

PROFESSIONAL TRAINING REPORT
At
SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering Degree in

Information Technology
By

YALAMANCHILI VENKATA SAI
(REG.NO37120085)



DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF COMPUTING

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI – 600119.

JANUARY 2021

DECLARATION

I **YALAMANCHILI VENKATA SAI** (Reg. No. 37120085) hereby declare that the Project Report entitled “FACE MASK DETECTION” done by me under the guidance of **Ms JITHINA JOSE M.E(Ph.D)**, is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology degree in Information Technology.

DATE :

PLACE :

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management Of SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D, Dean**, School of Computing , **Dr.SUBHASHINI M.E., Ph.D. ,** Head of the Department of Information and Technology for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Ms. JITHINA JOSE M.E., Ph.D.,**for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Information Technology** who were helpful in many ways for the completion of the project.

ABSTRACT

Face Mask Detection system built with OpenCV, Keras/TensorFlow using Deep Learning and Computer Vision concepts in order to detect face masks in static images as well as in real-time video streams. Our face mask detector didn't use any morphed masked images dataset. The model is accurate, and since we used the MobileNetV2 architecture, it's also computationally efficient and thus making it easier to deploy the model to embedded systems (Raspberry Pi, Google Coral, etc.). This system can therefore be used in real-time applications which require face-mask detection for safety purposes due to the outbreak of Covid-19. This project can be integrated with embedded systems for application in airports, railway stations, offices, schools, and public places to ensure that public safety guidelines are followed.

TABLE OF CONTENT

CHAPTER NO	TITLE	PAGENO
	ABSTRACT	vi
	LIST OF FIGURES	Vii
1.	INTRODUCTION	1
	1.1 FEATURES OF PYTHON	1
	1.2FEATURES OF MACHINE LEARNING	2
	1.2.1TYPES OF MACHINE LEARNING	3
	1.3FEATURES OF DEEP LEARNING	
	1.3.1 ACHITECTURE OF DEEP LEARNING	
	1.3.2APPLICATINS OF DEEP LEARNING	
	1.4 OUTLINE OF THE PROJECT	5
	1.5LITERATURE REVIEW	
2.	AIM AND SCOPE	8
	2.1 AIM	9
	2.2INPUTS AND OUTPUTS	
	2.2 SCOPE	9
3.	EXPERIMENTAL OR MATERIALS AND	10

	METHODS ALGORITHMS USED	
	3.1 ALGORITHMS USED	10
4.	REQUIREMENT SPECIFICATION	12
	4.1 SOFTWARE REQUIREMENTS	12
	4.2 HARDWARE REQUIREMENTS	12
5.	PROJECT DESCRIPTION	14
	5.1 FLOW CHART	14
	5.2 ALGORITHMS	18
6.	RESULTS AND DISCUSSION	20
	6.1 RESULTS	20
	6.1.1 SCREENSHOTS	20
7.	CONCLUSION AND FUTURE WORK	33
	7.1 CONCLUSION	33
	7.2 FUTURE WORK	33
	REFERENCES	33
	SOURCE CODE	35

LIST OF FIGURES

FIGURE NO	FIGURE NAME	PAGE NO
1.1	SUPERVISED MACHINE LEARNING	4
1.2	ARCHITECTURE OF DEEP LEARNING	10
5.1	USE CASE DIAGRAM	13
5.2	SEQUENCE DIAGRAM	14
5.3	ACTIVITY DIAGRAM	15
5.4	COLLABORATION DIAGRAM	16
5.5	CLASS DIAGRAM	16
6.1	DATASET RESULT	19
6.2	CURVE PLOT	19
6.3	SAMPLE IMAGE	19
6.4	SAMPLE IMAGE WITH MASK	20
6.5	SAMPLE IMAGE WITHOUT MASK	20
6.6	GRAPH OF TARGET	21

6.7	WITH MASK USING WEB CAM	21
6.8	WITHOUT MASK USING WEB CAM	22

CHAPTER 1

INTRODUCTION

Python is a widely used general-purpose, high level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code. Python is a programming language that lets you work quickly and integrate systems more efficiently.

1.1 FEATURES OF PYTHON

Python is a dynamic, high level, free open source and interpreted programming language. It supports object-oriented programming as well as procedural oriented programming. In Python, we don't need to declare the type of variable because it is a dynamic typed language.

1.Easy to code: Python is high level programming language .Python is very easy to learn language as compared to other language like c, c#, java script, java etc. It is very easy to code in python language and anybody can learn python basic in few hours or days .It is also developer-friendly language.

2.Free and Open Source: Python language is freely available at official website and you can download it from the given download link below click on the Download python keyword. Download Python Since, it is open-source, this means that source code is also available to the public. So you can download it as, use it as well as share it.

3.Object-OrientedLanguage: One of the key features of python is Object-Oriented programming. Python supports object oriented language and concepts of classes, objects encapsulation etc.

4.GUI Programming Support: Graphical Users interfaces can be made using a module such as PyQt5, PyQt4, wx Python or Tk in python. PyQt5 is the most popular option for creating graphical apps with Python.

5. High-Level Language: Python is a high-level language. When we write programs in python, we do not need to remember the system architecture, nor do we need to manage the memory.

6. Extensible feature: Python is a Extensible language. we can write our some python code into c or c++ language and also we can compile that code in c/c++ language.

7. Python is Portable language: Python language is also a portable language. for example, if we have python code for windows and if we want to run this code on other platform such as Linux, Unix and Mac then we do not need to change it, we can run this code on any platform.

8. Python is Integrated language: Python is also an Integrated language because we can easily integrated python with other language like c, c++ etc.

9. Interpreted Language: Python is an Interpreted Language. because python code is executed line by line at a time. like other language c, c++, java etc there is no need to compile python code this makes it easier to debug our code. The source code of python is converted into an immediate form called bytecode.

10. Large Standard Library: Python has a large standard library which provides rich set of module and functions so you do not have to write your own code for every single thing. There are many libraries present in python for such as regular expressions, unit-testing, web browsers etc.

11. Dynamically Typed Language: Python is dynamically-typed language. That means the type (for example- int, double, long etc) for a variable is decided at run time not in advance. Because of this feature we don't need to specify the type of variable

1.2 FEATURES OF MACHINE LEARNING

Introduction to machine learning: Machine learning is a type of artificial intelligence (AI) that provides computers with the ability to learn without being explicitly programmed. Machine learning focuses on the development of Computer Programs that can change when exposed to new data. In this article, we'll see basics of Machine Learning, and implementation of a simple machine learning algorithm using python.

Setting up the environment: Python community has developed many modules to help programmers implement machine learning. In this article, we will be using numpy, scipy and scikit-learn modules. We can install them using cmd command: a better option would be downloading miniconda or anaconda packages for python, which come prebundled with these packages. Follow the instructions given [here](#) to use anaconda

Machine Learning overview:

Machine learning involves computer to get trained using a given data set, and use this training to predict the properties of a given new data. For example, we can train computer by feeding it 1000 images of cats and 1000 more images which are not of a cat, and tell each time to computer whether a picture is cat or not. Then if we show the computer a new image, then from the above training, computer should be able to tell whether this new image is a cat or not. Process of training and prediction involves use of specialized algorithms. We feed the training data to an algorithm, and the algorithm uses this training data to give predictions on a new test data. One such algorithm is K Nearest Neighbor classification (KNN classification). It takes a test data, and finds k nearest data values to this data from test data set. Then it selects the neighbor of maximum frequency and gives its properties as the prediction result.

1.2.1 Types of Machine learning

a)Supervised Learning : Supervised learning is when the model is getting trained on a labelled dataset. **Labelled** dataset is one which have both input and output parameters. In this type of learning both training and validation datasets are labelled as shown in the figures.

User ID	Gender	Age	Salary	Purchased	Temperature	Pressure	Relative Humidity	Wind Direction	Wind Speed
15624510	Male	19	19000	0	10.69261758	986.882019	54.19337313	195.7150879	3.278597116
15810944	Male	35	20000	1	13.59184184	987.8729248	48.0648859	189.2951202	2.909167767
15668575	Female	26	43000	0	17.70494885	988.1119385	39.11965597	192.9273834	2.973036289
15603246	Female	27	57000	0	20.95430404	987.8500366	30.66273218	202.0752869	2.965289593
15804002	Male	19	76000	1	22.9278274	987.2833862	26.06723423	210.6589203	2.798230886
15728773	Male	27	58000	1	24.04233986	986.2907104	23.46918024	221.1188507	2.627005816
15598044	Female	27	84000	0	24.41475295	985.2338867	22.25082295	233.7911987	2.448749781
15694829	Female	32	150000	1	23.93361956	984.8914795	22.35178837	244.3504333	2.454271793
15600575	Male	25	33000	1	22.68800023	984.8461304	23.7538641	253.0864716	2.418341875
15727311	Female	35	65000	0	20.56425726	984.8380737	27.07867944	264.5071106	2.318677425
15570769	Female	26	80000	1	17.76400389	985.4262085	33.54900114	280.7827454	2.343950987
15606274	Female	26	52000	0	11.25680746	988.9386597	53.74139903	68.15406036	1.650191426
15746139	Male	20	86000	1	14.37810685	989.6819458	40.70884681	72.62069702	1.553469896
15704987	Male	32	18000	0	18.45114201	990.2960205	30.85038484	71.70604706	1.005017161
15628972	Male	18	82000	0	22.54895853	989.9562988	22.81738811	44.66042709	0.264133632
15697686	Male	29	80000	0	24.23155922	988.796875	19.74790765	318.3214111	0.329656571
15733883	Male	47	25000	1					

Figure A: CLASSIFICATION

Figure B: REGRESSION

Fig 1.1 Supervised Machine Learning

Types of Supervised Learning:

Classification: It is a Supervised Learning task where output is having defined labels(discrete value). For example in above Figure A, Output – Purchased has defined labels i.e. 0 or 1 ; 1 means the customer will purchase and 0 means that customer won't purchase. The goal here is to predict discrete values belonging to a particular class and evaluate on the basis of accuracy.

Regression: It is a Supervised Learning task where output is having continuous value. Example in above Figure B, Output – Wind Speed is not having any discrete value but is continuous in the particular range. The goal here is to predict a value as much closer to actual output value as our model can and then evaluation is done by calculating error value. The smaller the error the greater the accuracy of our regression model.

b)Unsupervised Learning: Unsupervised learning is the training of machine using information that is neither classified nor labeled 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. 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 unlabeled. Thus the machine has no idea about the features of dogs and cat so we can't categorize it in dogs and cats. But it can categorize them according to their similarities, patterns, and differences i.e., we can easily categorize the above picture into two parts. First part may contain all pics having dogs in it and second part may contain all pics having cats in it. Here you didn't learn anything before, means no training data or examples.

Unsupervised learning classified into two categories of algorithms:

Clustering: A clustering problem is where you want to discover the inherent groupings in the data, such as grouping customers by purchasing behavior.

Association: An association rule learning problem is where you want to discover rules that describe large portions of your data, such as people that buy X also tend to buy Y.

1.3 FEATURES OF DEEP LEARNING

Introduction to Deep Learning : Deep learning is a branch of machine learning which is completely based on artificial neural networks, as neural network is going to mimic the human brain so deep learning is also a kind of mimic of human brain. In deep learning, we don't need to explicitly program everything. The concept of deep learning is not new. It has been around for a couple of years now. It's on hype nowadays because earlier we did not have that much processing power and a lot of data. As in the last 20 years, the processing power increases exponentially, deep learning and machine learning came in the picture. A formal definition of deep learning is- neurons.

1.3.1 Architectures of Deep learning :

a) Deep Neural Network – It is a neural network with a certain level of complexity (having multiple hidden layers in between input and output layers). They are capable of modeling and processing non-linear relationships.

b) Deep Belief Network(DBN) – It is a class of Deep Neural Network. It is multi-layer belief-network.

Steps

- a. Learn a layer of features from visible units using Contrastive Divergence algorithm.
- b. Treat activations of previously trained features as visible units and then learn features.
- c. Finally, the whole DBN is trained when the learning for the final hidden layer is achieved.

c) Recurrent (perform same task for every element of a sequence) Neural Network – Allows for parallel and sequential computation. Similar to the human brain (large feedback network of connected neurons). They are able to remember important things about the input they received and hence enables them to be more precise.

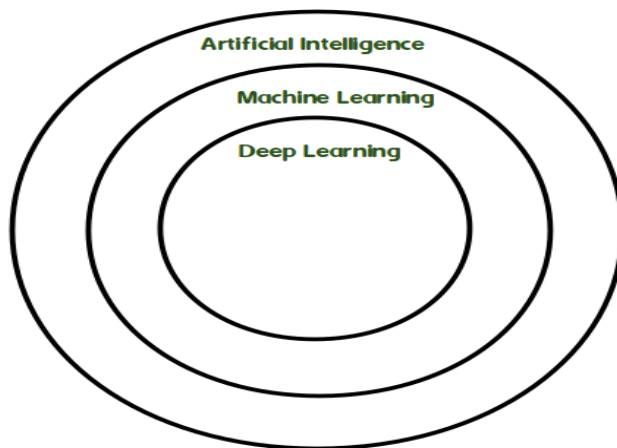


Fig 1.2 Architecture of deep learning.

1.3.2 Applications of Deep learning :

- a) **Automatic Text Generation** – Corpus of text is learned and from this model new text is generated, word-by-word or character-by-character. Then this model is capable of learning how to spell, punctuate, form sentences, or it may even capture the style.
- b) **Healthcare** – Helps in diagnosing various diseases and treating it.
- c) **Automatic Machine Translation** – Certain words, sentences or phrases in one language is transformed into another language (Deep Learning is achieving top results in the areas of text, images).
- d) **Image Recognition** – Recognizes and identifies peoples and objects in images as well as to understand content and context. This area is already being used in Gaming, Retail, Tourism, etc.
- e) **Predicting Earthquakes** – Teaches a computer to perform viscoelastic computations which are used in predicting earthquakes.

1.4 OUTLINE OF PROJECT

The main goal of the project is to find whether the person is wearing mask or not by training machine using predefined dataset of images using OpenCv, tensorflow, keras and deep learning algorithms.

my face mask detector didn't uses any morphed masked images dataset. The model is accurate, and since we used the MobileNetV2 architecture, it's also computationally efficient and thus making it easier to deploy the model to embedded systems (Raspberry Pi, Google Coral, etc.).

1.5 LITERATURE REVIEW

As the project determines whether the person is wearing mask or not based on previous data provided in dataset. we use deep learning learning techniques and tensorflow,keras to train data for future analysis of mask detection for a person .

STUDY OF THE SYSTEM

1.Numpy

2.Pandas

3.Matplotlib

4.Scikit –learn

1.Numpy: is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, Numpy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using Numpy which allows Numpy to seamlessly and speedily integrate with a wide variety of databases.

2.Pandas: Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load,

prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

3. Matplotlib: Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and [IPython](#) shells, the [Jupyter](#) notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery. For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

4. Scikit – learn: Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use. The library is built upon the SciPy (Scientific Python) that must be installed before you can use scikit-learn. This stack that includes:

NumPy: Base n-dimensional array package

SciPy: Fundamental library for scientific computing

Matplotlib: Comprehensive 2D/3D plotting

IPython: Enhanced interactive console

Sympy: Symbolic mathematics

Pandas: Data structures and analysis

Extensions or modules for SciPy are conventionally named SciKits. As such, the module provides learning algorithms and is named scikit-learn.

CHAPTER 2

AIM AND SCOPE

2.1 AIM

The main aim of the project is to find whether the person is wearing mask or not using Deep learning techniques.

2.2 INPUT AND OUTPUTS

The following some are the projects inputs and outputs.

Inputs:

Importing the all required packages like Opencv, tensorflow,keras, numpy, pandas, matplotlib, scikit – learn and required deep learning algorithms packages .

This dataset consists of **3835 images** belonging to two classes:

- **with_mask: 1916 images**
- **without_mask: 1919 images**

The images used were real images of faces wearing masks. The images were collected from the following sources:

Setting the dimensions of visualization graph.

Outputs:

preprocessing the importing data frame for inputing nulls with the related information.

All are displaying cleaned outputs.

After applying deep learning algorithms,tensorflow,keras it will give good results and visualization plots.

Scope:

In future due to COVID-19 effects all over the world corona effected patients increasing day by day. My main theme of the project is to find whether a person is wearing face mask or not through deep learning algorithms.

CHAPTER 3

EXPERIMENTAL OR MATERIALS AND METHODS ALGORITHMS USED

3.1 ALGORITHMS USED

a)OpenCv: OpenCV is a huge open-source library for computer vision, machine learning, and image processing. OpenCV supports a wide variety of programming languages like Python, C++, Java, etc. It can process images and videos to identify objects, faces, or even the handwriting of a human. When it is integrated with various libraries, such as Numpy which is a highly optimized library for numerical operations, then the number of weapons increases in your Arsenal i.e whatever operations one can do in Numpy can be combined with OpenCV.

Face Recognition: Face recognition is an easy task for humans. Experiments in [ui06] have shown, that even one to three day old babies are able to distinguish between known faces. So how hard could it be for a computer? It turns out we know little about human recognition to date. Are inner features (eyes, nose, mouth) or outer features (head shape, hairline) used for a successful face recognition? How do we analyze an image and how does the brain encode it? It was shown by David Hubel and Torsten Wiesel, that our brain has specialized nerve cells responding to specific local features of a scene, such as lines, edges, angles or movement. Since we don't see the world as scattered pieces, our visual cortex must somehow combine the different sources of information into useful patterns. Automatic face recognition is all about extracting those meaningful features from an image, putting them into a useful representation and performing some kind of classification on them.

b)Caffe Based face-Detector: Caffe is a deep learning framework made with expression, speed, and modularity in mind. It is developed by Berkely AI Research (BAIR) and by community contributors. Yangqing Jia creates the project during his PhD at UC Berkely.

Advantages:

1.Expressive architecture encourages application and innovation. Models and optimization are defined by configuration without hard-coding. Switch between CPU and GPU by setting a single flag to train on a GPU machine then deploy to commodity clusters or mobile devices.

2.Extensible code fosters active development. In Caffe's first year, it has been forked by over 1,000 developers and had many significant changes contributed back. Thanks to these contributors the framework tracks the state-of-the-art in both code and models.

3.Speed makes Caffe perfect for research experiments and industry deployment. Caffe can process over 60M images per day with a single NVIDIA K40 GPU*. That's 1 ms/image for inference and 4 ms/image for learning and more recent library versions and hardware are faster still. We believe that Caffe is among the fastest convnet implementations available.

c)Keras: Keras is a deep learning API written in Python, running on top of the machine learning platform TensorFlow. It was developed with a focus on enabling fast experimentation. Being able to go from idea to result as fast as possible is key to doing good research.

d)TensorFlow 2.0: TensorFlow 2.0 is an end-to-end, open-source machine learning platform. You can think of it as an infrastructure layer for differentiable programming. It combines four key abilities:

- a) Efficiently executing low-level tensor operations on CPU, GPU, or TPU.
- b) Computing the gradient of arbitrary differentiable expressions.
- c) Scaling computation to many devices (e.g. the Summit supercomputer at Oak Ridge National Lab, which spans 27,000 GPUs).
- d) Exporting programs ("graphs") to external runtimes such as servers, browsers, mobile and embedded devices.

Keras is the high-level API of TensorFlow 2.0: an approachable, highly-productive interface for solving machine learning problems, with a focus on modern deep learning. It provides essential abstractions and building blocks for developing and shipping machine learning solutions with high iteration velocity.

Keras empowers engineers and researchers to take full advantage of the scalability and cross-platform capabilities of TensorFlow 2.0: you can run Keras on TPU or on large clusters of GPUs, and you can export your Keras models to run in the browser or on a mobile device.

e)MobileNetV2: that improves the state of the art performance of mobile models on multiple tasks and benchmarks as well as across a spectrum of different model sizes. We also describe efficient ways of applying these mobile models to object detection in a novel framework we call SSDLite. Additionally, we demonstrate how to build mobile semantic segmentation models through a reduced form of DeepLabv3 which we call MobileDeepLabv3.

The MobileNetV2 architecture is based on an inverted residual structure where the input and output of the residual block are thin bottleneck layers opposite to traditional residual models which use expanded representations in the input. MobileNetV2 uses lightweight depthwise convolutions to filter features in the intermediate expansion layer. Additionally, we find that it is important to remove non-linearities in the narrow layers in order to maintain representational power. We demonstrate that this improves performance and provide an intuition that led to this design. Finally, our approach allows decoupling of the input/output domains from the expressiveness of the transformation, which provides a convenient framework for further analysis. We measure our performance on Imagenet classification, COCO object detection, VOC image segmentation. We evaluate the trade-offs between accuracy, and number of operations measured by multiply-adds (MAdd), as well as the number of parameters.

CHAPTER 4

REQUIREMENT SPECIFICATION

4.1 INTRODUCTION

To be used efficiently, all computer software needs certain hardware components or other software resources to be present on a computer. These prerequisites are known as computer system requirements and are often used as a guideline as opposed to an absolute rule. Most software defines two sets of system requirements: Minimum and Recommended. With increasing demand for higher processing power and resources in newer versions of software, system requirements tend to increase over time. Industry analysts suggest that this trend plays a bigger part in driving upgrades to existing computer systems than technological advancements.

4.2 SOFTWARE REQUIREMENTS

The software requirements are: Software Requirements deal with definition of software resource requirements and prerequisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or prerequisites are generally not included in the software installation package and need to be installed separately before the software is installed.

Software requirements for present project:

Operating system: - Windows 7/ XP/8/10

Language :- Python, Machine Learning, Deep Learning.

4.3 HARWARE REQUIREMENTS

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in the case of operating systems. An HCL list tested, compatibility and sometimes incompatible hardware devices for a particular operating system or application. The following subsections discuss the various aspects of hardware requirements.

Hardware requirements for this project:

Processor: - Intel dual Core i3/i5/i7 Ram: - 1 GB HARD

Disk: - 80 GB

CHAPTER 5

PROJECT DESCRIPTION

The main goal of the project is to find whether the person is wearing mask or not by training machine using predefined dataset of images using OpenCv, tensorflow, keras and deep learning algorithms.

my face mask detector didn't uses any morphed masked images dataset. The model is accurate, and since we used the MobileNetV2 architecture, it's also computationally efficient and thus making it easier to deploy the model to embedded systems (Raspberry Pi, Google Coral, etc.).

5.1 FLOW CHART

Use Case Diagram:

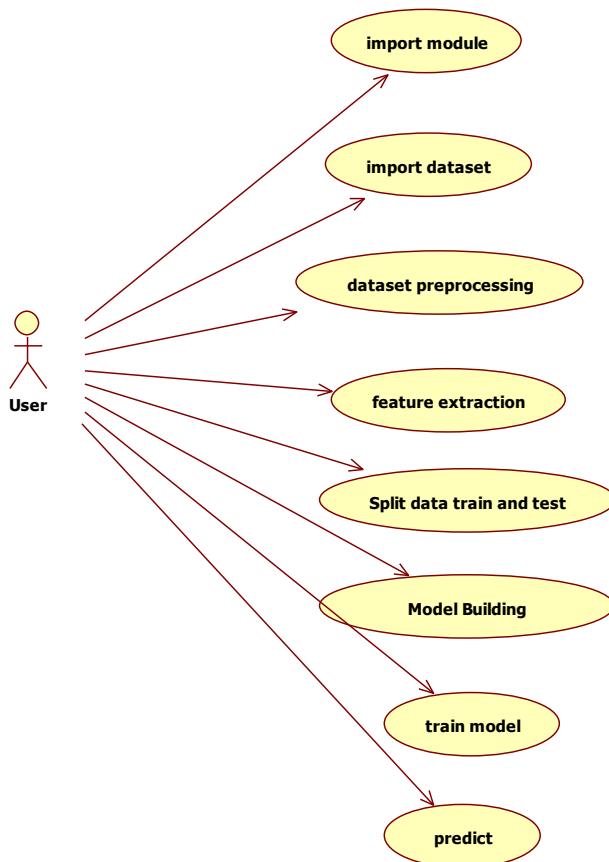


Fig 5.1 Use case diagram

Sequence diagram: Sequence Diagrams Represent the objects participating the interaction horizontally and time vertically. A Use Case is a kind of behavioral classifier that represents a declaration of an offered behavior. Each use case specifies some behavior, possibly including variants that the subject can perform in collaboration with one or more actors. Use cases define the offered behavior of the subject without reference to its internal structure. These behaviors, involving interactions between the actor and the subject, may result in changes to the state of the subject and communications with its environment. A use case can include possible variations of its basic behavior, including exceptional behavior and error handling.

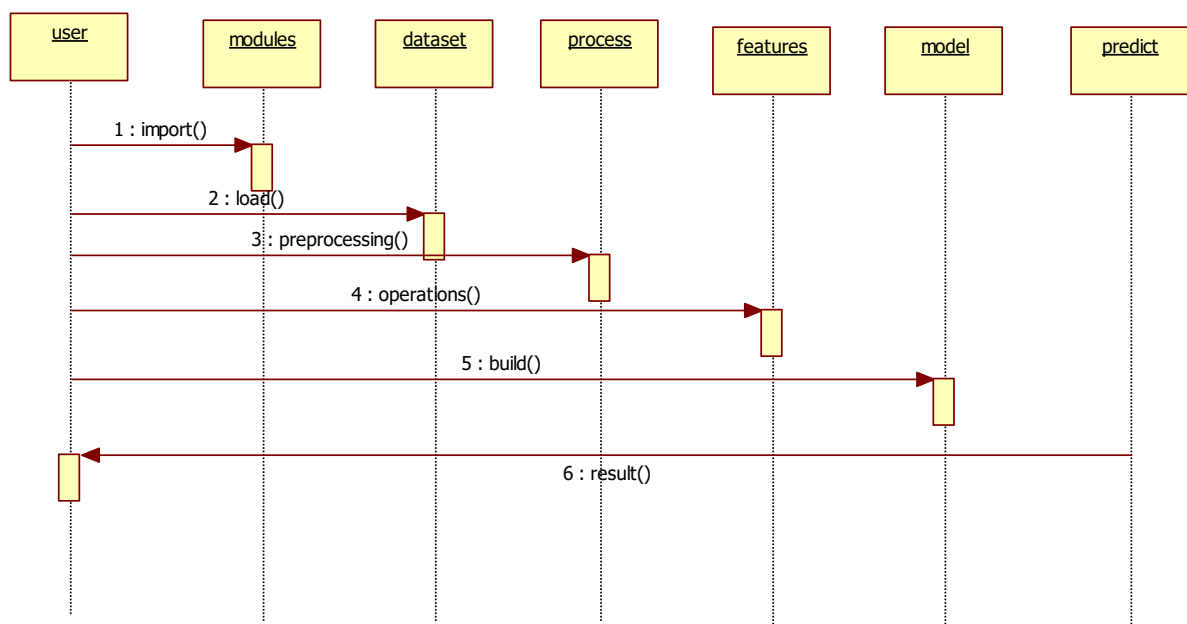


Fig 5.2 Sequence diagram

Activity diagram:

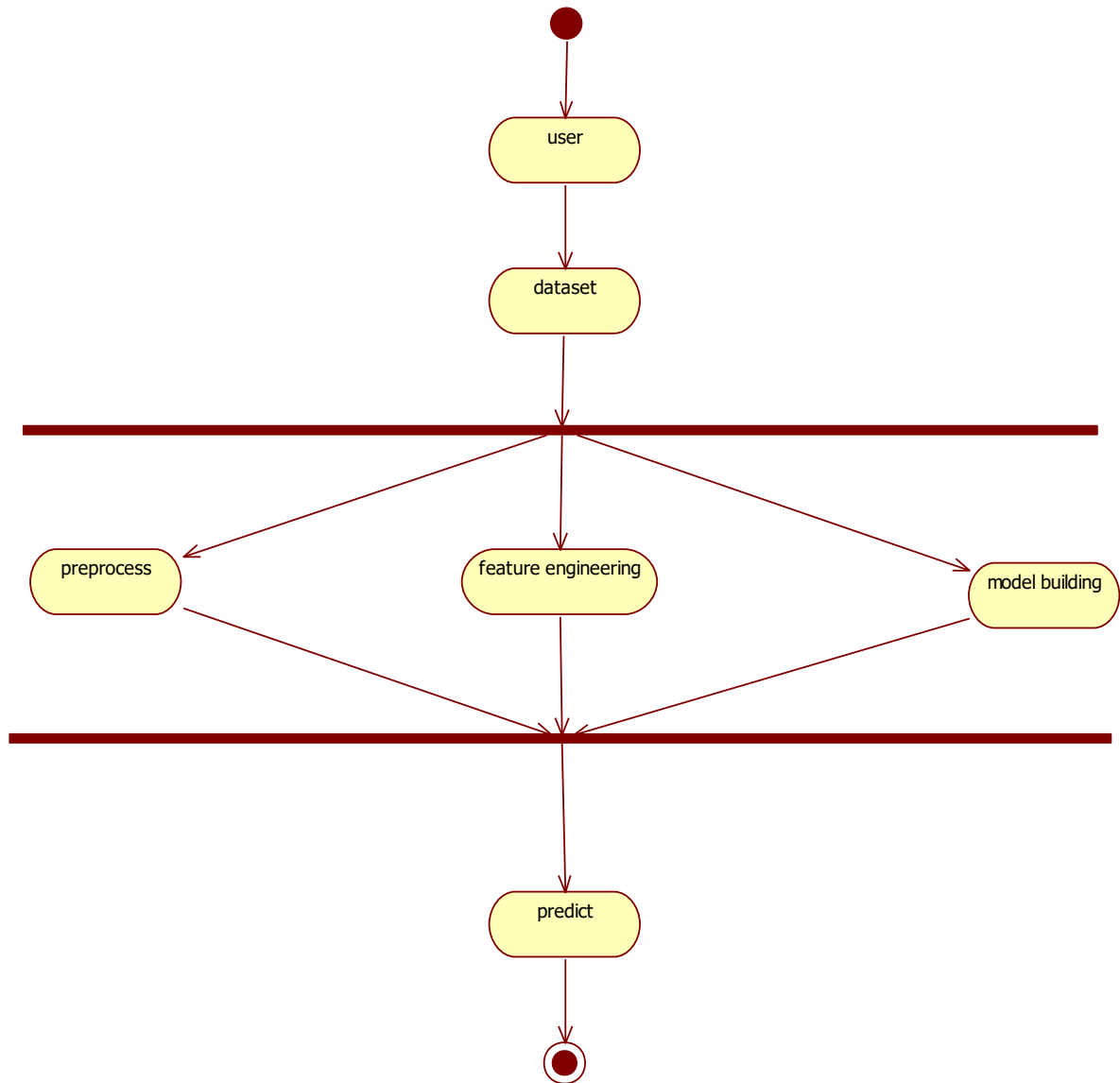


Fig 5.3 Activity diagram

Collaboration Diagram:

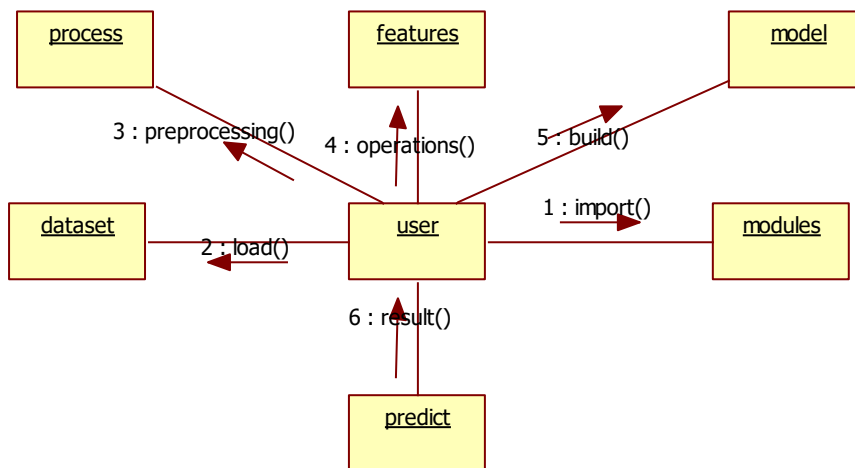


Fig 5.4 Collaboration diagram

Class Diagram:

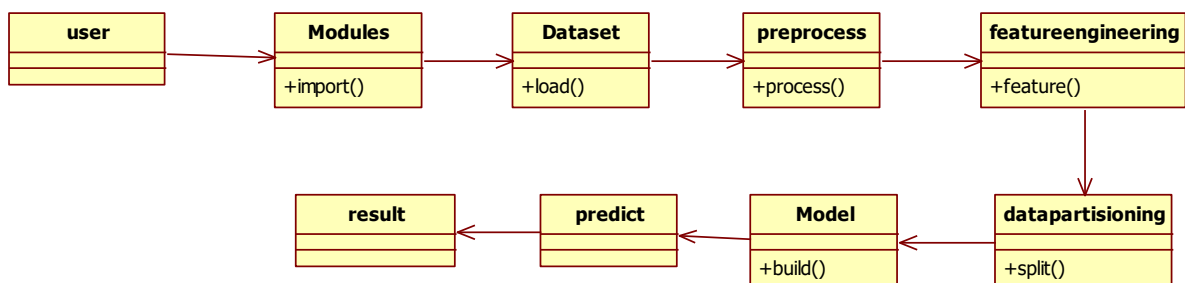


Fig 5.5 Class Diagram

CHAPTER 6

RESULTS AND DISCUSSION

6.1 RESULTS

6.1.1 SCREENSHOTS

Data set Execution:

```
[INFO] evaluating network...
      precision    recall  f1-score   support

 with_mask         0.99      0.86      0.92         383
without_mask         0.88      0.99      0.93         384

   accuracy              0.93         767
  macro avg              0.93      0.93      0.93         767
weighted avg              0.93      0.93      0.93         767

[INFO] saving mask detector model...
dict_keys(['loss', 'accuracy', 'val_loss', 'val_accuracy'])
```

Fig 6.1 Dataset results

My model has given 93% accuracy for Face Mask Detection after training via tensorflow-gpu==2.0.0

curve plot :



Fig 6.2 Curve plot

We got the following accuracy/loss training curve plot.

Image Detection:

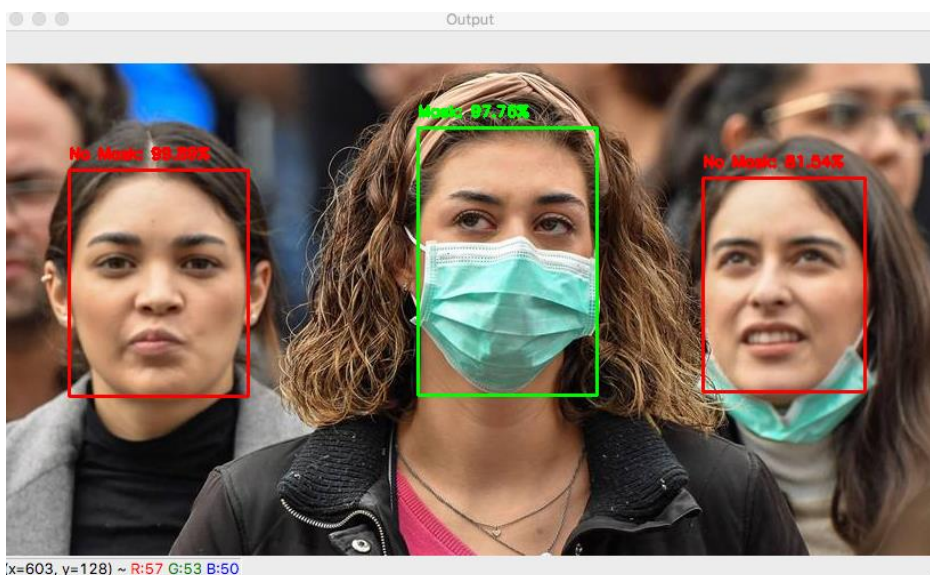


Fig 6.3 sample image.

In this image we can see it is detecting sample output that a person is wearing mask or not and printing accuracy.

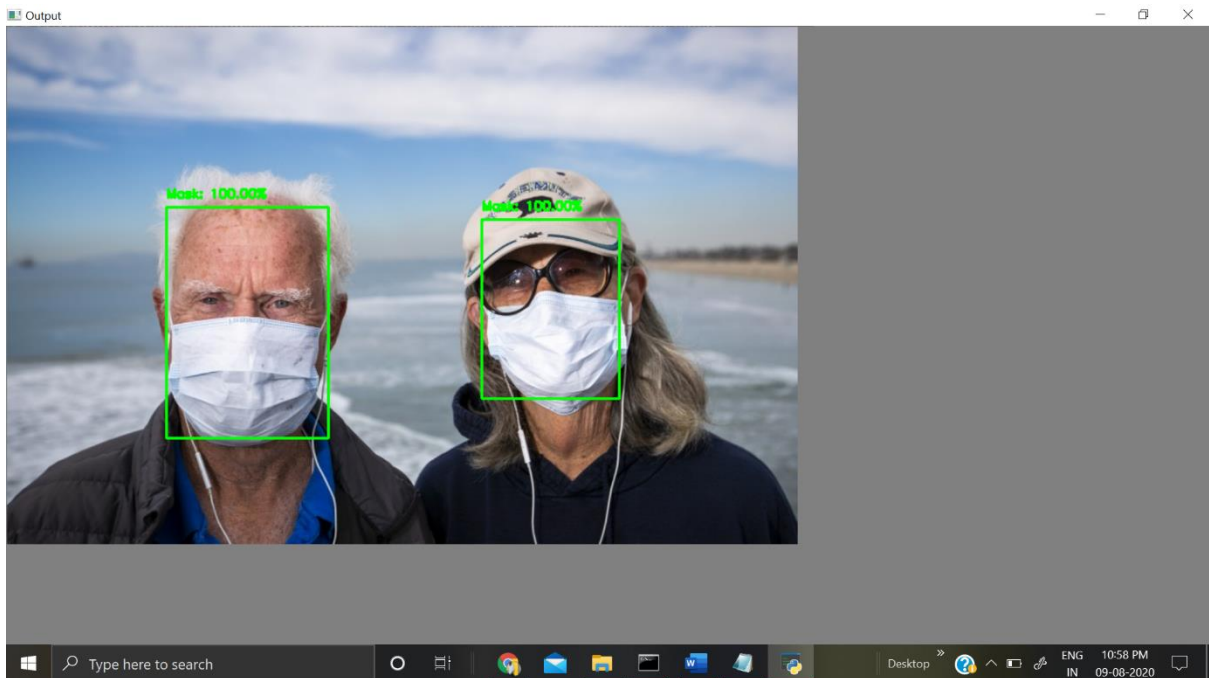


Fig 6.4 sample image mask

Detecting people who is having mask



Fig 6.5 Sample image Without mask

Detecting a person without mask

Using Tkinter and OpenCv



Fig 6.7 With mask using Webcam

This sample image is taken through my webcam it detects face mask in a video.

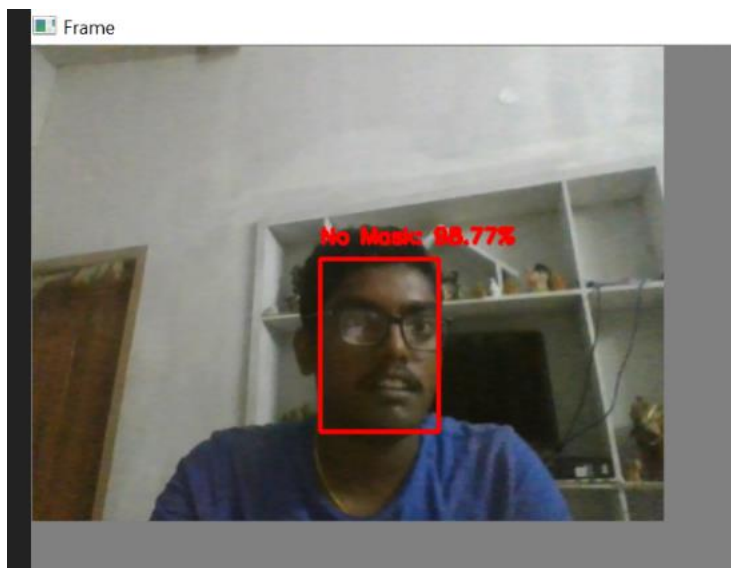


Fig 6.8 Without mask using Webcam

This sample image is taken through my webcam it detects no face mask in a video.

CHAPTER 8

CONCLUSION AND FUTURE WORK

8.1 CONCLUSION:

Face mask detection system uses artificial intelligence and computer vision to detect people who aren't wearing face masks. The app can be connected to any existing or new IP mask detection cameras to detect people without a mask. App users can also add faces and phone numbers to send them an alert in case they are not wearing a mask. It provides real-time insights and generates alerts to the concerned authorities, so that corrective action can be taken.

8.2 FUTURE WORK:

It provides real-time insights and generates alerts to the concerned authorities, so that corrective action can be taken.

References

1. T. Ojala, M. Pietikainen and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 7, pp. 971-987, July 2002.
2. T.-H. Kim, D.-C. Park, D.-M. Woo, T. Jeong and S.-Y. Min, "Multi-class classifier-based adaboost algorithm", Proceedings of the Second Sino-foreign-interchange Conference on Intelligent Science and Intelligent Data Engineering, pp. 122-127, 2012.
3. P. Viola and M. J. Jones, "Robust real-time face detection", Int. J. Comput. Vision, vol. 57, no. 2, pp. 137-154, May 2004.
4. P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features", Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, vol. 1, pp. I-I, Dec 2001.
5. J. Li, J. Zhao, Y. Wei, C. Lang, Y. Li and J. Feng, "Towards real world human parsing: Multiple-human parsing in the wild", CoRR.
6. A. Krizhevsky, I. Sutskever and G. E. Hinton, "Imagenet classification with deep convolutional neural networks", Advances in Neural Information Processing Systems 25, pp. 1097-1105, 2012.
7. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition", CoRR, 2014.

- 8.** C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, et al., "Going deeper with convolutions", 2015.
- 9.** K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition", 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770-778, 2016.
- 10.** K. Li, G. Ding and H. Wang, "L-fcn: A lightweight fully convolutional network for biomedical semantic segmentation", 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pp. 2363-2367, Dec 2018.
- 11.** X. Fu and H. Qu, "Research on semantic segmentation of high-resolution remote sensing image based on full convolutional neural network", 2018 12th International Symposium on Antennas Propagation and EM Theory (ISAPE), pp. 1-4, Dec 2018.
- 12.** S. Kumar, A. Negi, J. N. Singh and H. Verma, "A deep learning for brain tumor mri images semantic segmentation using fcn", 2018 4th International Conference on Computing Communication and Automation (ICCCA), pp. 1-4, Dec 2018.

