boundary**: anything that falls to one side of it we will classify as *blue*, and anything that falls to the other as *red*. Refer to *figure 2-3 SVM 2D EXAMPLE*

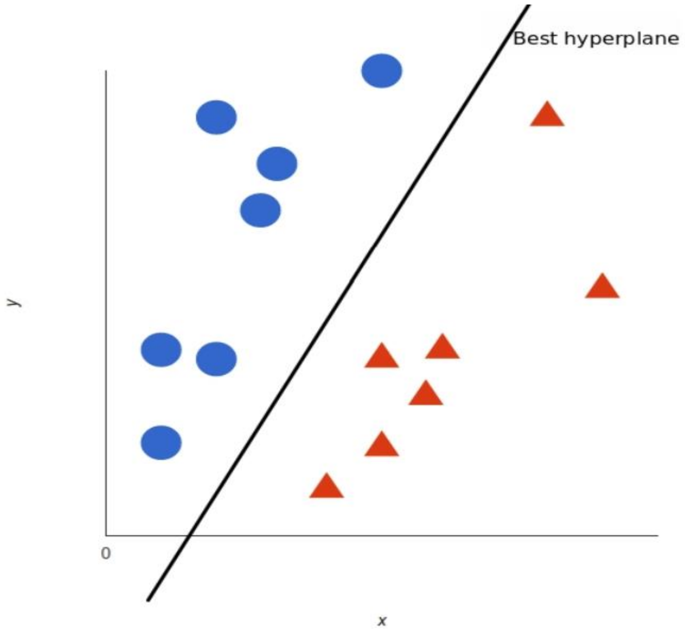


FIGURE 2-3 SVM 2D EXAMPLE

- ✓ **<u>Best hyperplane For SVM</u>**: The one maximizes the margins from both tags. In other words: the hyperplane whose distance to the nearest element of each tag is the largest.

For non-linear data, which is not represented in 2D manner, we have to convert the data or do some transformation from 2D planes into 3D planes and make the data separable by a plane instead of a simple line; *figure 2-4 non-linear data* shows an example of non-linear data and the separation of the data will be impossible to do and the trick here is transform the data into a higher dimension planes using simple mathematical equations $x^2 + y^2 =$ z this will give us a third dimension where the separation of these data can be done; *figure 2-5 3D plane conversion* can demonstrate this.

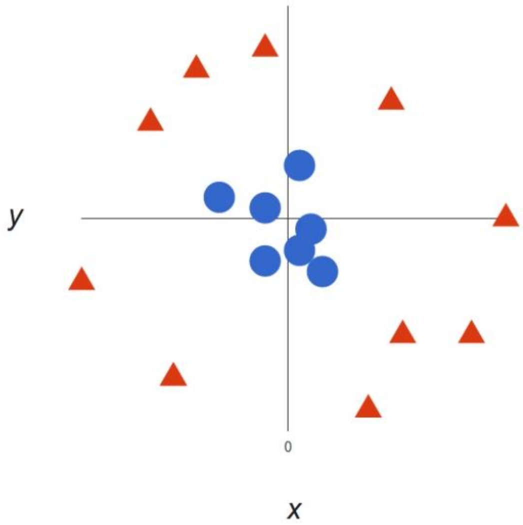


FIGURE 2-4 NON-LINEAR DATA

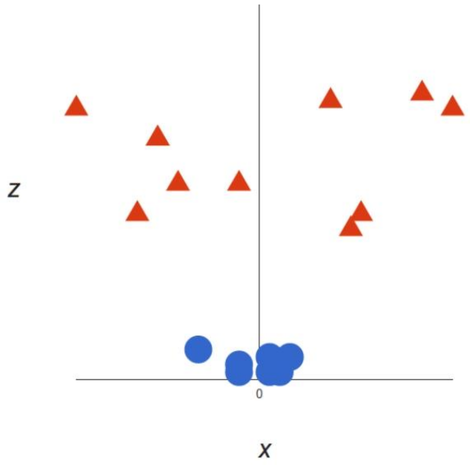


FIGURE 2-5 3D PLANE CONVERSION

After the transformation into higher dimension, we can the data separable by make the previous steps as before a separate them with the best hyperplane but we have to keep into consideration that it is a plane now instead of a line since we are in 3D dimension, referring to the *figures 2-6,2-7*; we can see that converting back into the original 2D plane the line separate the data will be a circle instead of a simple line.

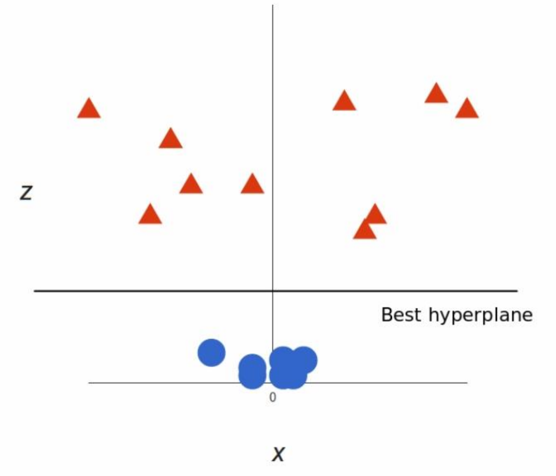


FIGURE 2-6 3D HYPERPLANE

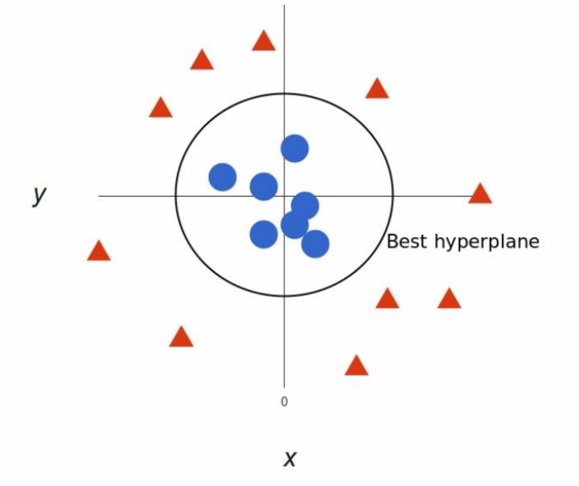


FIGURE 2-7 CONVERT BACK TO 2D

2.1.5 K-nearest neighbors (KNN) [31]

K-nearest neighbors (KNN) algorithm is a type of supervised ML algorithm, which can be used for both classification as well as regression predictive problems. However, it is mainly used for classification predictive problems in industry.

The following two properties would define KNN well:

- **Lazy learning algorithm** − KNN is a lazy learning algorithm because it does not have a specialized training phase and uses all the data for training while classification.
- **Non-parametric learning algorithm** − KNN is also a non-parametric learning algorithm because it does not assume anything about the underlying data.

K-nearest neighbors (KNN) algorithm uses 'feature similarity' to predict the values of new data points which further means that the new data point will be assigned a value based on how closely it matches the points in the training set. We can understand its working with the help of following steps −

Step 1 − For implementing any algorithm, we need dataset. So during the first step of KNN, we must load the training as well as test data.

Step 2 − Next, we need to choose the value of K i.e. the nearest data points. K can be any integer.

Step 3 − For each point in the test data do the following −

- **3.1** − Calculate the distance between test data and each row of training data with the help of any of the method namely: Euclidean, Manhattan or Hamming distance. The most commonly used method to calculate distance is Euclidean.
- **3.2** − Now, based on the distance value, sort them in ascending order.
- **3.3** − Next, it will choose the top K rows from the sorted array.
- **3.4** − Now, it will assign a class to the test point based on most frequent class of these rows.

Step 4 – End

The following *figures 2-8,2-9* are an example to understand the concept of K and working of KNN algorithm, referring to the steps above we need to classify new data point with black dot (at point 60,60) into blue or red class. We are assuming K = 3 i.e. it would find three nearest data points. We can see the three nearest neighbors of the data point with black dot. Among those three, two of them lies in Red class hence the black dot will also be assigned in red class.

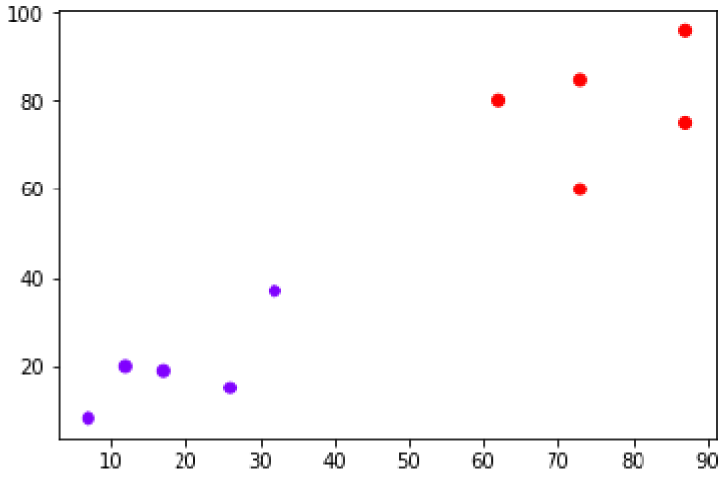


FIGURE 2-8 KNN DATA SET

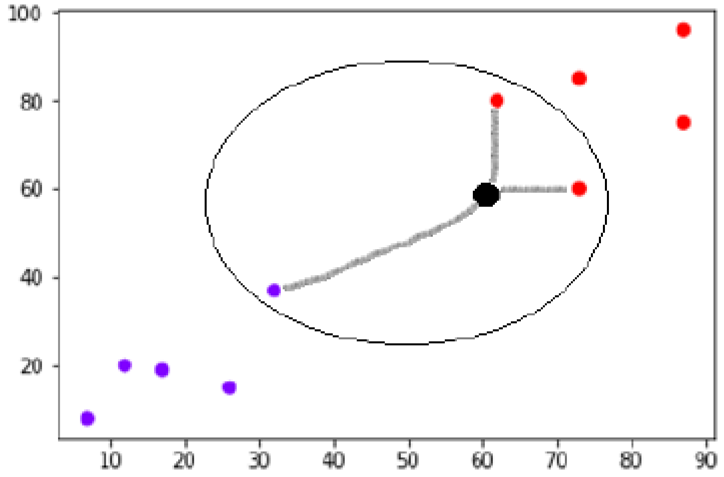


FIGURE 2-9 NEAREST NEIGHBORS

2.1.6 Morphological Operations [32]

Morphology is a broad set of image processing operations that process images based on shapes. Morphological operations apply a structuring element to an input image, creating an output image of the same size. In a morphological operation, the value of each pixel in the output image is based on a comparison of the corresponding pixel in the input image with its neighbors

The most basic morphological operations are dilation and erosion. Dilation adds pixels to the boundaries of objects in an image, while erosion removes pixels on object boundaries. The number of pixels added or removed from the objects in an image depends on the size and shape of the *structuring element* used to process the image. In the morphological dilation and erosion operations, the state of any given pixel in the output image is determined by applying a rule to the corresponding pixel and its neighbors in the input image. The rule used to process the pixels defines the operation as a dilation or an erosion. This table lists the rules for both dilation and erosion.

- Dilation
  - The value of the output pixel is the maximum value of all pixels in the neighborhood. In a binary image, a pixel is set to 1 if any of the neighboring pixels have the value 1.
  - Morphological dilation makes objects more visible and fills in small holes in objects.
- Erosion
  - The value of the output pixel is the minimum value of all pixels in the neighborhood. In a binary image, a pixel is set to 0 if any of the neighboring pixels have the value 0.
  - Morphological erosion removes islands and small objects so that only substantive objects remain.

Operations Based on Dilation and Erosion

Dilation and erosion are often used in combination to implement image-processing operations. For example, the definition of a morphological *opening* of an image is an erosion followed by a dilation, using the same structuring element for both operations. You can combine dilation and erosion to remove small objects from an image and smooth the border of large objects.

- Close
  - Perform morphological closing. The closing operation dilates an image and then erodes the dilated image, using the same structuring element for both operations.

- Morphological closing is useful for filling small holes from an image while preserving the shape and size of the objects in the image.

2.2 Database [33-34]

A team of researchers from Qatar University, Doha, Qatar, and the University Of Dhaka, Bangladesh along with their collaborators from Pakistan and Malaysia in collaboration with medical doctors have created a database of chest X-ray images for COVID-19 positive cases along with Normal and Viral Pneumonia images. This COVID-19, normal and other lung infection dataset is released in stages. In the first release, 219 COVID-19, 1341 normal and 1345 viral pneumonia chest X-ray (CXR) images. In the first update, increased the COVID-19 class to 1200 CXR images. In the 2nd update, increased the database to 3616 COVID-19 positive cases along with 10,192 Normal, 6012 Lung Opacity (Non-COVID lung infection) and 1345 Viral Pneumonia images. researchers will continue to update this database as soon as they have new x-ray images for COVID-19 pneumonia patients.

- **COVID-19 data**:

COVID data are collected from different publicly accessible dataset, online sources and published papers.

- ✓ 2473 CXR images are collected from padchest dataset [35].
- ✓ 183 CXR images from a Germany medical school [36].
- ✓ 559 CXR image from SIRM, Github, Kaggle & Tweeter [37,38,39,40]
- ✓ 400 CXR images from another Github source [41].

- **Normal images**:
  - ✓ 10192 Normal data are collected from three different dataset.
  - ✓ 8851 RSNA [42]
  - ✓ 1341 Kaggle [43]
- **Viral Pneumonia images:**
  - ✓ 1345 Viral Pneumonia data are collected from the Chest X-Ray Images (pneumonia) database [43]

3 Methodology, Challenges & Results

3.1 Methodology

The study focuses on three different aspects, Comparison between 2 machine-learning methods and choose the best one, secondly, use the best method and create an application with GUI to make the prediction and finally using the described morphology operation to do the lung segmentation.

3.1.1 COVID classification & prediction

As briefly described in chapter two the whole methodology of the study is based on CNN with the use of pre-defined VGG16 model and connect the feature vectors which is the output of the model to the machine learning algorithms in order to train the model to make it able to first classify between the three different clusters or sets and then save the best model with the chosen algorithm, hence, using the saved model with the trainable parameters to make a prediction on images outside our database which will assist the radiologist to diagnose patient faster and reduce the workload and time consumed for clinical diagnoses.

The *figure 3-1* summarize the study proposed methodology for the application.

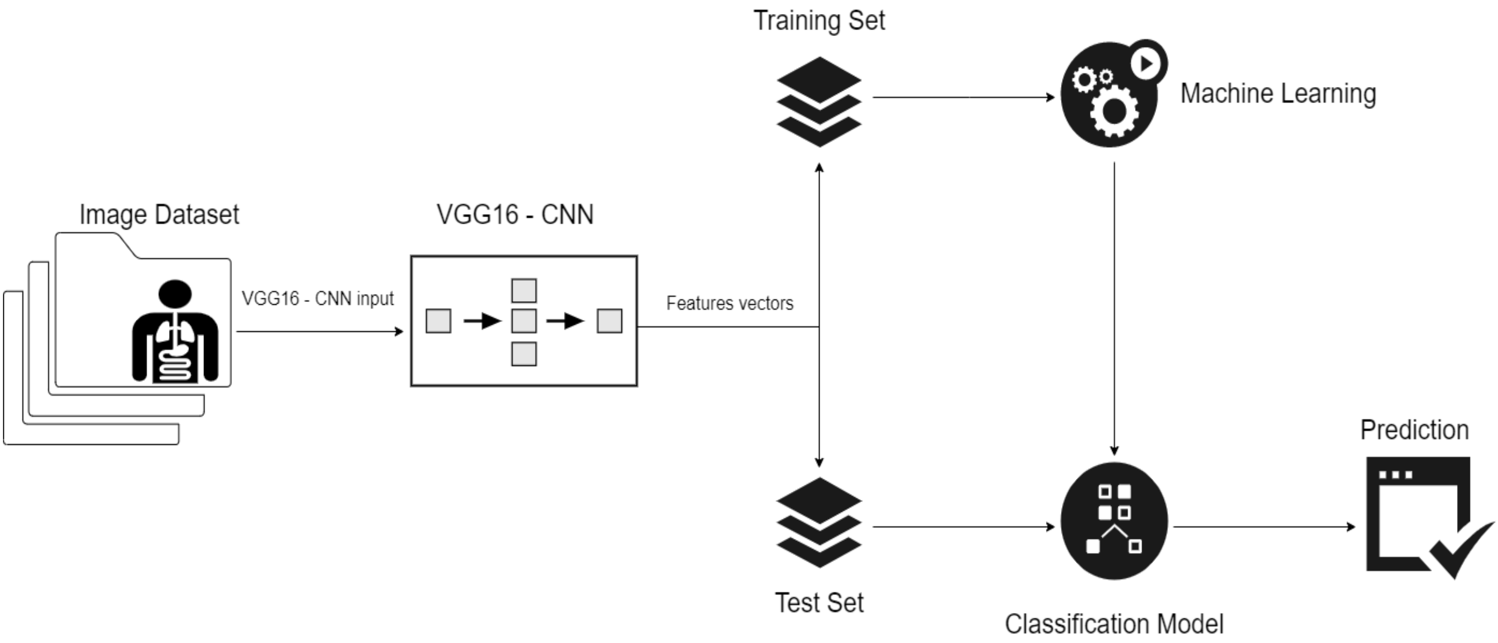


FIGURE 3-1 PROPOSED METHODOLOGY

Below, the detailed steps applied to create the proposed model and application are presented:

1. At the beginning, we started by simple advance image processing procedures on the collected database, making the data suitable for the need of the application and model proposed, first defining size of the images in order to resize the images and unify them so all of the images will be in the same size before use them as an input for the VGG16 model. After defining the size of the images and loading the libraries we talked about in chapter 2, we start by looping in each directory of the database and label the images and break the images into 3 sub categories or clusters (Normal, COVID, Viral) since we rely on using supervised machine-learning algorithms which means our database has to be labeled, after retrieving and loading the images from the directories, we save the data in numpy array then looping in each item of the array and do resizing and converting the images from BGR into RGB since VGG16 model requires the input images to be in RGB format.
2. Once the database is ready, we split the data into training and validation or test sets. The data was randomly divide 70% of the data considered as training set and the rest 30% is validation set. After splitting the data, we followed and used **Label Encoding** using Sklearn; Sklearn provides a very efficient tool for encoding the levels of categorical features into numeric values. *LabelEncoder* encode labels with a value between 0 and n_classes-1 where n is the number of distinct labels. If a label repeats, it assigns the same value to as assigned earlier. The final stage of this step is to normalize the pixel values in each image by dividing 255 to all X or input data since we encoded the output Y.
3. Now, the data is ready to be used and the in this stage we create the pre-defined VGG16 and hyper tune the model with the parameter suitable for our purpose (Tuning the model by converting the VGG16 from classification into feature extraction which was the first step and the second step is using the grid search in SVC with different gamma and c parameters which will be discussed later on) , referring to chapter 2, it was mentioned that VGG16 is a model built with 16 convolutional layers for the purpose of classification and in our study we have excluded the last 4 layers in order for us to use the model as a feature extractor.

   From the input layer to the last max pooling layer (labeled by 7 x 7 x 512) is regarded as **feature extraction part** of the model, while the rest of the network is regarded as **classification part** of the model [44]. Refer to *figure 3-2 VGG16 layers*

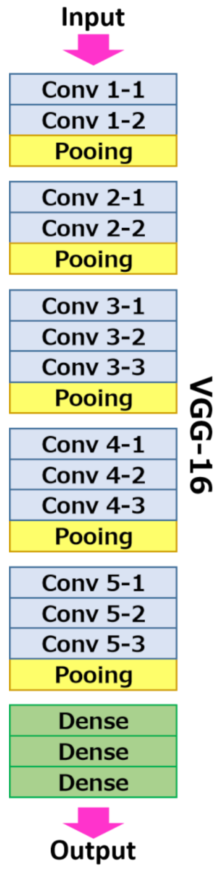


FIGURE 3-2 VGG16 LAYERS

FC (Fully Connected) layers are used to detect specific global configurations of the features detected by the lower layers in the net. Classification happens after feature extraction we need to classify the data into various classes, this can be done using a fully connected (FC) neural network but as mentioned above we are not using the model as a classifier hence deleting or removing the last 4 FC layers keeps our model as a feature extractor tool instead of a classifier tool.

The use of ***max pool layer*** is similar to convolution layer, but instead of doing convolution operation, we are selecting the max values in the receptive fields of the input, saving the indices and then producing a summarized output volume.

4. At this stage, we accomplished so far the preparation of the dataset as well as building our feature extractor model using VGG16 and the output of the model will be feature vector. Using the output as an input for our machine learning algorithms in order to make our classification and save the trained model to use it late as a prediction for unlabeled data or pictures.

   The machine learning methods or algorithms being used are KNN and SVM.

For our KNN model, we used **OneVsRestClassifier** (One-vs-the-rest (OvR) multiclass strategy). Our model here is a multiclass classifier and we are obligated to use this in order to achieve the classification we need for KNN.

Also known as, one-vs-all, this strategy consists in fitting one classifier per class. For each classifier, the class is fitted against all the other classes. In addition to its computational efficiency, (only n_classes classifiers are needed); one advantage of this approach is its interpretability. Since each class is represented by one and one classifier only, it is possible to gain knowledge about the class by inspecting its corresponding classifier. This is the most commonly used strategy for multiclass classification and is a fair default choice. [45]

For SVM model, we used **SVC (Support vector classifier)** which is the multi-class classifier for more multiple clusters or subsets in the dataset; also, we consider using two kernels for the SVM (Linear and Radial Basis Function RBF). We wanted to study the difference and gather the best model with the highest accuracy and F1 score.

While dealing with RBF kernel in SVM we have to consider two important parameters and we have to tune these parameters in order to achieve the required goal and purpose of the study.

Intuitively, Gamma is the parameter of a Gaussian Kernel (to handle non-linear classification), the gamma parameter defines how far the influence of a single training example reaches, with low values meaning ‘far’ and high values meaning ‘close’. The gamma parameters can be seen as the inverse of the radius of influence of samples selected by the model as support vectors. A small gamma will give you low bias and high variance while a large gamma will give you higher bias and low variance.

The C parameter trades off correct classification of training examples against maximization of the decision function’s margin. For larger values of C, a smaller margin will be accepted if the decision function is better at classifying all training points correctly. A lower C will encourage a larger margin, therefore a simpler decision function, at the cost of training accuracy. In other words, C behaves as a regularization parameter in the SVM.

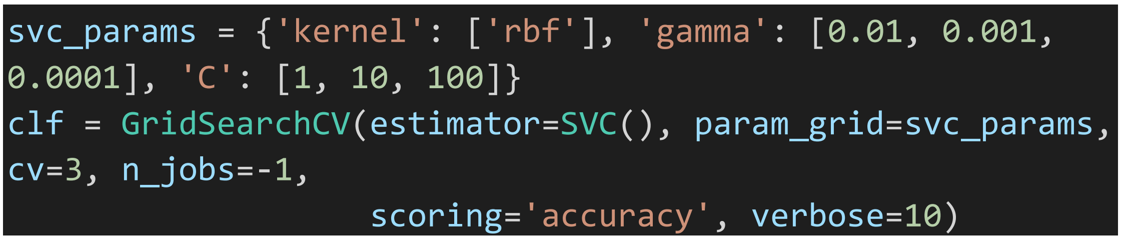
To summarize and recap Gamma and C hyper parameters; C is a hypermeter which is set before the training model and used to control error and Gamma is also a hypermeter which is set before the training model and used to give curvature weight of the decision boundary.

Instead of using trial and error method to find the best values for gamma and C respectively, we used ***GridSearchCV*** method from Sklearn, where an Exhaustive search over specified parameter values for an estimator.

GridSearchCV implements a “fit” and a “score” method. It also implements “score_samples”, “predict”, “predict_proba”, “decision_function”, “transform” and “inverse_transform” if they are implemented in the estimator used.

The parameters of the estimator used to apply these methods are optimized by cross-validated grid-search over a parameter grid. [46]

For this study, we have chosen the following parameters for the grid:



```
svc_params = {'kernel': ['rbf'], 'gamma': [0.01, 0.001,
0.0001], 'C': [1, 10, 100]}
clf = GridSearchCV(estimator=SVC(), param_grid=svc_params,
cv=3, n_jobs=-1,
                   scoring='accuracy', verbose=10)
```

For the above text box, we can see we chose three values for gamma and C respectively and using the grid search in order to find the best fit in term of scoring the highest accuracy.

5. Finally, after doing all the mentioned steps above we save the model with the best or highest accuracy and use this model to make a prediction for unlabeled data later.
6. After comparing all the three models, we chose the best one and use Tinkter that is a build in library in python. The tkinter library helps us to build our GUI; here we only considered to use the model SVM with RBF kernel since it has the highest scores compared to the other two.

3.1.2 Lung Segmentation

As illustrated in chapter 2, the core basis of lung segmentation was depending on the knowledge of advanced image processing methods and morphological operations.

The *figure 3-3* represent the workflow and the methodology to achieve the segmentation.

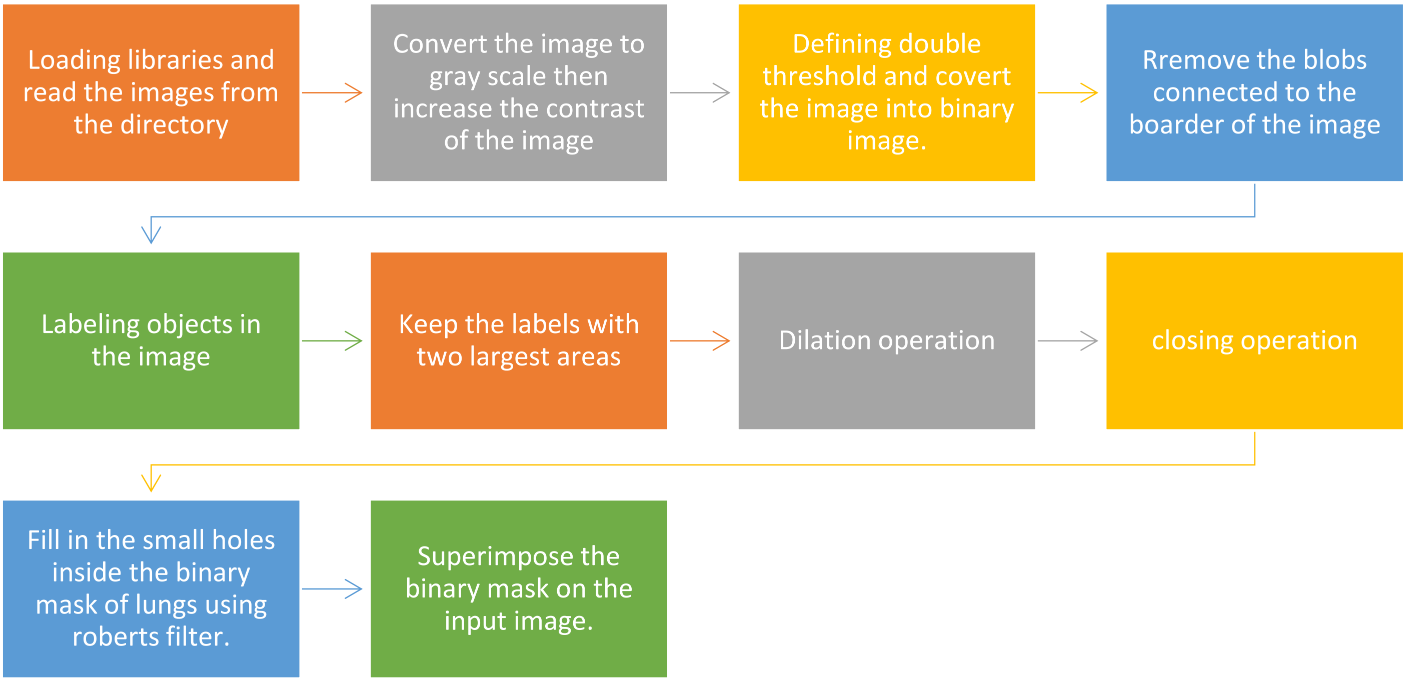

Loading libraries and read the images from the directory
Convert the image to gray scale then increase the contrast of the image
Defining double threshold and covert the image into binary image.
Rremove the blobs connected to the boarder of the image
Labeling objects in the image
Keep the labels with two largest areas
Dilation operation
closing operation
Fill in the small holes inside the binary mask of lungs using roberts filter.
Superimpose the binary mask on the input image.

FIGURE 3-3 SEGMENTATION WORKFLOW

3.2 Challenges

- First challenge we faced was the lack of information and the shortage of medical images for the study but after a very deep search and the help of the open source project we were able to find the database that include the images of X-RAY and those researches and studies helped our aim and goal to be achieved.

- At the beginning of the project we started by 2 clusters and we found out it is impractical because some lungs diseases have a similar effect on the lungs like COVID and to distinguish that after will be harder to achieve, hence, to mitigate the issue we decided to go for multi-clusters classification project to make our objective of the project more useful.

- Dealing with a huge amount of data and realizing the nonlinear issue (non-linear relationship between the dependent and independent variables. It is used in place when the data shows a curvy trend) of the data makes a huge difference in our study, it took some effort and consideration to use the correct machine leaning methods, and we found the best choice is to use KNN and SVM since these methods support nonlinear and multi-clusters classification issues.

- We started the study with the idea of building the feature extractor model from scratch and we faced overfitting issues, tuning, and accuracy issues while detecting the special features in the images, therefore, we started solving these issues one by one and after a while we saw we are implementing the VGG16 model from scratch because we were following the same footsteps, hence, we decided to use the pre-defined model of VGG16 since it is optimized and helps our aim of the project.

- The main issue we face for the lung segmentation is the images, some of the images are rotated and the other thing to be considered the size of the lungs and the quality of X-RAY images. We tried to solve these issues and our script work for most of the images but it can be enhanced and that will be something to look forward for future studies.

3.3 Results

3.3.1 Results of Classification & Prediction

The applied measures belong to the module ***sklearn.metrics*** and were implemented in python. The metrics of precision (PR), sensitivity (SE), specificity (SP), F1 scores and their respective macro-averages, use the values of the variables in the confusion matrix of each model. *Figure 3-4* presents an example of the confusion matrix for ***Binary Classification***.

The values of the variables true positives (*TP*) and true negatives (*TN*), quantify the hits. Errors are quantified by the values of the variables false positives (*FP*) and false negatives (*FN*). The metrics are defined in equations. PR calculates the probability of patients with positive results, actually having the disease. The SE calculates the probability of a positive outcome of the disease. SP, on the other hand, calculates the probability of negative results in patients without the disease. The F1 score combines the values of precision and sensitivity; it balances the relative importance of each metric. Finally, the macro averages of the metrics calculate their respective averages across classes. Thus, Equations below define the macro precision, macro sensitivity (macro recall), macro specificity, and macro F1 score, respectively.

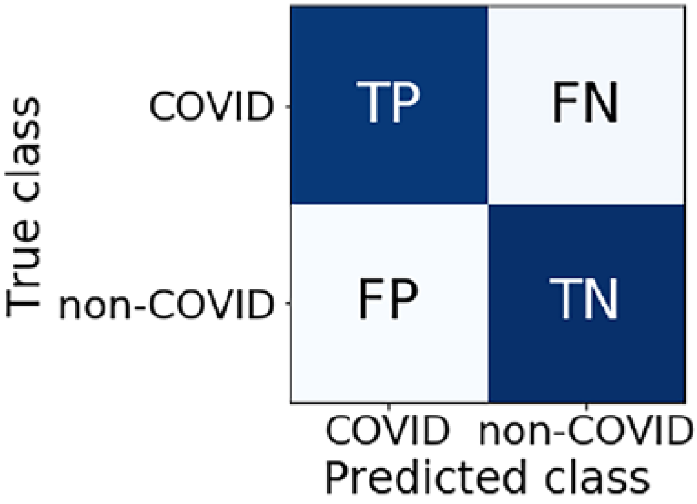


FIGURE 3-4 CONFUSION MATRIX EXAMPLE

$$PR = \frac{TP}{TP + FP}$$

$$SE = \frac{TP}{TP + FN}$$

$$SP = \frac{TN}{TN + FP}$$

$$F1 = \frac{2TP}{2TP + FP + FN}$$

$$PR(Macro) = \frac{1}{C}\sum_{i=1}^{C}\frac{TP_i}{TP_i + FP_i}$$

$$SE(Macro) = \frac{1}{C}\sum_{i=1}^{C}\frac{TP_i}{TP_i + FN_i}$$

$$SP(Macro) = \frac{1}{C}\sum_{i=1}^{C}\frac{TN_i}{TN_i + FP_i}$$

$$F1(Macro) = \frac{1}{C}\sum_{i=1}^{C}\frac{2TP_i}{2TP_i + FP_i + FN_i}$$

For ***Multi-Classification***, the approach to determine TP, TN, FP, and FN becomes different with small changes compared to the binary classification.

To determine the values the approach will be calculating each one of the above values independently to each class and the best way to visualize the confusion matrix and to understand the methodology of it to relate to *figure 3-5*.

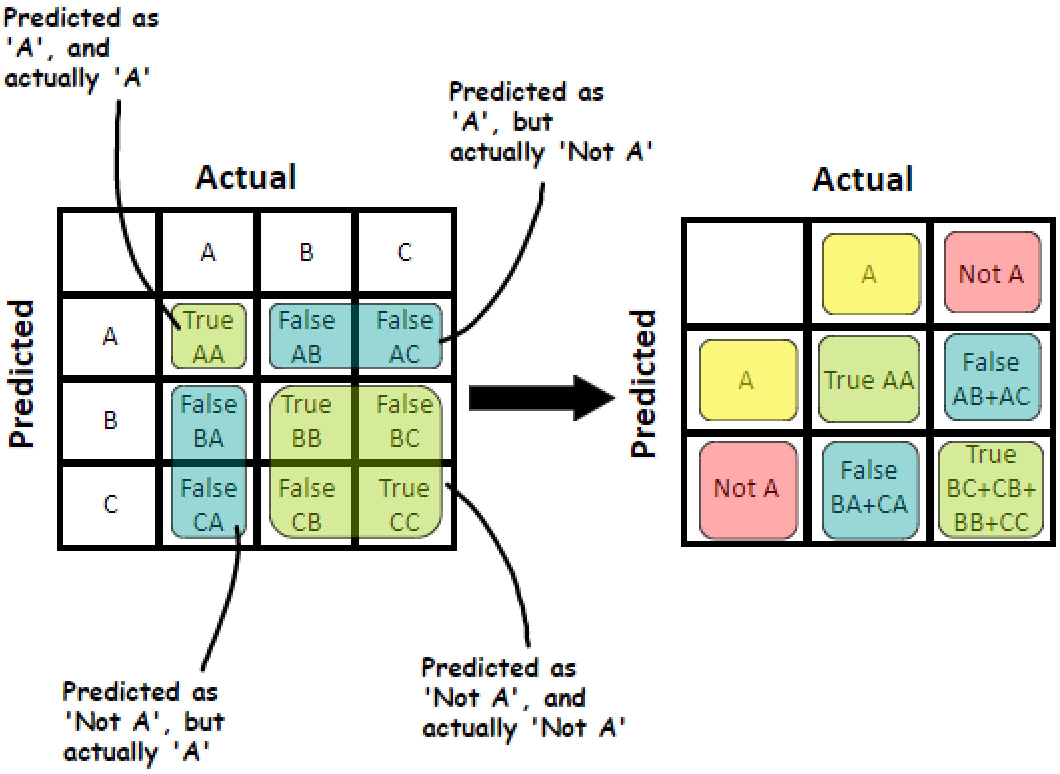


FIGURE 3-5 MULTI CLASSIFICATION

From above, we conclude that we have three TP values, each one represent the corresponding each class assigned to it and it is easy to determine the TP for each class but the tricky ones are TN, FP, and FN.

For example to determine the values for class A, the TP will be *True AA*, for FP, which is the value that represent the predicted, is true or positive but the actual value is negative and from this we conclude FP for class A will be the summation of *False AB* and *False AC*, and the same concept applies to the other classes B and C.

$$FP_A = Flase_{AB} + \ Flase_{AC}$$

The TN for a particular class is calculated by taking the sum of the values in every row and column except the row and column of the class we are trying to find the **TN** for.

We omit the row and columns belonging to the **A** class and sum the variables that are left, which are the rows and columns of the other classes (**B** and **C**).

$$TN_A = True_{BB} + Flase_{BC} + True_{CC} + Flase_{CB}$$

For FN, which is the value that represent the predicted, is Negative but the actual value is Positive and from this we conclude FN for class A will be the summation of *False BA* and *False CA*.

$$FP_A = Flase_{BA} + Flase_{CA}$$

Referring back to the previous equations, calculating the accuracy, precision and F1 scores will be calculated for a specific class except for the accuracy since the accuracy is calculated as the ratio of the number of correct classifications to the total number of classifications. From the confusion matrix, the correct classifications are the **TP** for each class and the total number of classifications is the sum of every value in the confusion matrix, including the **TP**.

Precision is a multi-class confusion matrix is the measure of the accuracy relative to the prediction of a specific class. It is calculated as the ratio of the **TP** of the specific class to the sum of its **TP** and **FP** for that class.

$$PR_A = \frac{TP_A}{TP_A + FP_A}$$

The same method is applied to each class. In addition, we can calculate SE, SP, and F1 using the same equations as before but we have to keep in mind that these calculations for multi-classifications have to be done per class or for a specific class.

From *Table 3-1* and *figures 3-6--8*, we can see the side comparison of the three methods KNN, SVM (Linear), and SVM (RBF).

TABLE 3-1 ML MEASURES COMPARISON

	KNN				SVM (Linear)				SVM (RBF)			
	precision	Recall	F1	Support	precision	Recall	F1	Support	precision	Recall	F1	Support
COVID	0.93	0.96	0.94	1094	0.98	0.98	0.98	1094	0.98	0.98	0.98	1094
Normal	0.94	0.9	0.92	933	0.97	0.97	0.97	933	0.97	0.97	0.97	933
Viral Pneumonia	0.95	0.94	0.95	418	0.97	0.97	0.97	418	0.98	0.97	0.97	418
accuracy			0.94	2445			0.97	2445			0.98	2445
macro avg	0.94	0.94	0.94	2445	0.97	0.97	0.97	2445	0.98	0.98	0.98	2445
weighted avg	0.94	0.94	0.94	2445	0.97	0.97	0.97	2445	0.98	0.98	0.98	2445

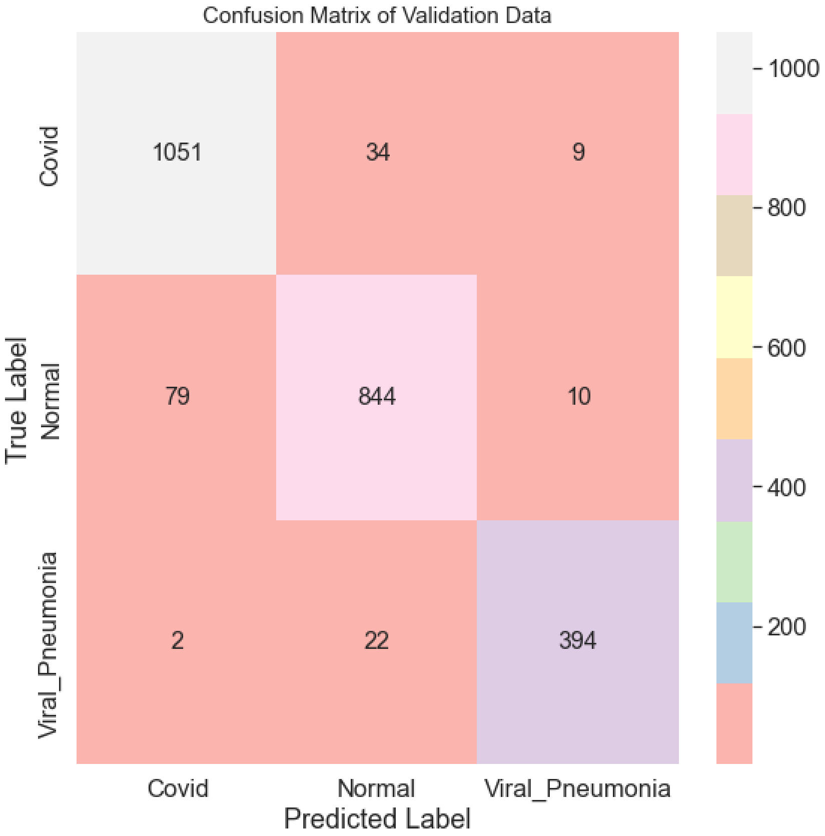


FIGURE 3-6 KNN CM

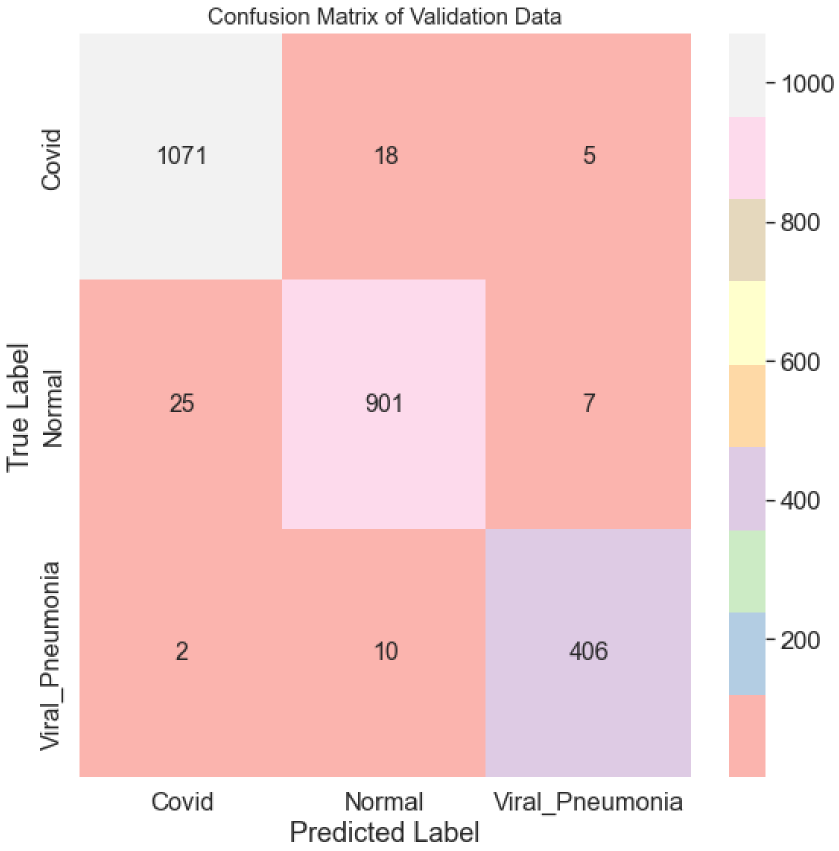


FIGURE 3-7 3 SVM LINEAR CM

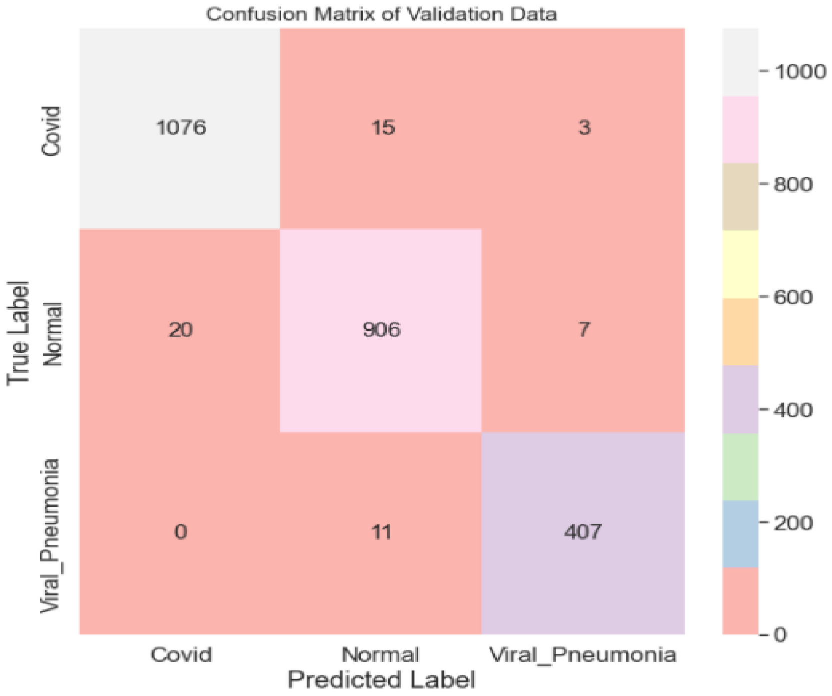


FIGURE 3-8 SVM (RBF) CM

From the previous table and figures, we can see that SVM with RBF kernel scored the highest accuracy between the two methods, accuracy results respectively from the highest to the lowest 98%, 97%, and 94%. The lowest result was for KNN algorithm. On the other hand, using only VGG16 as a classification model will achieve accuracy of 91%, which is lower than the result we achieved.

3.3.2 Lung Segmentation Results

The *figure 3-8* represent the output of each stage in the methodology described earlier in the beginning of this section.

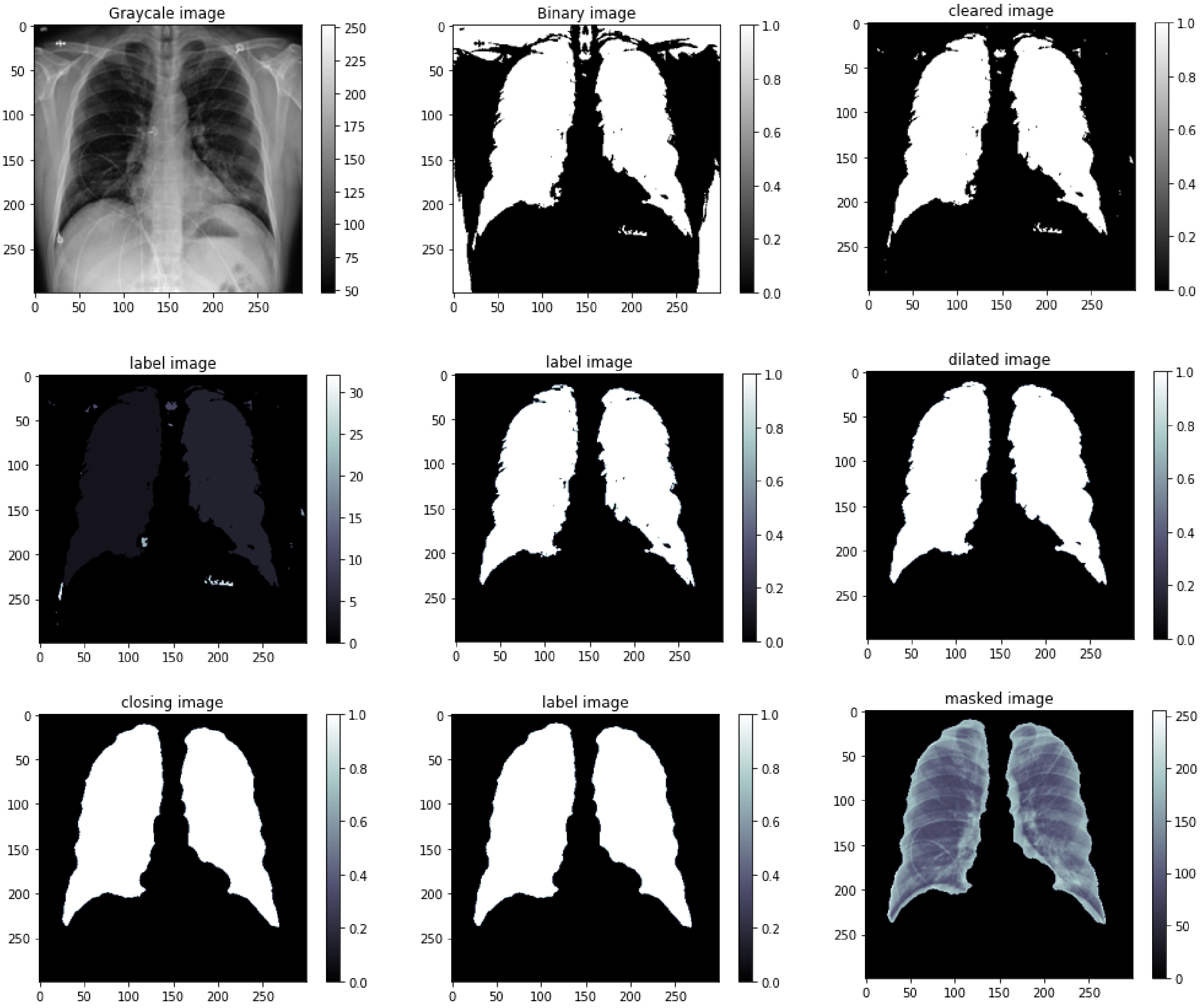


FIGURE 3-9 SEGMENTATION PROCESS AND RESULT

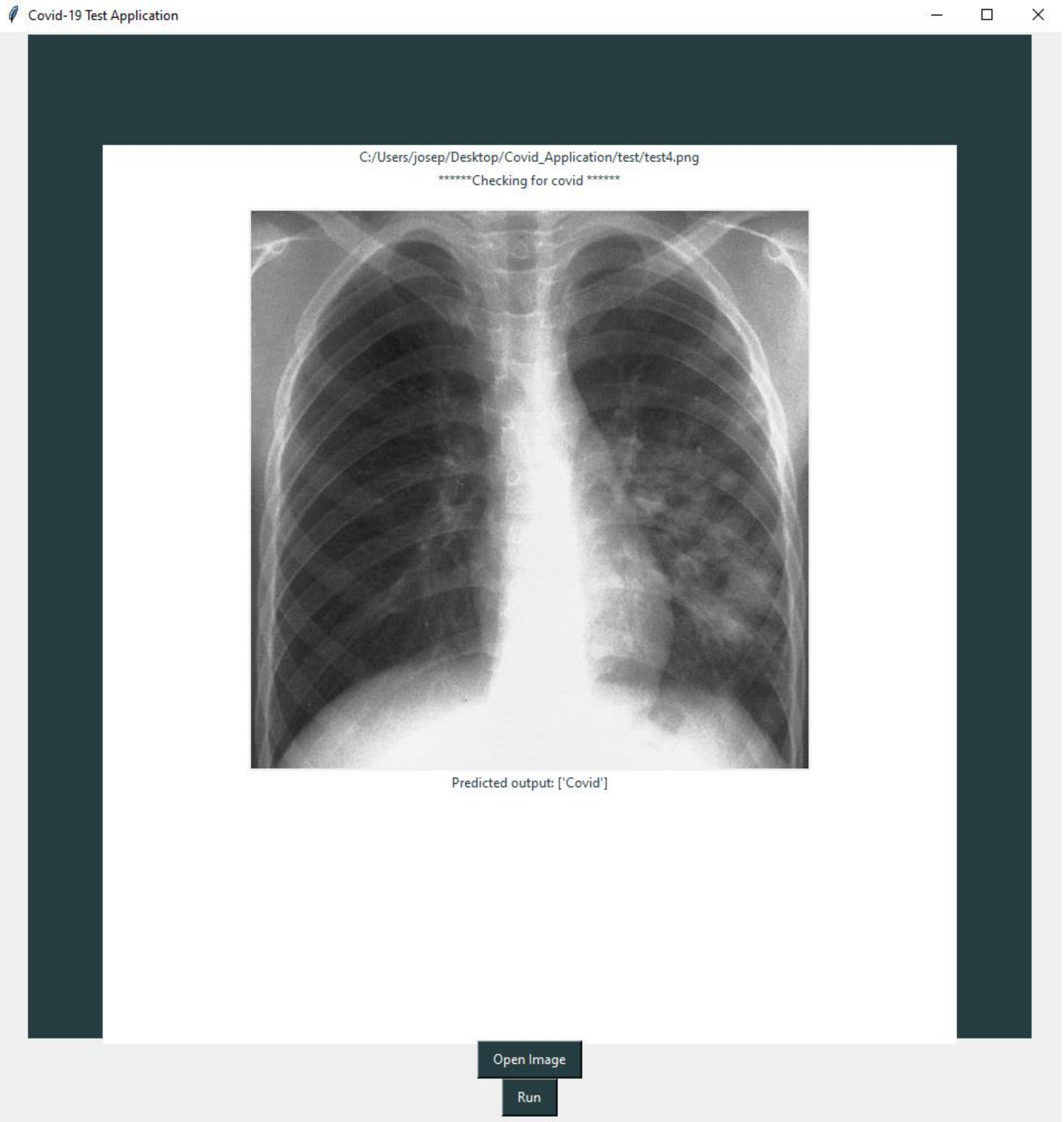
3.3.3 COVID-19 Application

In this section, we will provide the application that we built based on the model SVC RBF and VGG16 since this model scored the highest accuracy and F1 scores.

The GUI application was created using tkinter the built-in library in python.

The application has three python files, preparing dataset.py, model.py, and main.py. For more information about how to use the application, check appendix A.

The figure shows an example of the output of the application.



C:/Users/josep/Desktop/Covid_Application/test/test4.png

******Checking for covid ******

Predicted output: ['Covid']

FIGURE 3-10 GUI OUTPUT EXAMPLE

4 Conclusion

This study implemented pre-trained CNNs to extract the vectors of global features from X-RAY images. The vectors were used to design classification and a prediction models for COVID, non- COVID-19, and viral Pneumonia. The Algorithms of CNNs applied were VGG16. The classification algorithms applied were *K Nearest Neighbor Classifier* (KNN), *Support Vector Machine* (SVM).

In the experimental evaluation, the performances of each model were compared in an accuracy analysis using the Sklearn measurement tools. These measures identified the best classification models, and chose this model to build an application and be able to predict images from outside our dataset or database. Thus, the KNN classifier in conjunction with VGG16 achieved an accuracy score of 94% with 0.94 in both F1 and recall macro averages. The SVC linear classifier obtained a higher performance with VGG16 baseline, compared to KNN where SVC linear classifier achieved an accuracy of 97% with 0.97 both F1 and recall macro averages. The SVC RBF with data extracted by VGG16 was the model that obtained the highest average performances: 98% in the accuracy, 0.98 in the sensitivity (recall) macro, 0.98 in the specificity macro, and 0.98 in the F1 score macro. In this scenario, the SVC RBF model with VGG16 is the best to extract features and classifier, as it was the only one that achieved the best performances in each classification model compared to KNN and SVC linear.

Therefore, the use of ML and CNN on transfer learning to extract features from X-RAY images has an enormous impact in the medical application and as it was shown in this study, the model was able to identify and predict COVID, non- COVID-19, and viral Pneumonia in a validation or test set.

Finally, this study also shows the importance of advance image processing in the medical field, where the use of these methods can increase the efficiency of the work done and it can analyze the problem faster and we can achieve and progress in the medical field with such methods, as an example, this study shows that advance image processing methods especially morphological operations were able to make a separation of the lung and segment the right side from the left side.

In future works, we can include another predefined CNN models and compare these model in order to achieve a higher accuracy results, also we think about include parallel programming to speed up the process since dealing with a huge amount of images can be very exhaustive on the hardware and operating system. Finally, we should always think about these methods with the medical field because it can solve problematic issues faster and it is a time saver.

5 References

1. Wang, D. *et al.* Clinical characteristics of 138 hospitalized patients with 2019 novel coronavirus-infected pneumonia in Wuhan, China. *JAMA* https://doi.org/10.1001/jama.2020.1585 (2020).

2. Aboughdir, M. *et al.* Prognostic value of cardiovascular biomarkers in COVID-19: A review. *Viruses* https://doi.org/10.3390/v12050527 (2020).

3. Acharya, A. *et al.* SARS-CoV-2 infection leads to neurological dysfunction. *J. Neuroimmune Pharmacol.* https://doi.org/10.1007/s11481- 020- 09924-9 (2020).

4. Kiran, G. *et al.* In silico computational screening of Kabasura Kudineer—Official Siddha Formulation and JACOM against SARSCoV-2 Spike protein. *J. Ayurveda Integr. Med.* https://doi.org/ 10. 1016/j. jaim. 2020. 05. 009 (2020).

5. Ackermann, M. *et al.* Pulmonary vascular endothelialitis, thrombosis, and angiogenesis in Covid-19. *New Engl. J. Med.* https://doi.org/ 10. 1056/ NEJMo a2015 432 (2020).

6. Cao, Y. *et al.* Potent neutralizing antibodies against SARS-CoV-2 identified by high-throughput single-cell sequencing of convalescent patients' B cells. *Cell* https://doi.org/ 10. 1016/j. cell. 2020. 05. 025 (2020).

7. Addeo, A. *et al.* COVID-19 and lung cancer: risks, mechanisms and treatment interactions. *J. Immunother. Cancer* https://doi.org/10. 1136/ jitc- 2020- 000892 (2020).

8. Agarwal, A. *et al.* Guidance for building a dedicated health facility to contain the spread of the 2019 novel coronavirus outbreak. *Indian J. Med. Res.* **151**(2), 177–183. https://doi.org/ 10. 4103/ ijmr. IJMR_ 518_ 20 (2020).

9. D. J. Cennimo, "Coronavirus disease 2019 (COVID-19) clinical presentation," vol. 8, pp. 101489–101499, 2020, https://emedicine.medscape.com/article/2500114-clinical#b2, 2020. Online.

10. World Health Organization (WHO).

11. "Coronavirus Disease 2019 (COVID-19) Symptoms". Centers for Disease Control and Prevention. United States.

12. "Coronavirus (COVID-19) Mortality Rate". Www.worldometers.info. 5 March 2020. Retrieved 23 March 2020.

13. Hui, D., et al. (2020). "The continuing 2019-nCoV epidemic threat of novel coronaviruses to global health—The latest 2019 novel coronavirus outbreak in Wuhan, China". Int J Infect Dis. 91: 264–66. doi:10.1016/j.ijid.2020.01.009. PMID 31953166.

14. "Anosmia, Hyposmia, and Dysgeusia Symptoms of Coronavirus Disease". American Academy of Otolaryngology-Head and neck surgery. 22 March 2020.

15. "New coronavirus stable for hours on surfaces". National Institutes of Health. 17 March 2020. Retrieved 23 March 2020.

16. "Symptoms of Novel Coronavirus (2019-nCoV)". www.cdc.gov. Retrieved 11 February 2020.

17. Velavan, T. ; Meyer, C. (2020). "The COVID-19 epidemic". Tropical Medicine & International Health. n/a (n/a): 278–80. doi:10.1111/tmi.13383.

18. Jin YH, Cai L., et al. (February 2020). "A rapid advice guideline for the diagnosis and treatment of 2019 novel coronavirus (2019-nCoV) infected pneumonia (standard version)". Military Medical Research. 7 (1): 4. doi:10.1186/s40779-020-0233-6. PMC 7003341. PMID 32029004.

19. "Guidance on social distancing for everyone in the UK". GOV.UK. Retrieved 25 March 2020.

20. www.fda.gov

21. Different COVID-19 Vaccines | CDC

22. Aijaz Ahmad Reshi ,1 Furqan Rustam ,2 Arif Mehmood ,3 Abdulaziz Alhossan,4,5 Ziyad Alrabiah,4 Ajaz Ahmad ,4 Hessa Alsuwailem,4 and Gyu Sang Choi 6, "AnEfficientCNNModel forCOVID-19Disease Detection Based on X-Ray Image Classification"

23. Amir Rehman, Muhammad Azhar Iqbal, Huanlai Xing * and Irfan Ahmed " COVID-19 Detection Empowered with Machine Learning and Deep Learning Techniques: A Systematic Review"

24. Role of Python in Artificial Intelligence (AI) | Python Machine Learning (cuelogic.com)

25. SKLearn | Scikit-Learn In Python | SciKit Learn Tutorial (analyticsvidhya.com)

26. Your First Deep Learning Project in Python with Keras Step-By-Step (machinelearningmastery.com)

27. Introduction to the Python Deep Learning Library TensorFlow (machinelearningmastery.com)

28. What is convolutional neural network? - Definition from WhatIs.com (techtarget.com)

29. What is VGG16? — Introduction to VGG16 | by Great Learning | Medium

30. Support Vector Machines (SVM) Algorithm Explained (monkeylearn.com)

31. KNN Algorithm - Finding Nearest Neighbors (tutorialspoint.com)

32. Types of Morphological Operations - MATLAB & Simulink (mathworks.com)

33. M.E.H. Chowdhury, T. Rahman, A. Khandakar, R. Mazhar, M.A. Kadir, Z.B. Mahbub, K.R. Islam, M.S. Khan, A. Iqbal, N. Al-Emadi, M.B.I. Reaz, M. T. Islam, "Can AI help in screening Viral and COVID-19 pneumonia?" IEEE Access, Vol. 8, 2020, pp. 132665 - 132676.

34. Rahman, T., Khandakar, A., Qiblawey, Y., Tahir, A., Kiranyaz, S., Kashem, S.B.A., Islam, M.T., Maadeed, S.A., Zughaier, S.M., Khan, M.S. and Chowdhury, M.E., 2020. Exploring the Effect of Image Enhancement Techniques on COVID-19 Detection using Chest X-ray Images. arXiv preprint arXiv:2012.02238.

35.https://bimcv.cipf.es/bimcv-projects/bimcv-covid19/#1590858128006-9e640421-6711

36.https://github.com/ml-workgroup/covid-19-image-repository/tree/master/png

37.https://sirm.org/category/senza-categoria/covid-19/

38. https://eurorad.org

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

40. https://figshare.com/articles/COVID-19_Chest_X-Ray_Image_Repository/12580328

41. https://github.com/armiro/COVID-CXNet

42. https://www.kaggle.com/c/rsna-pneumonia-detection-challenge/data

43. https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia

44. Extract Features, Visualize Filters and Feature Maps in VGG16 and VGG19 CNN Models | by Roland Hewage | Towards Data Science

45. sklearn.multiclass.OneVsRestClassifier — scikit-learn 1.0.2 documentation

46. sklearn.model_selection.GridSearchCV — scikit-learn 1.0.2 documentation

6 Appendix A

This appendix describe the model of the application and a simple instruction on how to use the application.

The preparing.py file is simply the file that handle the database and the data processing such as resizing and splitting the data into training and cross validation set.

Model.py is the file that build the model, first loading and training VGG16 to be a feature extractor, build our SVC RBF model and save it in a pickle file in order to load the model later on to use it to predict images outside the database. Two outputs will be generated from this file, the confusion matrix as PNG image *figure 6-1* and model parameter as text file *table 6-1*.

Main.py is the file that handle the GUI. We load the pickle file to the backend of the application, and then we ask the user to choose an image, from the backend side we resize the image and convert it to RGB and normalize the image because all of these parameters were used to build the VGG16 and SVC RBF model.

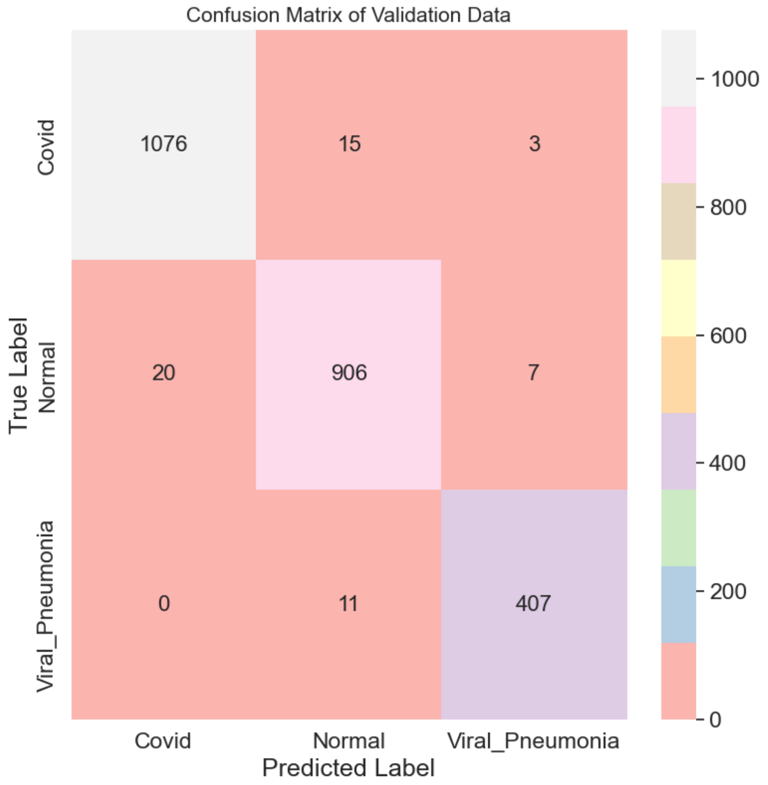


FIGURE 6-1 CM OUTPUT FROM GUI

Model: "vgg16"		
***		
Layer (type)	Output Shape	Param #
input_1 (Input Layer)	(None, 256, 256, 3)	0
block1_conv1 (Conv2D)	(None, 256, 256, 64)	1792
block1_conv2 (Conv2D)	(None, 256, 256, 64)	36928
block1_pool (MaxPooling2D)	(None, 128, 128, 64)	0
block2_conv1 (Conv2D)	(None, 128, 128, 128)	73856
block2_conv2 (Conv2D)	(None, 128, 128, 128)	147584
block2_pool (MaxPooling2D)	(None, 64, 64, 128)	0
block3_conv1 (Conv2D)	(None, 64, 64, 256)	295168
block3_conv2 (Conv2D)	(None, 64, 64, 256)	590080
block3_conv3 (Conv2D)	(None, 64, 64, 256)	590080
block3_pool (MaxPooling2D)	(None, 32, 32, 256)	0
block4_conv1 (Conv2D)	(None, 32, 32, 512)	1180160
block4_conv2 (Conv2D)	(None, 32, 32, 512)	2359808
block4_conv3 (Conv2D)	(None, 32, 32, 512)	2359808
block4_pool (MaxPooling2D)	(None, 16, 16, 512)	0
block5_conv1 (Conv2D)	(None, 16, 16, 512)	2359808
block5_conv2 (Conv2D)	(None, 16, 16, 512)	2359808
block5_conv3 (Conv2D)	(None, 16, 16, 512)	2359808
block5_pool (MaxPooling2D)	(None, 8, 8, 512)	0
***		
Total params: 14,714,688		
Trainable params: 0		
Non-trainable params: 14,714,688		

TABLE 6-1 MODEL PARAMETERS

Figures 6-2, 3 represent the steps and the output of the GUI application.

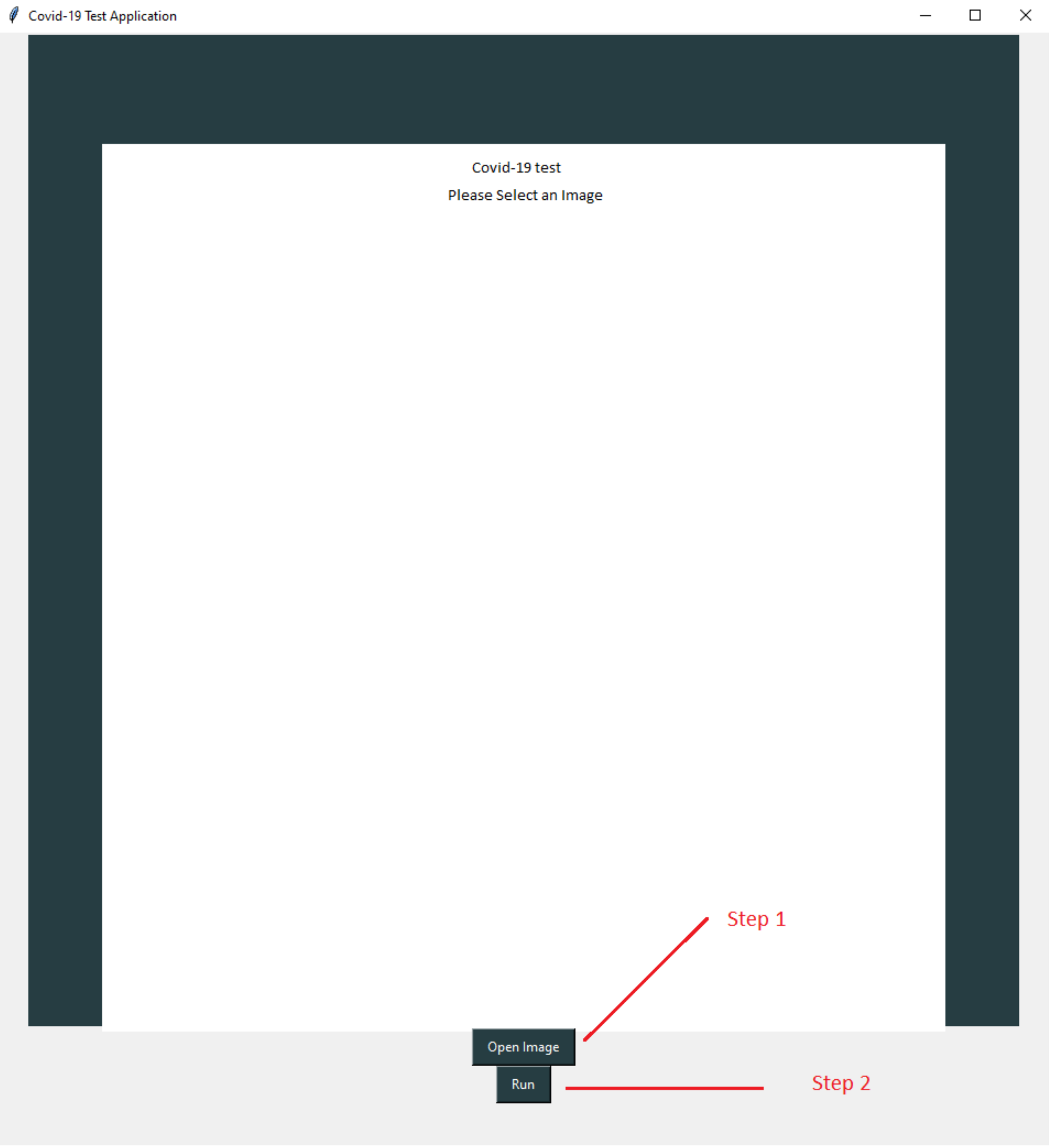
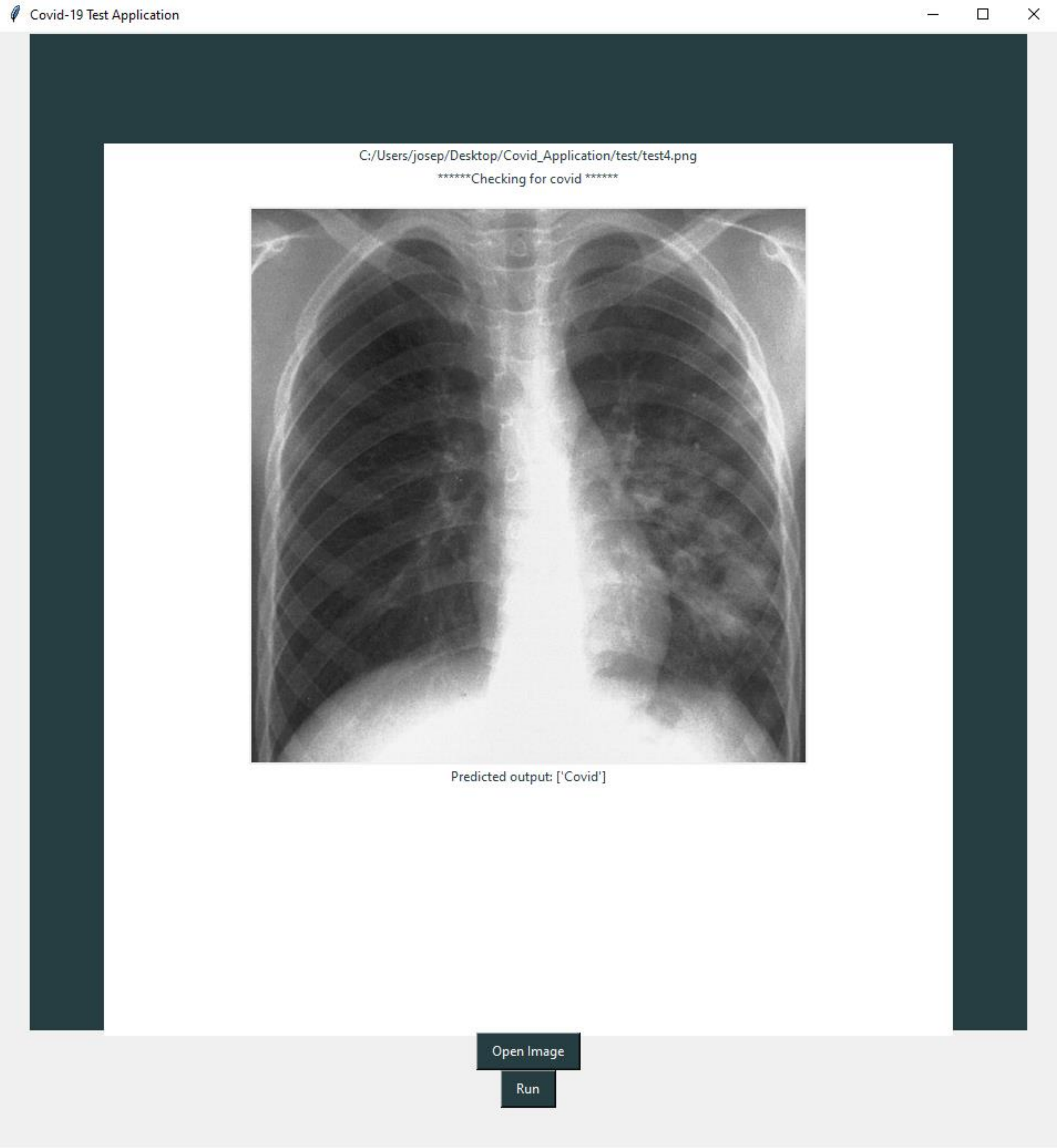


FIGURE 6-2 GUI STARTING STEPS



C:/Users/josep/Desktop/Covid_Application/test/test4.png

******Checking for covid ******

Predicted output: ['Covid']

FIGURE 6-3 GUI OUTPUT