1

A Report on Major project

**Covid-19 Detection through X-Rays**

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

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER ENGINEERING**

**OF**

**VISHWAKARMA INSTITUTE OF TECHNOLOGY**

**Savitribai Phule Pune University**

*BY*

**Raunak Rathi             (GR No. 11910379)**

**Vedant Rathi               (GR No. 11910476)**

**Shoaib Shaikh             (GR No. 11911135)**

**Devarshi Wadadkar   (GR No. 11910830)**

*UNDER THE GUIDANCE OF*

Prof. Kaushalya Thopate

DEPARTMENT OF COMPUTER ENGINEERING

BANSILAL RAMNATH AGARWAL CHARITABLE TRUST’S

VISHWAKARMA INSTITUTE OF TECHNOLOGY (An Autonomous Institute affiliated to Savitribai Phule Pune University) PUNE - 411037

2022 - 2023

2

**BANSILAL RAMNATH AGARWAL CHARITABLE TRUST’S**

**VISHWAKARMA INSTITUTE OF TECHNOLOGY (An Autonomous Institute affiliated to Savitribai Phule Pune University) PUNE – 411037**

****

**CERTIFICATE**

This is to certify that the Major Project titled **Covid-19 Detection through X-Rays** submitted by **Raunak Rathi(GR No. 11910379), Vedant Rathi (GR No. 11910476),  Shoaib Shaikh(GR No. 11911135), Devarshi Wadadkar(GR No.11910830)** is in partial fulfillment for the award of Degree of Bachelor of Technology in Computer Engineering of Vishwakarma Institute of Technology, Savitribai Phule Pune University. This project report is a record of bonafide work carried out by him under my guidance during the academic year 2022-23.

**Guide Head of Computer Department Prof. Kaushalya Thopate Prof. Dr. S. R. Shinde** Dept. of Computer Engg. Vishwakarma Institute of Technology, Pune.

**Sign of External Examiner Date**

3

Bansilal Ramnath Agarwal Charitable Trust’s

**Vishwakarma Institute Of Technology, Pune-37**

**Department Of Computer Engineering**

**PROJECT SYNOPSIS**

Group No : CS B.Tech 49

Group Members:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Roll**  **No** | **Name** | **Class** | **Contact No** | **Email-ID** |
| 5 | Raunak Rathi | B. Tech | 7028706328 | raunak.rathi19@vit.edu |
| 6 | Vedant Rathi | B. Tech | 9182075890 | vedant.rathi19@vit.edu |
| 36 | Shoaib Shaikh | B. Tech | 9307211348 | shoaib.shaikh19@vit.edu |
| 69 | Devarshi Wadadkar | B.Tech | 9571005925 | devarshi.wadadkar19@vit.edu |

Academic Year : 2022-23

Project Title : Covid-19 Detection through X-Rays

Project Area : Deep Learning,Web Technology

Sponsor Company : -

Company Address : -

Internal Guide : Prof. Kaushalya Thopate

Name of the External Guide:

Contact No:

**Signature of Internal Guide**

4

**ACKNOWLEDGEMENT**

It gives us great satisfaction to be able to present this project on Covid-19 Detection through X-Rays. We would like to express our deep gratitude towards our project guide Prof.Kaushalya Thopate, for all the guidance and the co-operation, without whom this project would have been an uphill task. We would also like to thank Prof. Dr. Sandeep Shinde for his invaluable guidance during the making of the project.

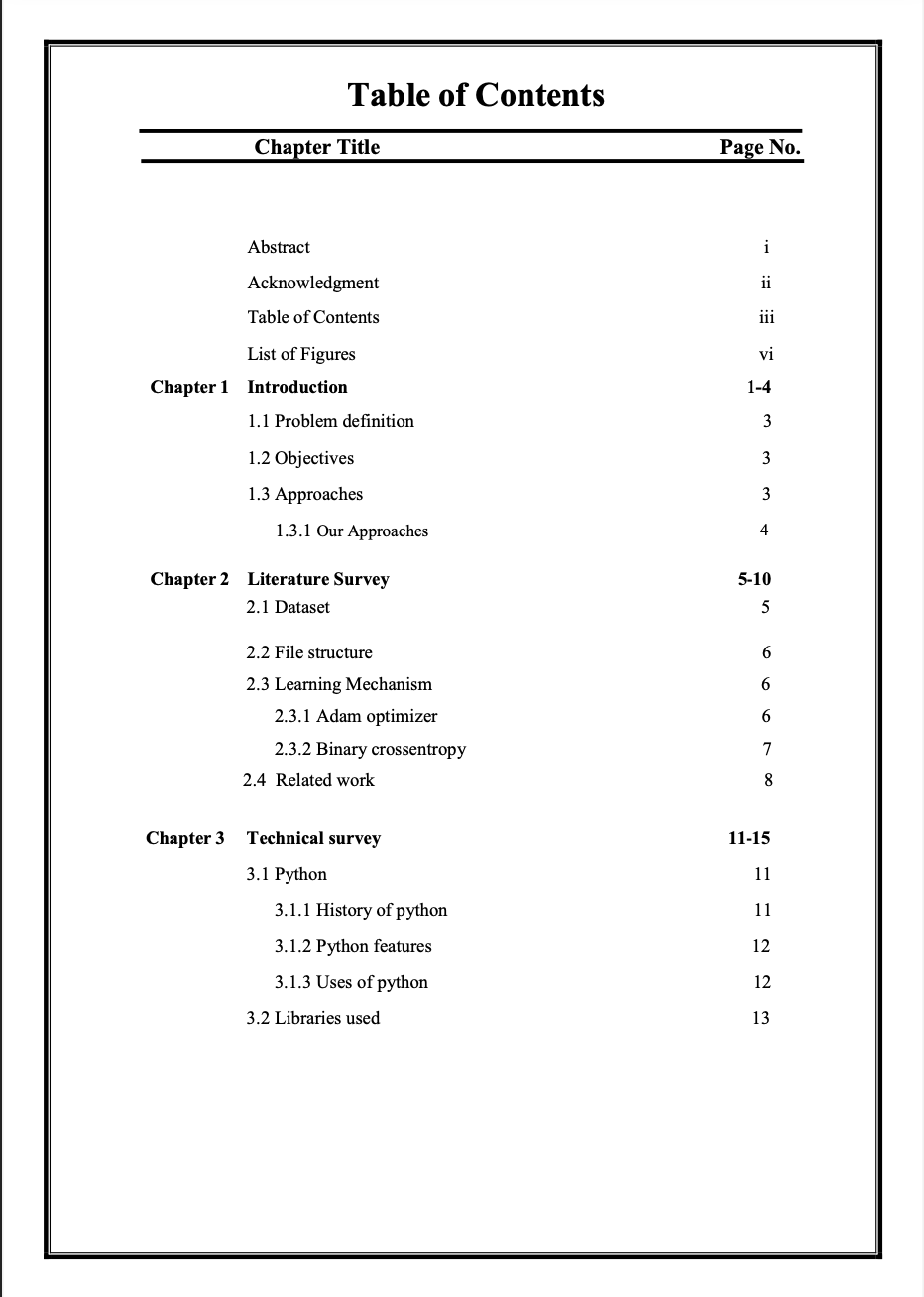
Raunak Rathi

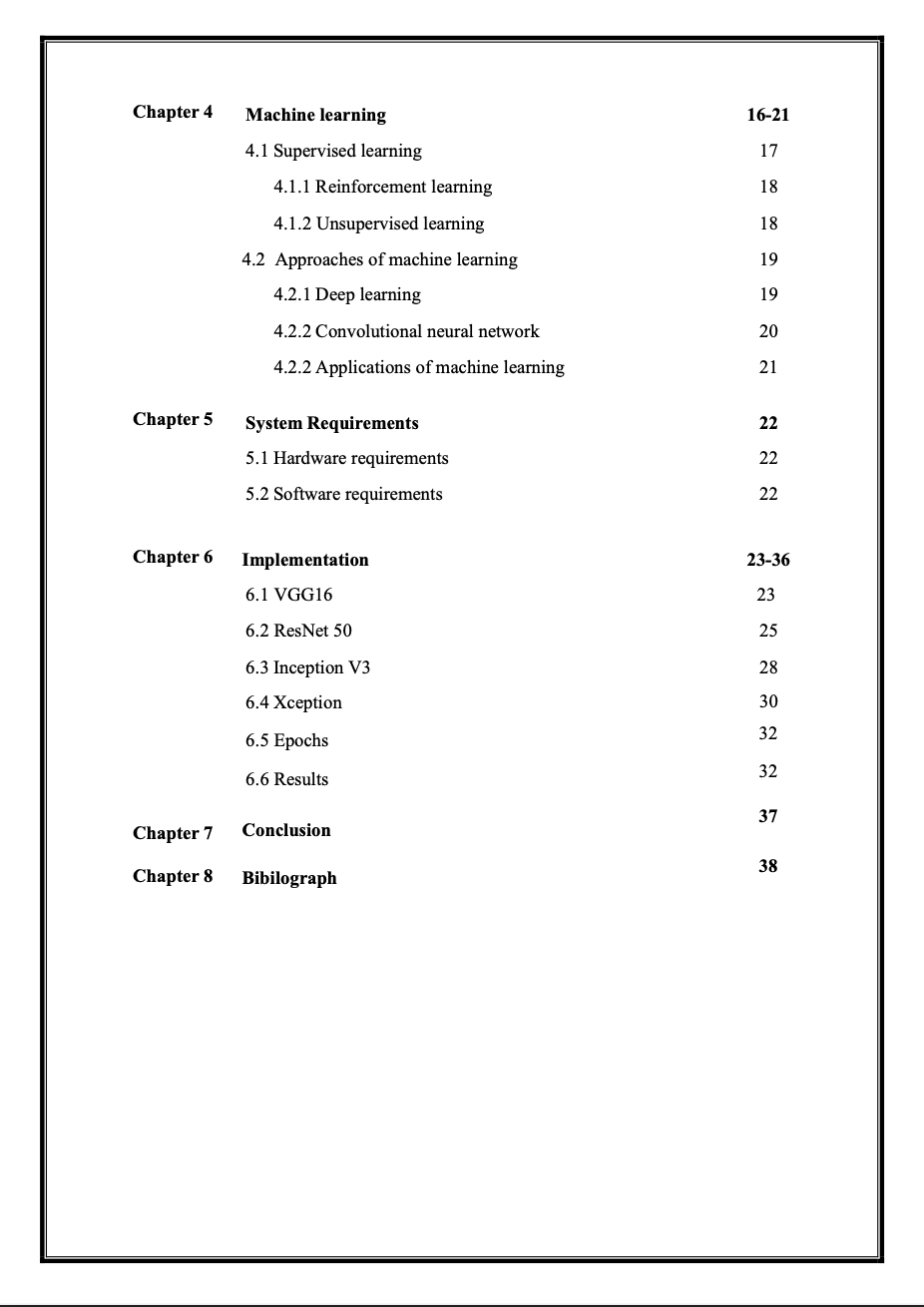
Vedant Rathi

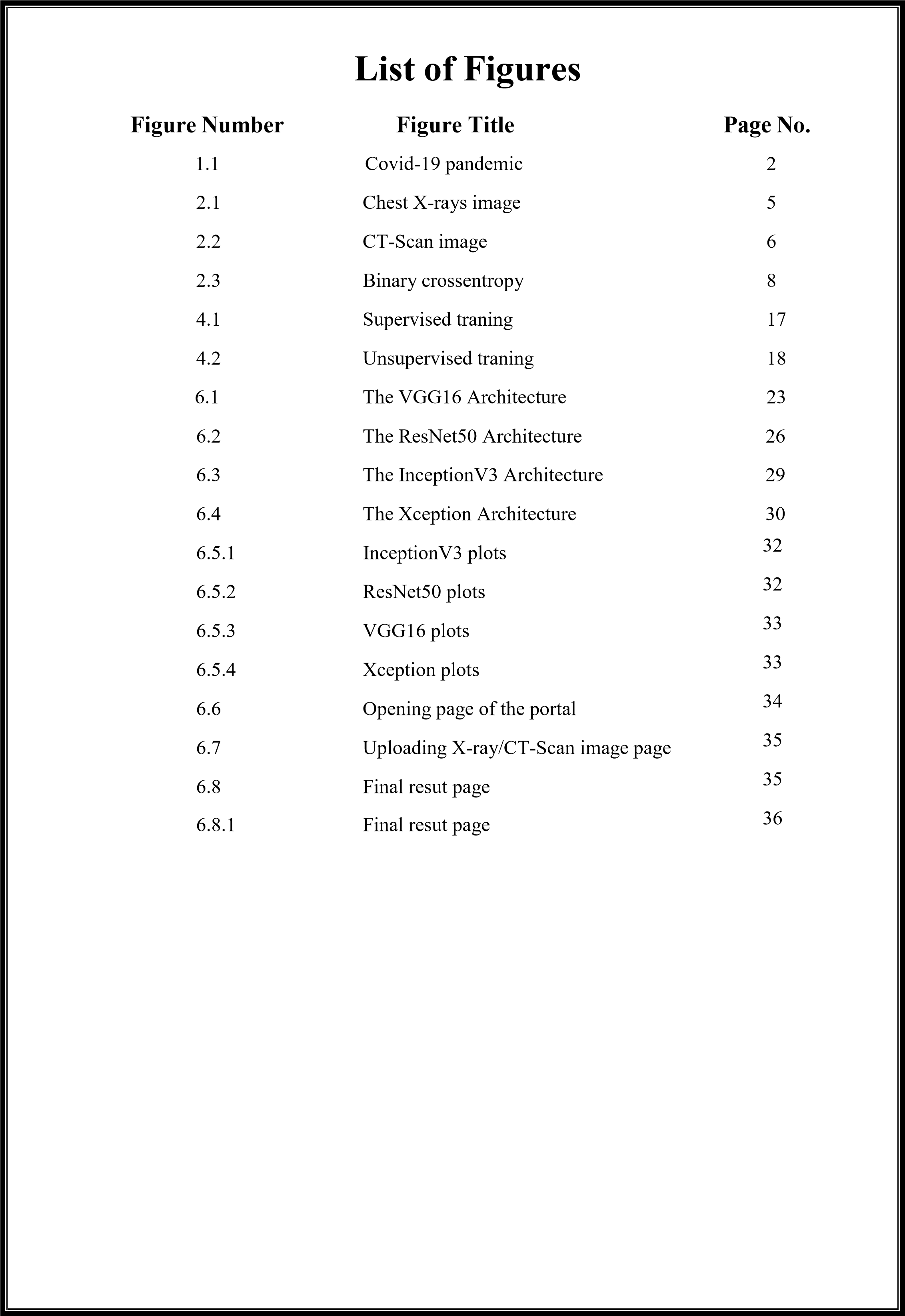
Shoiab Shaikh

Devarshi Wadadkar

5







**Chapter 1**

**INTRODUCTION**

The World Health Organisation (WHO) has declared the coronavirus disease 2019 (COVID-19) a pandemic. A global coordinated effort is needed to stop the further spread of the virus. A pandemic is defined as “occurring over a wide geographic area and affecting an exceptionally high proportion of the population.” The last [pandemic r](https://www.physio-pedia.com/Endemics%2C_Epidemics_and_Pandemics)eported in the world was the H1N1 flu pandemic in 2009.

On 31 December 2019, a cluster of cases of [pneumonia o](https://www.physio-pedia.com/Pneumonia)f unknown cause, in the city of Wuhan, Hubei province in China, was reported to the World Health Organisation. In January 2020, a previously unknown new [virus w](https://www.physio-pedia.com/Viral_Infections)as identified, subsequently named the 2019 novel coronavirus, and samples obtained from cases and analysis of the virus’ genetics indicated that this was the cause of the outbreak. This novel [coronavirus w](https://www.physio-pedia.com/Coronaviruses)as named Coronavirus Disease 2019 (COVID-19) by WHO in February 2020. The virus is referred to as [SARS-](https://www.physio-pedia.com/SARS_Severe_Acute_Respiratory_Syndrome)CoV-2 and the associated disease is COVID-19.

In March 2020 and after spreading to more than 100 countries and leading to several thousands of cases, the World Health Organization (WHO) officially declared the outbreak of the new coronavirus as a pandemic. Although COVID-19 affects the entire population, young people that have been affected by COVID 19 are in most cases either asymptomatic or present mild symptoms like cough, headache, fatigue and fever. For non-young ages and especially for elders and/or for patients with chronic conditions COVID-19 positive cases may progress to more serious symptoms like diarrhea, dyspnea, pneumonia and death. Young and middle-aged patients being diagnosed with COVID-19 are having significantly lower mortality rates comparing to elder patients with COVID-19 which are more likely to progress to severe disease.

|  |
| --- |
| **COVID-19 pandemic** |
| Confirmed cases per 100,000 population as of 7 June 2020:  >1000 10–30  300–1000 0–10  100–300 None or no data  30–100    show  **Total confirmed cases**    show  **Confirmed deaths (per 1,000,000 population)**    show  **Confirmed cases (per capita interactive timeline.** |

Fig 1.1 Covid-19 pandemic

With COVID-19 being highly infectious it can easily be spread from asymptomatic to vulnerable population. To stop the spread of COVID-19 virus and to protect vulnerable people many countries worldwide have introduced isolation measures like social distancing and lockdown and in parallel they perform diagnostic tests to key staff and the general population to detect COVID-19 positive cases. level, and so forth. It affects the hormone insulin, resulting in abnormal metabolism of crabs and improves level of sugar in the blood. According to World Health Organization about 422 million people suffering from diabetes particularly from low or idle income countries. And this could be increased to 490 billion up to the year of 2030. However prevalence of diabetes is found among various Countries like Canada, China, and India etc. Population of India is now more than 100 million so the actual number of diabetics in India is 40 million.

## 1.1 Problem Definition

The standard method of diagnosis is by real-time reverse transcription polymerase chain reaction (RT-PCR) from a nasopharyngeal swab. Chest CT imaging may also be helpful for diagnosis in individuals where there is a high suspicion of infection based on symptoms and risk factors. However most of them do have x-ray & CT-scan machines which can be used for COVID19 detection with a suitable model to analyse that x-ray & CT-scan. Although x- rays & CT-scan can be used for complete screening of COVID19, it could be a very effective pre-screening tool. Patient’s whose treatment might be delayed waiting for the result of the RT-PCR test could be avoided by using x-rays & CT-scan as a pre-screening tool.

## 1.2 Objectives

The goal of the project is to classify indications of X Ray & CT-scan of patients as early as possible based on Machine Learning Methods. This can be split up into the following objectives:

1. To analyse the portion of the body affected vastly by the virus.

1. Finest Dataset gathering of required x-rays & CT-scan.

1. Devising methods to diagnose it effectively.

1. Designing the model and testing for its effectiveness.

1. To derive conclusions from the model designed.

## 1.3 Approaches

The usual approach followed by WHO and other major institutions are using the RT-PCR testing or anti-viral testing. These testing options take a lot of time or are not that accurate. [Anti-body tests a](https://www.thehindu.com/sci-tech/health/coronavirus-can-antibody-tests-help-tackle-covid-19/article31490919.ece)lso known as serological tests, are used to find out the presence of virus in a body. In this method of testing, blood samples are used to find anti-bodies. This process also detects the quantity of antibodies that are produced by the immune system. The RT-PCR test is a method of testing by taking a nasal/throat swab from a patient. It involves extracting ribonucleic acid or RNA, which is the genetic material of the virus.

### 1.3.1 Our Approach

The approach we suggest is to make use of X-rays and CT-scans of COVID19 patient’s and the x-rays,CT-scans of a healthy person to spot the differences to where the virus affects. Usually the first and the worst affected organ is the lungs. COVID19 can be identified as a white patch on both the x-ray and CT-scan film. We can also use x-rays and CT-scans of people who have some other kind of infection as they are not misdiagnosed as COVID19. Not only is this approach going to be faster to detect but is also cost effective. This could also be used for pre-screening which could limit the delay a patient has to go through before being declared positive by the rRT-PCR testing.

# CHAPTER 2

**LITERATURE SURVEY**

##### 2.1 Dataset

Sample an open source dataset of X-ray & CT-scan images for patients who have tested positive for COVID19. Sample “normal” (i.e., not infected) X-ray & CT-scan images from healthy patients. Train a Algorithm to automatically detect COVID-19 in X-ray & CT- scan images via the dataset we created. The COVID-19 Xray and CT-scan image dataset we’ll be using for this project was curated by Dr. Joseph Cohen, a postdoctoral fellow at the University of Montreal. Dr. Cohen started collecting X-ray images of COVID-19 cases and publishing them in the GitHub repo.

The learning set contains around 500 images of COVID positive x-rays & 350 images CT-scan of the lungs and around 500 images of COVID negative x-rays & 350 images CT-scan. These non COVID x-rays & CT-scan include all types of lung deformities that could be confused with COVID. We also take into consideration of the x-rays & CT- scan of smokers to improve the efficiency of the model.

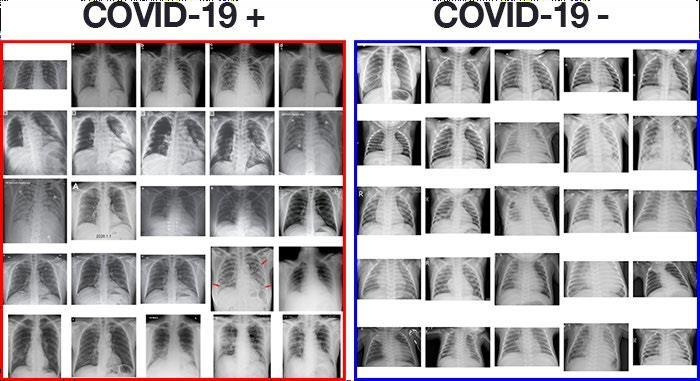


Fig 2.1 :- Chest X-rays image

##### POSITIVE COVID-19 CT-SCAN NEGATIVE COVID-19 CT-SCAN

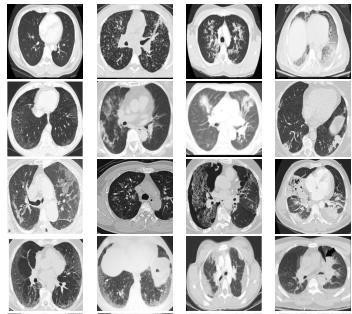
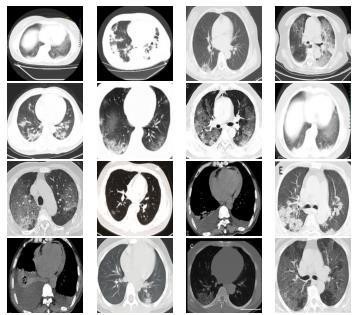


Fig 2.2:- CT-Scan image

###### 2.2 File Structure

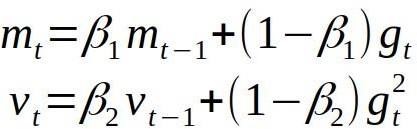
Any standard form of image file is accepted by this model. All formats other than image are not accepted by the model. Image file is easier to handle when analysing the model. We may have a conflict when the person uploads an image which is not an x-ray or CT-scan. This might lead to further complications regarding the result of the model. So there is another model employed with the sole purpose of accepting only x-rays or CT-scan. All other images get rejected by the model. This increases the scalability of the model.

###### 2.3 Learning Mechanism

The model relies on something called learning rate decay. Learning rate decay (lrDecay) is a technique for training modern neural networks. It starts with a large learning rate and then decays it multiple times. It is empirically observed to help both optimization and generalization. To achieve this we make use of Adam optimizer. Given that this is a 2-class problem, we use "binary\_crossentropy" loss rather than categorical crossentropy

2.3.1 Adam Optimizer

Adam can be looked at as a combination of RMSprop and Stochastic Gradient Descent with momentum. It uses the squared gradients to scale the learning rate like RMSprop and it takes advantage of momentum by using moving average of the gradient instead of gradient itself like SGD with momentum. The first moment is mean, and the second moment is uncentered variance (meaning we don’t subtract the mean during variance calculation). We will see later how we use these values, right now, we have to decide on how to get them. To estimates the moments, Adam utilizes exponentially moving averages, computed on the gradient evaluated on a current minibatch:



Where m and v are moving averages, g is gradient on current mini-batch, and betas — new introduced hyper-parameters of the algorithm. They have really good default values of 0.9 and 0.999 respectively. Almost no one ever changes these values. The vectors of moving averages are initialized with zeros at the first iteration.

2.3.2 Binary Crossentropy

Binary crossentropy is a loss function that is used in binary classification tasks. These are tasks that answer a question with only two choices (yes or no, A or B, 0 or 1, left or right). Several independent such questions can be answered at the same time, as in multi-label classification or in binary image segmentation. Formally, this loss is equal to the average of the categorical crossentropy loss on many two-category tasks

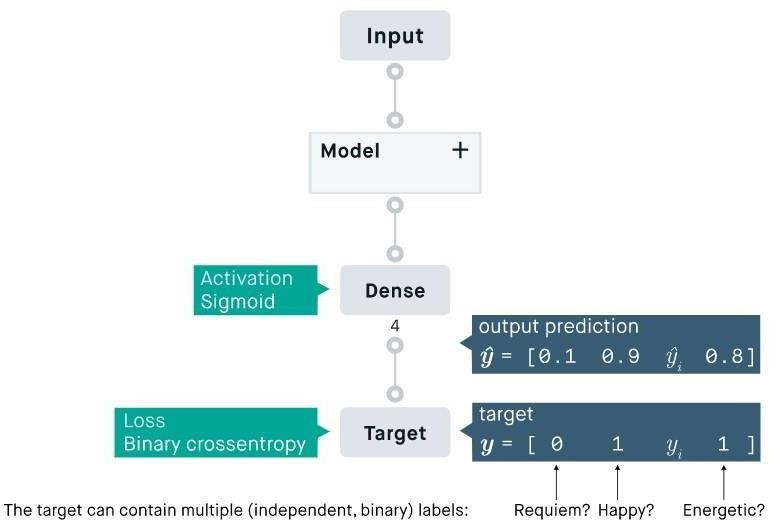


Fig 2.3 :- Binary Crossentropy

#### 2.4 Related Works

* **Khair Ahammed ,Md. Shahriare Satu and Mohammad Zoynul Abedin’s** paper on early detection of Covid 19 uses the concept of convolutional neural networks to analyze the X-rays. Detection through X-rays might be able to reduce the testing time for Covid by a huge factor. Various approaches were suggested by experts such as AlexNet, VGGNet, DeTrac Neural networks. The usual procedure for detection can consist of Data Collection, pre-processing, classification and evaluation. Data was divided into 3 types which were normal, Covid and Pneumonia. Data for training the model were collected from the public available datasets of COVID-19 Radiography Database and Cohen et al Database. Metrics like accuracy, AVC, F- measure, sensitivity , specificity ,fall out and miss rate were considered. Proposed CNN with 2 convolution layers of 32 and 64 filters provided an accuracy of 94.3%.
* **Farhana Sultana , Abu Sufian and Paramartha Dutta’s** paper on evolution of CNN from LeNet-5 to SENet Model show us how various factors in image analysis were overcome. CNN is composed of single or multiple blocks of convolution and subsampling layer. Convolution layers are the heart of the CNN and work on the static nature of images. Pooling layers are used to sample down an image to reduce the number of parameters for analysis. AlexNet is used to classify ImageNet data

and uses around 3 convolution layers with RELU. ZFNet is similar to AlexNet but uses deconvolution network. VGGNet uses smaller filters and is more focused on reducing the parameters being evaluated. SENet uses ‘Squeeze and Excitation’ on individual channels to reduce error from 5 to 2.5% ans is considered the most optimal model.

##### • Muthukrishnan Ramprasath, M.Vijay Anand and Shanmugasundaram

**Hariharan’s** paper on solving image analysis problems uses CNN with Digit of

MNSIT data sets as a bench mark for classification of greyscale images with around 98% accuracy. CNN extracts features from Hyper Spectral Images using CNN deep learning which can be used for perfect classification and detection. CNN is a feed forward network inspired by neurons in animal vision cortex. Training has 4 parts of pre-processing, feature extraction(using LBP or convolution network), data normalization and classification using SVM. Proposed model has 3 layers with 2 convolution layers of 32 and 64 filters with maxpooling layer as the 3rd. Stacking model with more layers and using more images to train will improve the accuracy by a great factor.

* **Tulin Ozturk , Muhammed Talo , Eylul Azra Yildirim , Ulas Baran Baloglu’s** paper on detecting COVID through X-ray and CT scans provides an alternative for the RT-PCR test. RT-PCR test is fairly 60%-70% accurate. To overcome this doctors in Turkey started using CT scans for analysing early onset of COVID. This called for shortage of radiologists. Therefore to overcome this an AI model was suggested to analyse the scans. Instead of creating a model from scratch, a pre existing model such as DarkNet-19 was adopted. It constituted 19 convolution layers with 5 max pooling layers. In conclusion this model saw an accuracy of 98.08% with sensitivity of 85.35%, specificity of 92.18% and 87.37% F1-score.
* **Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton’s** paper on image analysis using deep neural networks was used for ImageNet LSVRC-2010 contest. It manages to obtain 2nd place. It could obtain normal human performance at normal recognition tasks. Shortcomings in real world testing was overcome by using larger datasets. Images in datasets of varied resolutions were down sampled to 256 X 256 pixel resolution. The model was trained on multiple GPU’s with RELU. Therefore it has 5 convolution layers with 3 fully connected layers. We have to be considerate about overfitting and dropout factor. It achieved top-1 and top-5 error rates of 67.4% and 40.9%.
* Rahib H. Abiyev and Mohammad Khaleel Sallam Ma’aitah’s paper on Deep Neural networks for chest Disease detection deals with which type or neural networks to consider while analysing x-rays. X-rays are effective ways to reveal pathological alteration, in addition to its non-invasive characteristics and economic factors. Classifying the chest xray abnormalities is a tedious task for radiologists, here algorithms were developed to reduce the workload. Convolutional neural networks use the brain structure of mammals by abstracting different levels of features. The input images of size 32X32 were considered to reduce computation time. Learning rate of 0.010, momentum rate of 0.040 and sigmoid activation functions were used. It was seen CNN was more effective than BPNN and CpNN although it needed more computation time.

**CHAPTER 3 TECHNICAL SURVEY**

#### 3.1 Python

Python is an interpreted, high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales. Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library. Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

Python interpreters are available for many operating systems. A global community of programmers develops and maintains CPython, an open source reference implementation. A nonprofit organization, the Python Software Foundation, manages and directs resources for Python and CPython development.

##### 3.1.1 History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68,

SmallTalk, and Unix shell and other scripting languages. Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL). Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

##### 3.1.2 Python Features

Python's features include:

* Easy to learn: Python has few keywords, simple structure, and a clearly defined syntax.

This allows the student to pick up the language quickly.

* Easy to read: Python code is more clearly defined and visible to the eyes.
* Easy to maintain: Python's source code is fairly easy-to-maintain.
* Interactive Mode: Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
* Portable: Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
* Extendable: You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
* Scalable: Python provides a better structure and support for large programs than shell scripting.

##### 3.1.3 Uses of Python

Python is used by hundreds of thousands of programmers and is used in many places. Sometimes only Python code is used for a program, but most of the time it is used to do simple jobs while another programming language is used to do more complicated tasks. Its standard library is made up of many functions that come with Python when it is installed. On the Internet there are many other libraries available that make it possible for the Python language to do more things. These libraries make it a powerful language; it can do many different things.

Some things that Python is often used for are:

* Web development
* Game programming
* Desktop GUIs • Scientific programming
* Network programming.

**3.2 Libraries used**

#### Numpy

NumPy is a library for the Python programming language, adding support for large, multidimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. The ancestor of NumPy, Numeric, was originally created by Jim Hugunin with contributions from several other developers. In 2005, Travis Oliphant created NumPy by incorporating features of the competing Num array into Numeric, with extensive modifications. NumPy is open-source software and has many contributors.

##### Matplotlib

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython.

##### Keras

Keras is an open-source neural-network library written in Python. It is capable of running on top of TensorFlow, Microsoft Cognitive Toolkit, Theano, or PlaidML. Designed to enable fast experimentation with deep neural networks, it focuses on being user-friendly, modular, and extensible. 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. The code is hosted on GitHub, and community support forums include the GitHub issues page, and a Slack channel.

##### 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 was developed by the Google Brain team for internal Google use. It was released under the Apache License 2.0 on November 9, 2015. TensorFlow provides stable Python (for version 3.7 across all platforms) and C APIs; and without API backwards compatibility guarantee: C++, Go, Java, JavaScript and Swift (early release). Third-party packages are available for C#, Haskell, Julia, R, Scala, Rust, OCaml, and Crystal.

##### Scikit-Learn

Scikit-learn (formerly scikits.learn) is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy. The scikit-learn project started as scikits.learn, a Google Summer of Code project by David Cournapeau. Its name stems from the notion that it is a

"SciKit" (SciPy Toolkit), a separately-developed and distributed third-party extension to SciPy. The original codebase was later rewritten by other developers. Of the various scikits, scikit-learn as well as scikit-image were described as "wellmaintained and popular" in November 2012. As of 2018, scikit-learn is under active development.

##### OpenCV-Python

OpenCV (Open Source Computer Vision Library) is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by Willow Garage then Itseez (which was later acquired by Intel The library is cross-platform and free for use under the open-source BSD license.) OpenCV-Python makes use of Numpy, which is a highly optimized library for numerical operations with a MATLAB-style syntax. All the OpenCV array structures are converted to and from Numpy arrays. This also makes it easier to integrate with other libraries that use Numpy such as SciPy and Matplotlib.

##### ImageNet

ImageNet is a dataset of over 15 million labelled high-resolution images belonging to roughly 22,000 categories. The images were collected from the web and labelled by human labellers using Amazon’s Mechanical Turk crowd-sourcing tool. Starting in 2010, as part of the Pascal Visual Object Challenge, an annual competition called the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) has been held. ILSVRC uses a subset of ImageNet with roughly 1000 images in each of 1000 categories. In all, there are roughly 1.2 million training images, 50,000 validation images, and 150,000 testing images. ImageNet consists of variable-resolution images. Therefore, the images have been down-sampled to a fixed resolution of 256×256. Given a rectangular image, the image is rescaled and cropped out the central 256×256 patch from the resulting image.

#### Seaborn

Seaborn is a library for making statistical graphics in Python. It builds on top of matplotlib and integrates closely with pandas data structures. Seaborn helps you explore and understand your data. Its plotting functions operate on dataframes and arrays containing whole datasets and internally perform the necessary semantic mapping and statistical aggregation to produce informative plots. Its dataset-oriented, declarative API lets you focus on what the different elements of your plots mean, rather than on the details of how to draw them.

#### Flask

[Flask i](http://flask.pocoo.org/)s a web framework. This means flask provides you with tools, libraries and technologies that allow you to build a web application. This web application can be some web pages, a blog, a wiki or go as big as a web-based calendar application or a commercial website. Flask is part of the categories of the micro-framework. Micro-framework are normally framework with little to no dependencies to external libraries. This has pros and cons. Pros would be that the framework is light, there are little dependency to update and watch for security bugs, cons is that some time you will have to do more work by yourself or increase yourself the list of dependencies by adding plugins.

**CHAPTER 4 MACHINE LEARNING**

Machine learning is a field of computer science that often uses statistical techniques to give computer systems the ability to "learn" (e.g., progressively improve performance on a specific task) with data, without being explicitly programmed.

The name machine learning was coined in 1959 by Arthur Samuel. Evolved from the study of pattern recognition and computational learning theory in artificial intelligence, machine learning explores the study and construction of algorithms that can learn from and make predictions on data – such algorithms overcome following strictly static program instructions by making datadriven predictions or decisions, through building a model from sample inputs. Machine learning is employed in a range of computing tasks where designing and programming explicit algorithms with good performance is difficult or infeasible; example applications include email filtering, detection of network intruders or malicious insiders working towards a data breach, optical character recognition (OCR), learning to rank, and computer vision.

Machine learning is closely related to (and often overlaps with) computational statistics, which also focuses on prediction-making through the use of computers. It has strong ties to mathematical optimization, which delivers methods, theory and application domains to the field. Machine learning is sometimes conflated with data mining, where the latter subfield focuses more on exploratory data analysis and is known as unsupervised learning. Machine learning can also be unsupervised and be used to learn and establish baseline behavioural profiles for various entities and then used to find meaningful anomalies.

Within the field of data analytics, machine learning is a method used to devise complex models and algorithms that lend themselves to prediction; in commercial use, this is known as predictive analytics. These analytical models allow researchers, data scientists, engineers, and analysts to "produce reliable, repeatable decisions and results" and uncover "hidden insights" through learning from historical relationships and trends in the data.

Machine learning tasks are typically classified into two broad categories, depending on whether there is a learning "signal" or "feedback" available to a learning system:

#### 4.1 Supervised learning

Supervised learning is when the model is getting trained on a labelled dataset. Labelled dataset is one which has both input and output parameters. In this type of learning both training and validation sets are labelled. (Refer Fig 4.1).

While training the model, data is usually split in the ratio of 80:20 i.e. 80% as training data and rest as testing data. In training data, we feed input as well as output for 80% data. The model learns from training data only. We use different machine learning algorithms (which we will discuss in detail in next articles) to build our model. By learning, it means that the model will build some logic of its own.

Once the model is ready then it is good to be tested. At the time of testing, input is fed from remaining 20% data which the model has never seen before, the model will predict some value and we will compare it with actual output and calculate the accuracy.

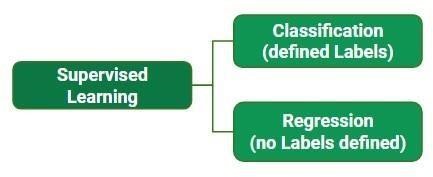


Fig 4.1: Supervised training Example of Supervised Learning Algorithms:

* Linear Regression

* Nearest Neighbor

* Guassian Naive Bayes

* Decision Trees

* Support Vector Machine (SVM)

##### 4.1.1 Reinforcement Learning

Reinforcement is about taking suitable action to maximize reward in a particular situation. It is employed by various software and machines to find the best possible behaviour or path it should take in a specific situation. Reinforcement learning differs from the supervised learning in a way that in supervised learning the training data has the answer key with it so the model is trained with the correct answer itself whereas in reinforcement learning, there is no answer but the reinforcement agent decides what to do to perform the given task. In the absence of training dataset, it is bound to learn from its experience.

Main points in Reinforcement learning –

* Input: The input should be an initial state from which the model will start

* Output: There are many possible output as there are variety of solution to a particular problem
* Training: The training is based upon the input, The model will return a state and the user will decide to reward or punish the model based on its output. • The model keeps continues to learn.

* The best solution is decided based on the maximum reward.

##### 4.1.2 Unsupervised Learning

Unsupervised learning is the training of machine using information that is neither classified nor labelled and allowing the algorithm to act on that information without guidance. Here the task of machine is to group unsorted information according to similarities, patterns and differences without any prior training of data (Refer Fig 4.2). Unlike supervised learning, no teacher is provided that means no training will be given to the machine. Therefore machine is restricted to find the hidden structure in unlabelled data by our-self.

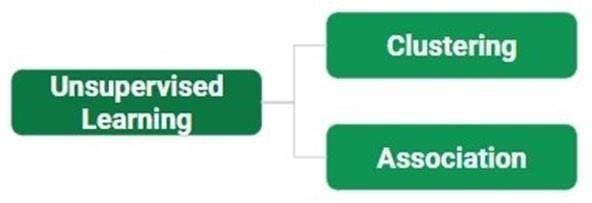


Fig 4.2: Unsupervised Training

Another categorization of machine learning tasks arises when one considers the desired output of a machine-learned system:

* In classification, inputs are divided into two or more classes, and the learner must produce a model that assigns unseen inputs to one or more (multi-label classification) of these classes. This is typically tackled in a supervised way.
* In regression, also a supervised problem, the outputs are continuous rather than discrete.
* In clustering, a set of inputs is to be divided into groups. Unlike in classification, the groups are not known beforehand, making this typically an unsupervised task.
* Dimensionality reduction simplifies inputs by mapping them into a lower- dimensional space. Topic modelling is a related problem, where a program is given a list of human language documents and is tasked to find out which documents cover similar topics.

#### 4.2 Approaches to machine learning

##### 4.2.1 Deep Learning

Deep learning (also known as deep structured learning or hierarchical learning) is part of a broader family of machine learning methods based on artificial neural networks. Learning can be supervised, semi-supervised or unsupervised.

Deep learning architectures such as deep neural networks, deep belief networks, recurrent neural networks and convolutional neural networks have been applied to fields including computer vision, speech recognition, natural language processing, audio recognition, social network filtering, machine translation, bioinformatics, drug design, medical image analysis, material inspection and board game programs, where they have produced results comparable to and in some cases superior to human experts. Artificial Neural Networks (ANNs) were inspired by information processing and distributed communication nodes in biological systems. ANNs have various differences from biological brains. Specifically, neural networks tend to be static and symbolic, while the biological brain of most living organisms is dynamic (plastic) .

##### 4.2.2 Convolutional Neural Network (CNN)

CNNs are regularized versions of multilayer perceptrons. Multilayer perceptrons usually refer to fully connected networks, that is, each neuron in one layer is connected to all neurons in the next layer. The "fully-connectedness" of these networks makes them prone to overfitting data. Typical ways of regularization includes adding some form of magnitude measurement of weights to the loss function. However, CNNs take a different approach towards regularization: they take advantage of the hierarchical pattern in data and assemble more complex patterns using smaller and simpler patterns. Therefore, on the scale of connectedness and complexity, CNNs are on the lower extreme. They are also known as shift invariant or space invariant artificial neural networks (SIANN), based on their shared-weights architecture and translation invariance characteristics.

Convolutional networks were inspired by biological processes in that the connectivity pattern between neurons resembles the organization of the animal visual cortex. Individual cortical neurons respond to stimuli only in a restricted region of the visual field known as the receptive field. The receptive fields of different neurons partially overlap such that they cover the entire visual field. CNNs use relatively little pre- processing compared to other image classification algorithms. This means that the network learns the filters that in traditional algorithms were hand-engineered. This independence from prior knowledge and human effort in feature design is a major advantage. They have applications in image and video recognition, recommender systems, image classification, medical image analysis, and natural language processing.

###### 4.2.3 Applications of Machine Learning

The applications of Machine learning include:

* Agriculture
* Automated theorem proving
* Adaptive websites
* Affective computing
* Bioinformatics
* Brain–machine interfaces
* Classifying DNA sequences
* Computational anatomy
* Computer Networks
* Telecommunication
* Computer vision, including object recognition
* Detecting credit-card fraud
* General game playing
* Information retrieval

**CHAPTER 5**

**SYSTEM REQUIREMENTS**

**5.1 Hardware Requirements:**

1. Processor: Intel Core i3 or better / AMD Phenom ii X6 or better
2. Operating System: Windows 10 or Linux based Operating System
3. RAM: 8 GB
4. Storage: 256 GB Solid State Drive or Higher

**5.2 Software Requirements:**

1. Language: Python with the following modules installed:

* Keras
* Virtualenv(not mandatory)
* Tensorflow
* Numpy
* OpenCV
* Matplotlib
* Scikit learn
* ImageNet
* SeaBorn
* Flask

**CHAPTER 6 IMPLEMENTATION**

#### 6.1 VGG16

VGG16 is a convolutional neural network model proposed by K. Simonyan and A.

Zisserman from the University of Oxford in the paper “Very Deep Convolutional Networks for Large-Scale Image Recognition”. The model achieves 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 model submitted to ILSVRC-2014.

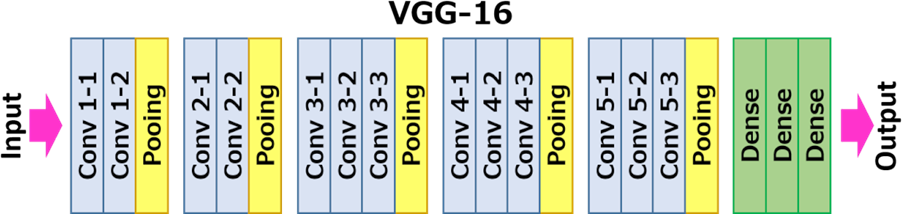


Fig 6.1:- The VG16 Architecture

The input to cov1 layer is of fixed size 224 x 224 RGB image. The image is passed through a stack of convolutional (conv.) layers, where the filters were used with a very small receptive field: 3×3 (which is the smallest size to capture the notion of left/right, up/down, center). In one of the configurations, it also utilizes 1×1 convolution filters, which can be seen as a linear transformation of the input channels (followed by non-linearity). The convolution stride is fixed to 1 pixel; the spatial padding of conv. layer input is such that the spatial resolution is preserved after convolution, i.e. the padding is 1-pixel for 3×3 conv. layers. Spatial pooling is carried out by five max-pooling layers, which follow some of the conv. layers (not all the conv. layers are followed by max-pooling). Max-pooling is performed over a 2×2 pixel window, with stride 2.

Three Fully-Connected (FC) layers follow a stack of convolutional layers (which has a different depth in different architectures): the first two have 4096 channels each, the third performs 1000way 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.

All hidden layers are equipped with the rectification (ReLU) non-linearity. It is also noted that none of the networks (except for one) contain Local Response Normalisation (LRN), such normalization does not improve the performance on the ILSVRC dataset, but leads to increased memory consumption and computation time.

**model.summary()**

**Model: "model"**

**Layer (type) Output Shape Param #**

**========================================================= input\_1 (InputLayer) [(None, 224, 224, 3)] 0**

|  |  |  |
| --- | --- | --- |
| **block1\_conv1 (Conv2D) (None, 224, 224, 64)** | | **1792** |
| **block1\_conv2 (Conv2D) (None, 224, 224, 64)** | | **36928** |
| **block1\_pool (MaxPooling2D) (None, 112, 112, 64)** | | **0** |
| **block2\_conv1 (Conv2D) (None, 112, 112, 128)** | | **73856** |
| **block2\_conv2 (Conv2D) (None, 112, 112, 128)** | | **147584** |
| **block2\_pool (MaxPooling2D) (None, 56, 56, 128)** | | **0** |
| **block3\_conv1 (Conv2D)** | **(None, 56, 56, 256)** | **295168** |
| **block3\_conv2 (Conv2D)** | **(None, 56, 56, 256)** | **590080** |
| **block3\_conv3 (Conv2D)** | **(None, 56, 56, 256)** | **590080** |
| **block3\_conv4 (Conv2D)** | **(None, 56, 56, 256)** | **590080** |

**block3\_pool (MaxPooling2D) (None, 28, 28, 256) 0**

|  |  |  |
| --- | --- | --- |
| **block4\_conv1 (Conv2D)** | **(None, 28, 28, 512)** | **1180160** |
| **block4\_conv2 (Conv2D)** | **(None, 28, 28, 512)** | **2359808** |
| **block4\_conv3 (Conv2D)** | **(None, 28, 28, 512)** | **2359808** |
| **block4\_conv4 (Conv2D)** | **(None, 28, 28, 512)** | **2359808** |
| **block4\_pool (MaxPooling2D) (None, 14, 14, 512)** | | **0** |
| **block5\_conv1 (Conv2D) (None, 14, 14, 512)** | | **2359808** |

**block5\_conv2 (Conv2D) (None, 14, 14, 512) 2359808**

**block5\_conv3 (Conv2D) (None, 14, 14, 512) 2359808**

|  |  |
| --- | --- |
| **block5\_conv4 (Conv2D) (None, 14, 14, 512)** | **2359808** |
| **block5\_pool (MaxPooling2D) (None, 7, 7, 512)** | **0** |
| **flatten (Flatten) (None, 25088) 0** |  |

**dropout (Dropout) (None, 25088) 0**

**dense (Dense) (None, 2) 50178**

**=================================================================**

**Total params: 20,074,562**

**Trainable params: 50,178**

**Non-trainable params: 20,024,384**

#### 6.2 ResNet 50

In 2012 at the LSVRC2012 classification contest [AlexNet w](https://iq.opengenus.org/architecture-and-use-of-alexnet/)on the the first price, After that ResNet was the most interesting thing that happened to the computer vision and the deep learning world.

Because of the framework that ResNets presented it was made possible to train ultra deep neural networks and by that i mean that i network can contain hundreds or thousands of layers and still achieve great performance.

The ResNets were initially applied to the image recognition task but as it is mentioned in the paper that the framework can also be used for non computer vision tasks also to achieve better accuracy.

Many of you may argue that simply stacking more layers also gives us better accuracy why was there a need of Residual learning for training ultra deep neural networks.

**Building ResNet and 1× 1 Convolution:**

We will build the ResNet with 50 layers following the method adopted in the original paper by He. et al. The architecture adopted for ResNet-50 is different from the 34 layers architecture. The shortcut connection skips 3 blocks instead of 2 and, the schematic diagram below will help us clarify some points

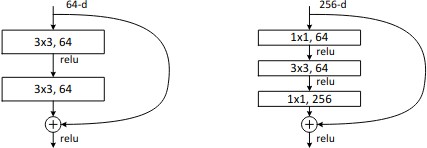


Fig :-Left: Skip 2 layers, ResNet-34. Right: Skip 3 layers including 1× 1 convolution in ResNet-50.

The identity and convolution blocks are then combined to create a ResNet-50 model with the architecture shown below:

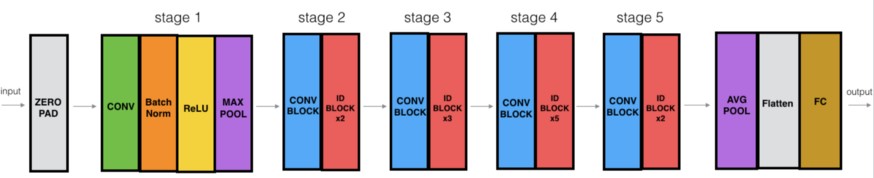


Fig 6.2 :-The ResNet-50 Architecture

The ResNet-50 model consists of 5 stages each with a convolution and Identity block. Each convolution block has 3 convolution layers and each identity block also has 3 convolution layers. The ResNet-50 has over 23 million trainable parameters.

**Model summary():**



##### 6.3 Inception-v3

**Inception-v3** is a convolutional neural network architecture from the Inception family that makes several improvements including using Label Smoothing, Factorized 7 x 7

convolutions, and the use of an auxiliary classifer to propagate label information lower down the network (along with the use of batch normalization for layers in the sidehead).

##### Inception v3 Architecture

The architecture of an Inception v3 network is progressively built, step-by-step, as explained below:

1. **Factorized Convolutions:** this helps to reduce the computational efficiency as it reduces the number of parameters involved in a network. It also keeps a check on the network efficiency.
2. **Smaller convolutions:** replacing bigger convolutions with smaller convolutions definitely leads to faster training. Say a 5 × 5 filter has 25 parameters; two 3 × 3 filters replacing a 5 × 5 convolution has only 18 (3\*3 + 3\*3) parameters instead.
3. **Asymmetric convolutions:** A 3 × 3 convolution could be replaced by a 1 × 3 convolution followed by a 3 × 1 convolution. If a 3 × 3 convolution is replaced by a 2 × 2 convolution, the number of parameters would be slightly higher than the asymmetric convolution proposed.
4. **Auxiliary classifier:** an auxiliary classifier is a small CNN inserted between layers during training, and the loss incurred is added to the main network loss. In GoogLeNet auxiliary classifiers were used for a deeper network, whereas in Inception v3 an auxiliary classifier acts as a regularizer.
5. **Grid size reduction:** Grid size reduction is usually done by pooling operations. However, to combat the bottlenecks of computational cost, a more efficient technique is proposed:

All the above concepts are consolidated into the final architecture.

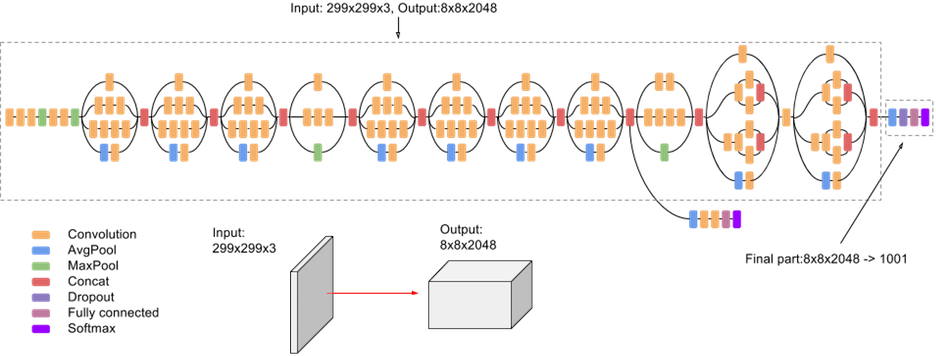
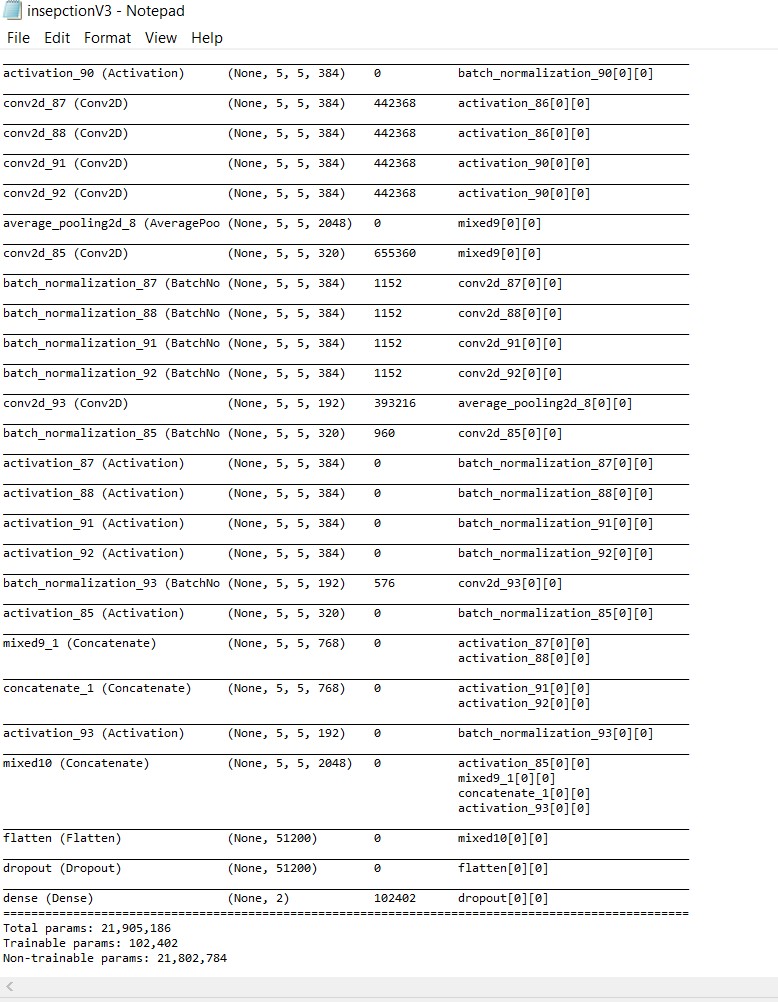


Fig 6.3:- The Inception v3 Architecture

The model itself is made up of symmetric and asymmetric building blocks, including convolutions, average pooling, max pooling, concats, dropouts, and fully connected layers. Batchnorm is used extensively throughout the model and applied to activation inputs. Loss is computed via Softmax.

###### Model summary ();



###### 6.4 Xception

The **Xception architecture** has 36 convolutional layers forming the feature extraction base of the network. In short, the **Xception architecture** is a linear stack of depthwise separable convolution layers with residual connections.

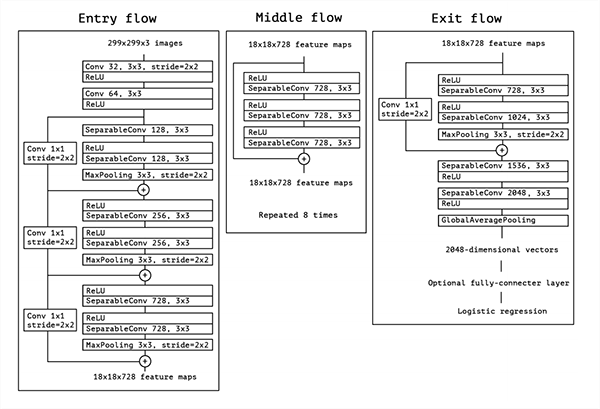


Fig 6.4**:** The Xception architecture.

Convolutional Neural Networks (CNN) have come a long way, from the LeNet-style, AlexNet, VGG models, which used simple stacks of convolutional layers for feature extraction and max-pooling layers for spatial sub-sampling, stacked one after the other, to Inception and ResNet networks which use skip connections and multiple convolutional and max-pooling blocks in each layer. Since its introduction, one of the best networks in computer vision has been the Inception network. The Inception model uses a stack of modules, each module containing a bunch of feature extractors, which allow them to learn richer representations with fewer parameters.

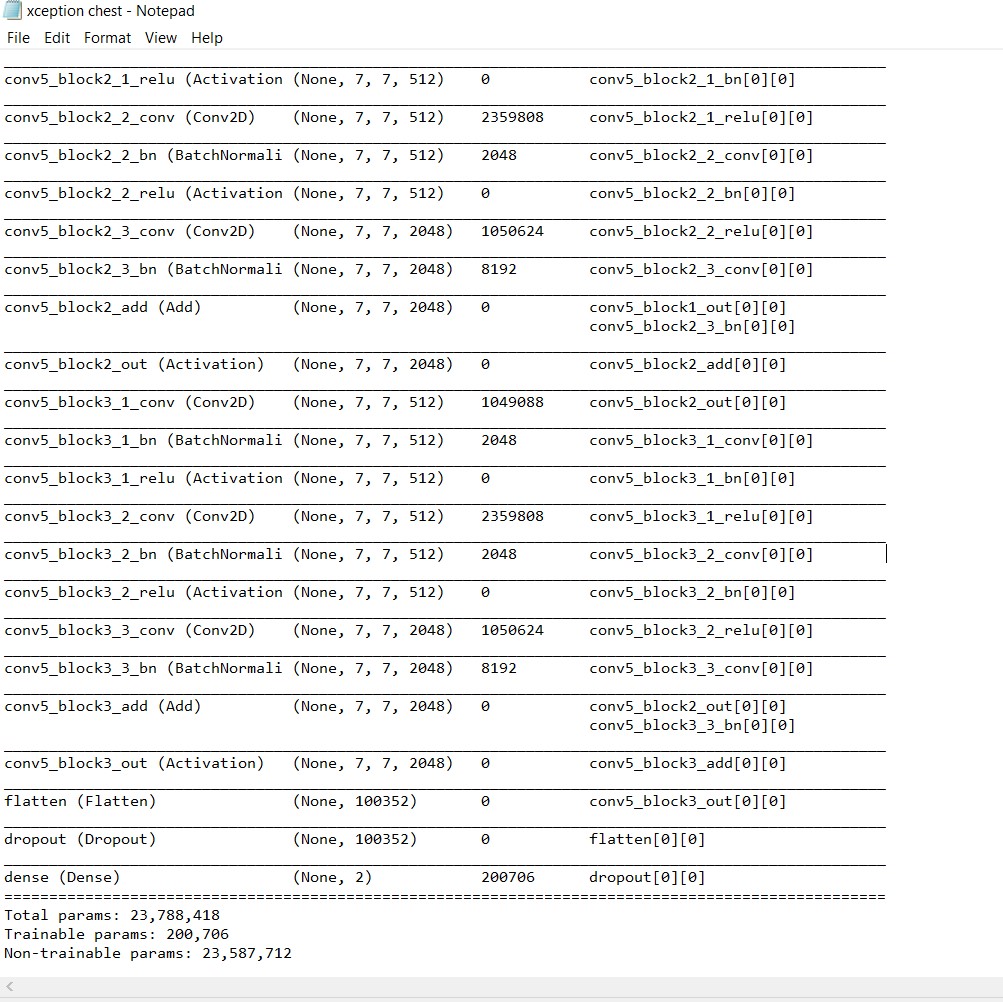
The Xception module has 3 main parts. The Entry flow, the Middle flow (which is repeated 8 times), and the Exit flow.

The entry flow has two blocks of convolutional layer followed by a ReLU activation. The diagram also mentions in detail the number of filters, the filter size (kernel size), and the strides There are also various Separable convolutional layers. There are also Max Pooling layers. When the strides are different than one, the strides are also mentioned. There are also

Skip connections, where we use ‘ADD’ to merge the two tensors. It also shows the shape of the input tensor in each flow. For example, we begin with an image size of 299x299x3, and after the entry flow, we get an image size of 19x19x728.

Similarly, for the Middle flow and the Exit flow, this diagram clearly explains the image size, the various layers, the number of filters, the shape of filters, the type of pooling, the number of repetitions, and the option of adding a fully connected layer in the end.

**Model Summary():**



6.5 Epochs

The number of epochs is a hyperparameter that defines the number times that the learning algorithm will work through the entire training dataset. One epoch means that each sample in the training dataset has had an opportunity to update the internal model parameters. An epoch is comprised of one or more batches. In this model we are implementing 500 Epochs with a batch of 32.

6.6 Results

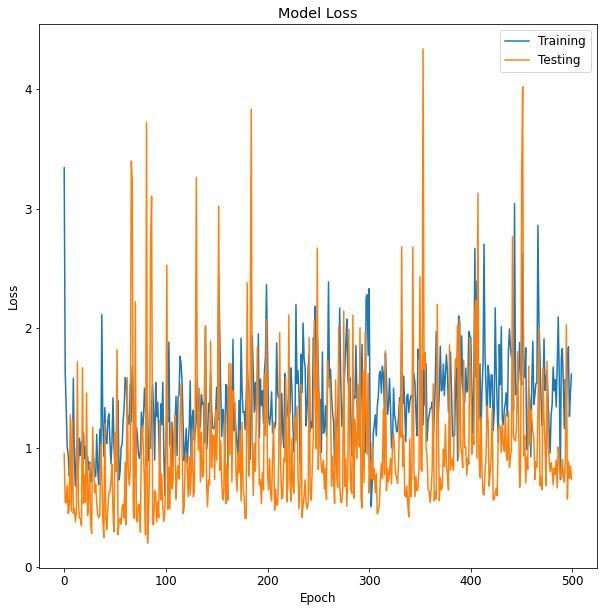
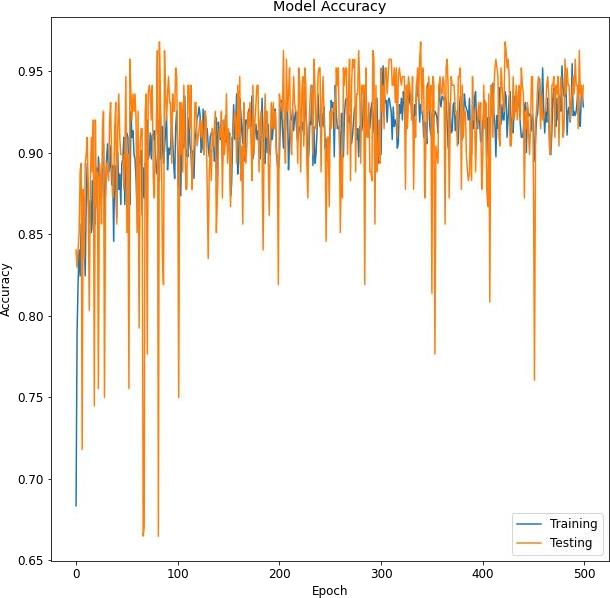


Fig 6.5.1 Inception V3 Plots

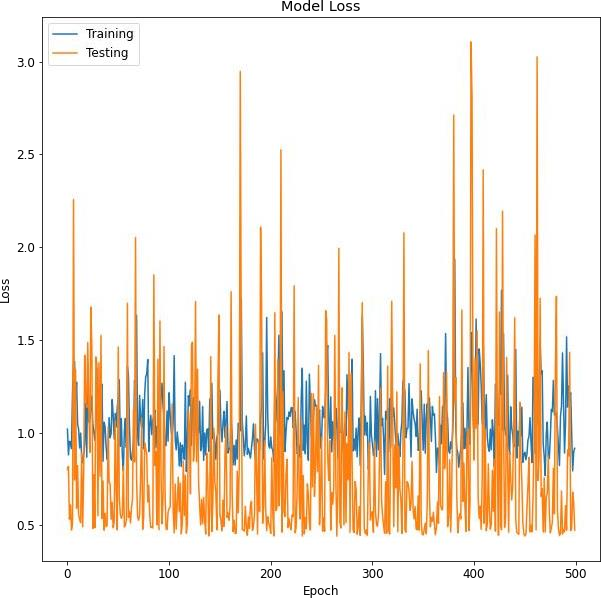
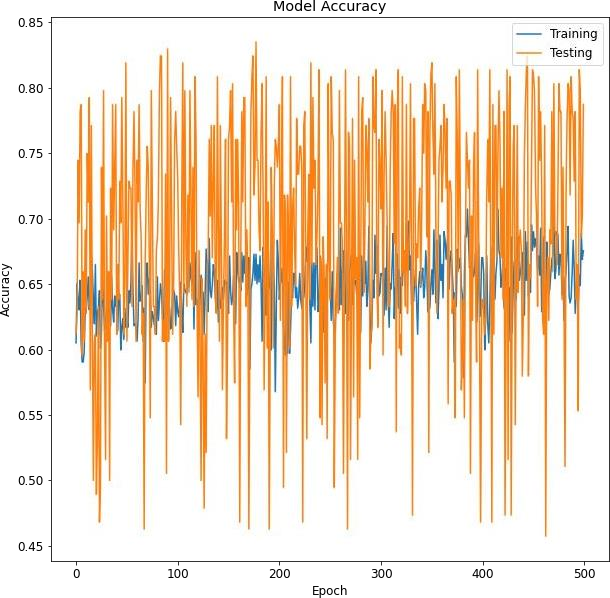


Fig 6.5.2:-ResNet 50 Plots

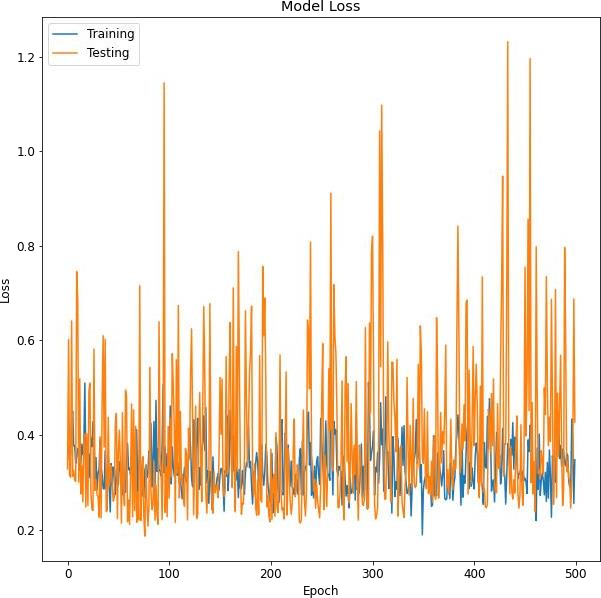
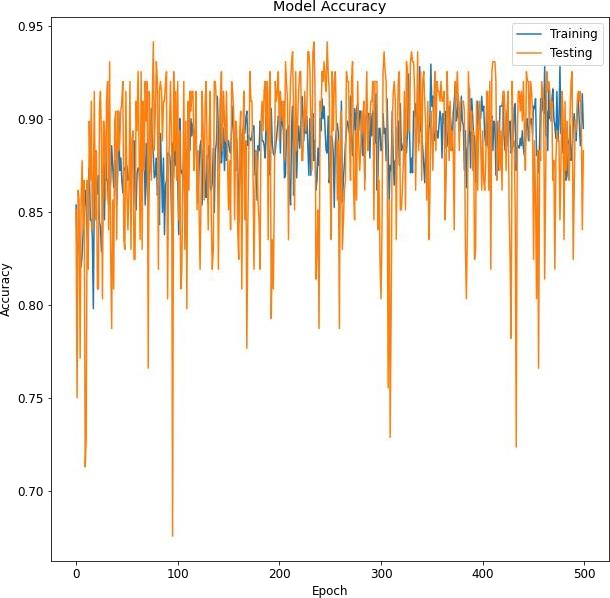


Fig 6.5.3:- Vgg16 Plots

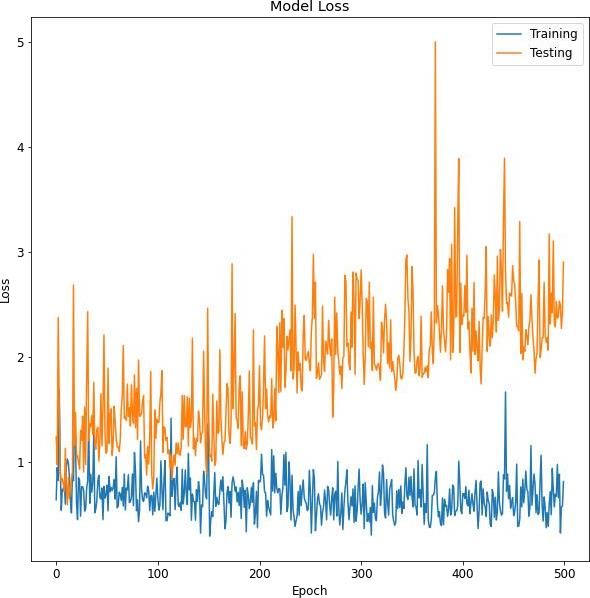
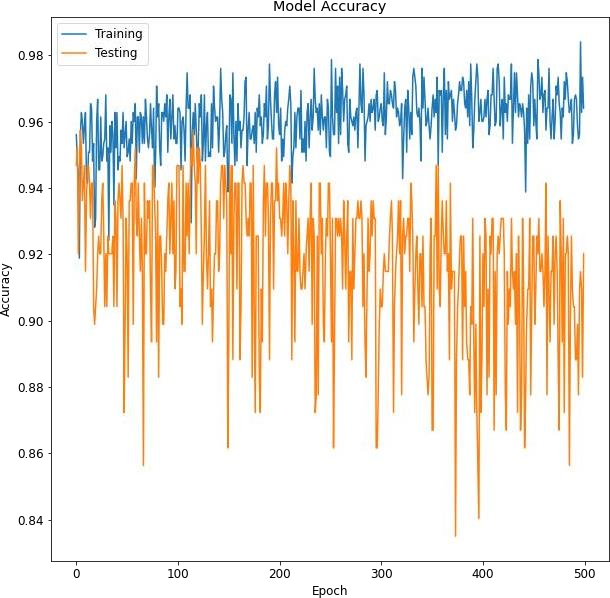


Fig 6.5.4 :- Xception Plots

This deep learning training history plot showing accuracy and loss curves demonstrates that our model is not overfitting despite limited COVID-19 X-ray & CT-Scan training data used in our Keras/TensorFlow model.

Being able to accurately detect COVID-19 with 100% accuracy is great; however, our true negative rate is a bit concerning — we don’t want to classify someone as “COVID-19 negative” when they are “COVID-19 positive”. In fact, the last thing we want to do is tell a patient they are COVID-19 negative, and then have them go home and infect their family and friends; thereby transmitting the disease further.

We also want to be really careful with our false positive rate — we don’t want to mistakenly classify someone as “COVID-19 positive”, quarantine them with other COVID-19 positive patients, and then infect a person who never actually had the virus.

Balancing sensitivity and specificity is incredibly challenging when it comes to medical applications, especially infectious diseases that can be rapidly transmitted, such as COVID- 19. As you can see from the results above, our automatic COVID-19 detector is obtaining ~90-92% accuracy on our sample dataset based solely on X-ray images — no other data, including geographical location, population density, etc. was used to train this model. We are also obtaining 100% sensitivity and 92% specificity implying that:

Of patients that do have COVID-19 (i.e., true positives), we could accurately identify them as “COVID-19 positive” 100% of the time using our model.

Of patients that do not have COVID-19 (i.e., true negatives), we could accurately identify them as “COVID-19 negative” only 92% of the time using our model*.*

The outputs in the web portal are as follows:

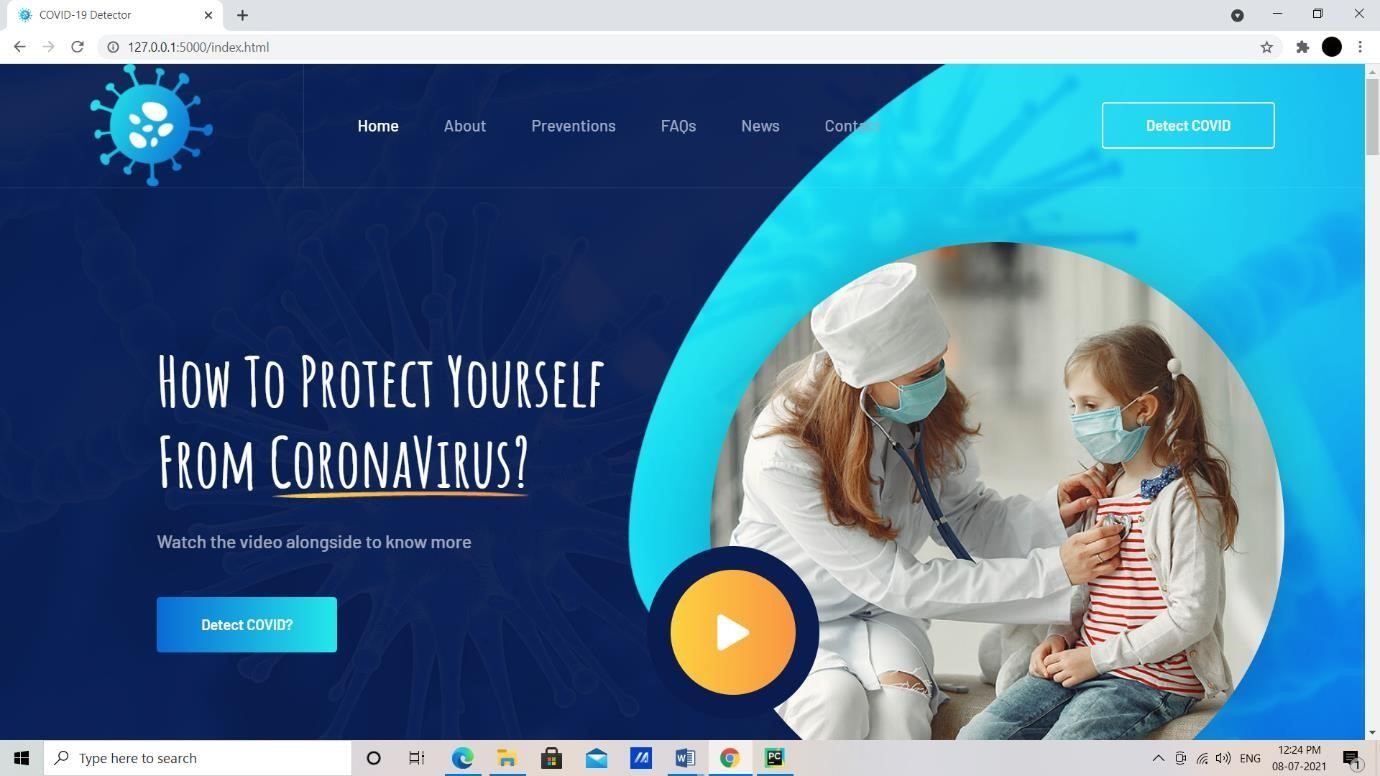


Fig 6.6 :- Opening page of the portal

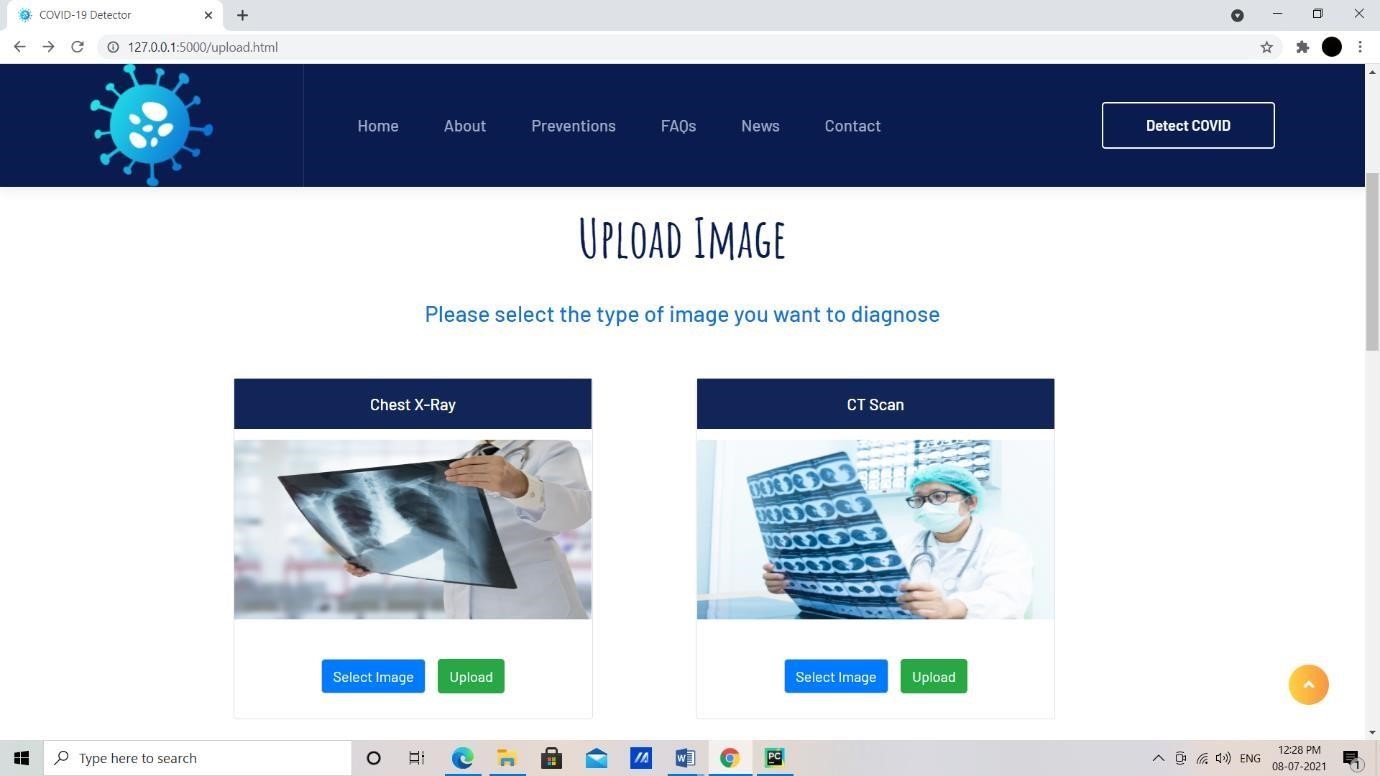
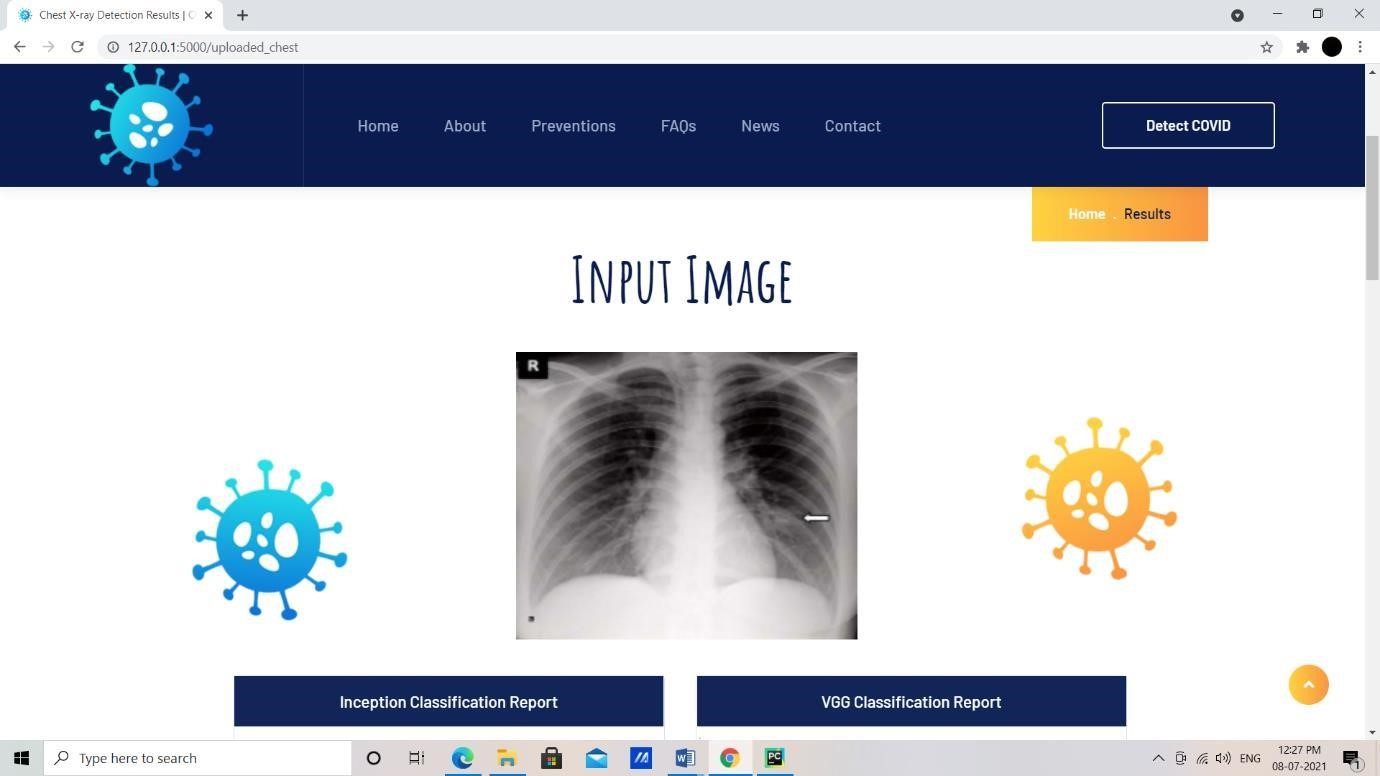


Fig 6.7 :- Uploading X-ray/CT-scan image page

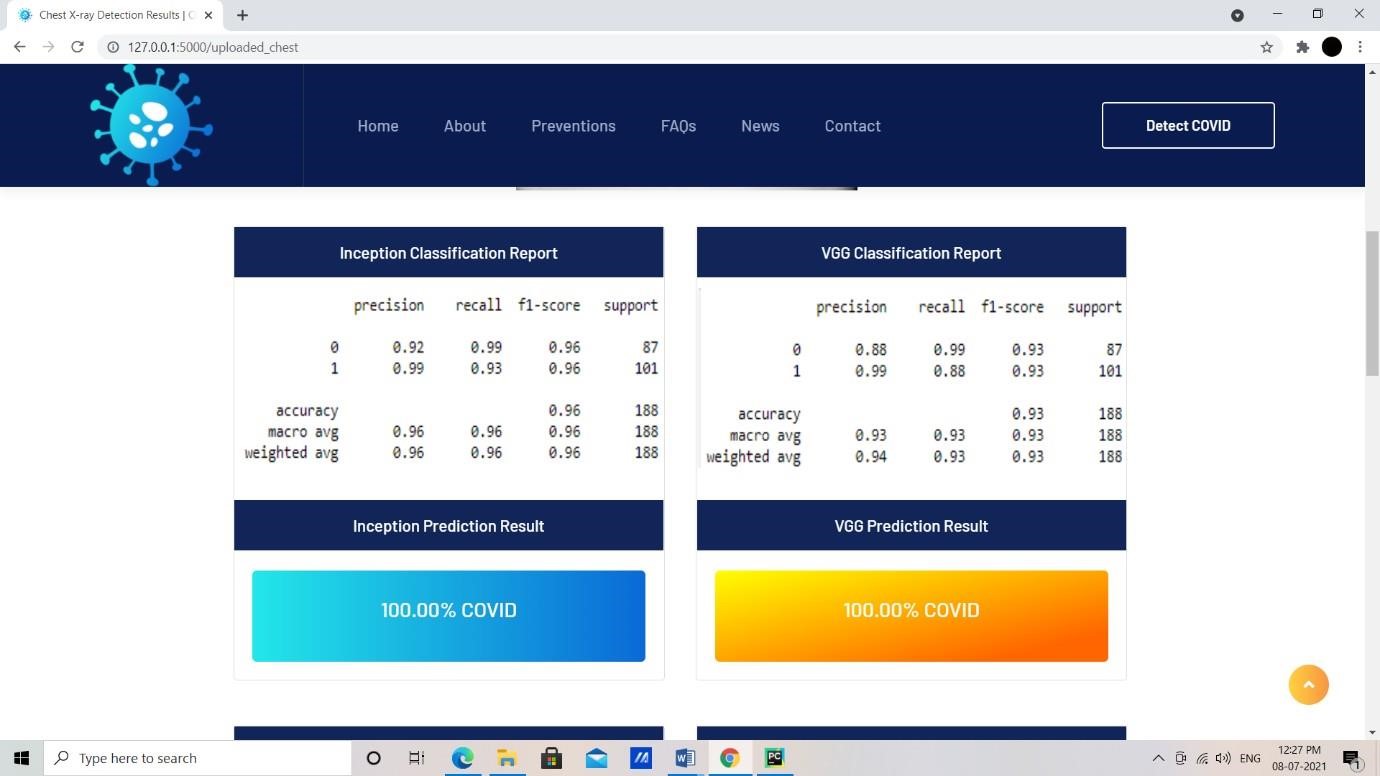


Fig 6.8 :- final result page

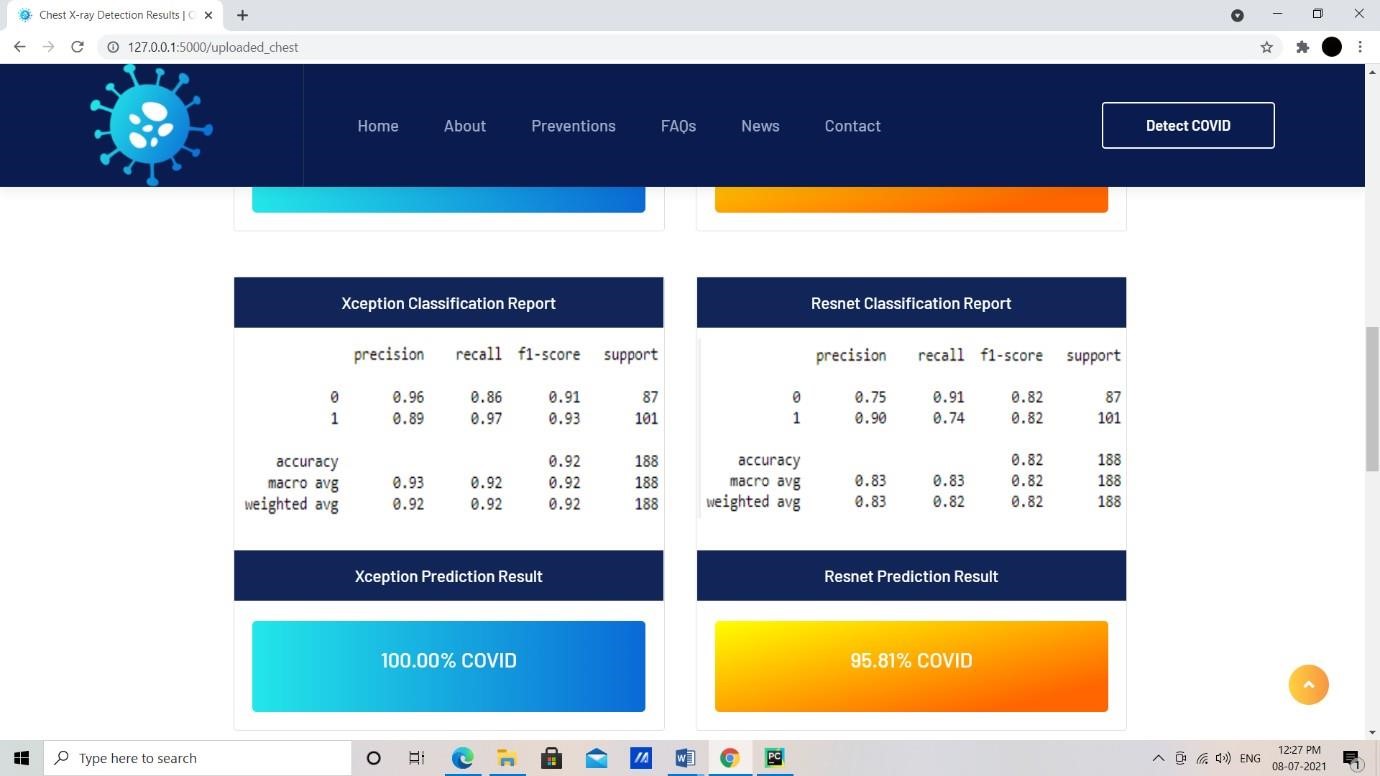


Fig 6.8.1 :- final result page

**CHAPTER 7 CONCLUSION**

In conclusion, I would like to throw light on the fact that the analysis has been done on a limited dataset and that the results are preliminary. Medical validations have not been done on the approach and hence the results might differ from those observed in practical use cases.

In future, I plan to improve the performance of the models by training them on more images and possibly including other factors like age, nationality, gender

**CHAPTER 8 BIBILOGRAPHY**

1. “Early detection of Corona virus cases using X-rays” submitted by Khair Ahammed, Md. Shahriare, Mohammad Zoynul Abedin, Auhidur Rahaman, Sheikh Mohammed at MedRixv submitted on June 8,2020.
2. “Advancements in Image Classification Using CNN” by Farhana Sultana, Abu

Sufian, Paramartha Dutta at 2018 Fourth International Conference on Research in Computational Intelligence and Communication Networks.

1. “Image Classification using CNN” by Muthukrishnan Ramprasath, M.Vijay Anand ,

Shanmugasundaram Hariharan at International journal of Pure and Applied Mathematics, 2018.

1. “Automated detection of COVID-19 cases using DNN with x-ray images” by Tulin

Ozturk , Muhammed Talo , Eylul Azra Yildirim , Ulas Baran Baloglu , Ozal Yildirim, U.

Rajendra Acharya at Elsevier Journal, 2020.

1. “ImageNet Classification with Deep Convolutional Neural Networks” by Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton at ImageNet, 2019.
2. “Deep Convolutional Neural Networks for Chest Diseases Detection” by Rahib H.

Abiyev and Mohammad Khaleel Sallam Ma’aitah at Hindawi Journal of Healthcare Education, 2018.