##### 

Detecting Face Mask using Artificial Intelligence, Machine Learning and Deep Learning for COVID-19 Prevention

### 

### A Project Work

*Submitted in the partial fulfillment for the award of the degree of*

# BACHELOR OF ENGINEERING

### IN

### AIML

### Submitted by:

|  |  |
| --- | --- |
| NAME OF THE STUDENT | University Roll Number |
| Toshiba Ansari | 20BCS6671 |
| Abhishek Singh | 20BCS6673 |
| Pushp Jain | 20BCS6706 |
| Rahul Goyal | 20BCS6713 |

### Under the Supervision of:

### Mr. Leeza Sharma



# DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING APEX INSTITUE OF TECHNOLOGY

### CHANDIGARH UNIVERSITY, GHARUAN, MOHALI - 140413,

**PUNJAB**

**May & 2022**

##### Annexure-2

**DECLARATION**

I, **‘Toshiba Ansari’**, **‘Abhishek Singh’**, **‘Rahul Goyal’**, **‘Pushp Jain’** student of **‘Bachelor of Engineering in AIML’**, **session:2020-2024**, Department of Computer Science and Engineering, Apex Institute of Technology, Chandigarh University, Punjab, hereby declare that the work presented in this Project Work entitled ‘**Detecting Face Mask using Artificial Intelligence, Machine Learning and Deep Learning’** is the outcome of our own bona fide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. It contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

#### Date: 20/4/2022

**Place: Chandigarh University, Mohali**

### (Rahul Goyal) Candidate UID: 20BCS6713

ii

**Abstract**

COVID-19 pandemic has rapidly affected our day-to-day life disrupting the world trade and movements. Wearing a protective face mask has become a new normal. In the near future, many public service providers will ask the customers to wear masks correctly to avail of their services. Therefore, face mask detection has become a crucial task to help global society. This paper presents a simplified approach to achieve this purpose using some basic Machine Learning packages like TensorFlow, Keras, OpenCV and Scikit-Learn. The proposed method detects the face from the image correctly and then identifies if it has a mask on it or not. As a surveillance task performer, it can also detect a face along with a mask in motion. The method attains accuracy up to 98.23% respectively on two different datasets. We explore optimized values of parameters using the MobilenetV2 model to detect the presence of masks correctly without causing over-fitting.

**Acknowledgement**

In successfully completing this project, many people have helped me. I would like to thank all those who are related to this project.

Primarily, I would thank God for being able to complete this project with success. Then I will thank my teacher, under whose guidance I learned a lot about this project. His suggestions and directions have helped in the completion of this project.

Finally, I would like to thank my parents and friends who have helped me with their valuable suggestions and guidance and have been very helpful in various stages of project completion.

***Annexure-3* (A typical specimen of table of contents)**

# Table of Contents

|  |  |  |
| --- | --- | --- |
|  | Title Page  Declaration of the Student Abstract Acknowledgement | i |
| ii |
| iii |
| iv |
|  |
|  |
|  |
| **1.** | **INTRODUCTION\*** |  |
|  | 1.1 Introduction  1.2 Project Motivation  1.3 Problem Statement  1.4 Project Contribution  1.5 Related Work  1.6 Software and Libraries |  |
| 1. **LITERATURE SURVEY** 2. **PROBLEM FORMULATION** 3. **OBJECTIVES** 4. **METHODOLOGY**   **5.1** Data collection  **5.2** Preprocessing  **5.3** Classic vs. Depthwise Separable Convolutions  **6. RESULT AND DISCUSSION**  **7. REFERENCES**  **8. RESEARCH PAPER** | |  |
|  |
|  |
|  | |

## List of Figures

##### Figure Title page

viii

# 1. INTRODUCTION

The COVID-19 pandemic emerged in December 2019 in metropolis town within the Hubei province of central China. Perceptive the virus’s growth and unfold among humans, the World Health Organization declared the corona virus (i.e., Sars-CoV-2) to be a world pandemic in March 2020. This pandemic has devastating effects on societies and economies round the world inflicting a world health crisis . It's associate degree rising metastasis communicable disease caused by Severe Acute metastasis Syndrome Coronavirus2. Everywhere on the planet, particularly within the third wave, COVID-19 has been a major health care challenge. Several shutdowns in numerous industries are caused by this pandemic. According to the centers for Disease Control and Prevention (CDC), corona viral infection is transmitted predominantely by respiratory droplets made once individuals breathe, talk, cough, or sneeze with common drop size 5–10’m however aerosol emission will increase once humans speak and shout loudly .People with COVID-19 have had a good scope of symptoms reported like shortness of breath or issue in respiratory. Elder individuals having respiratory organ unwellness are at higher risk of obtaining corona virus than most. Therefore, to forestall speedy COVID-19 infection, several solutions, like confinement and lockdowns, are recommended by the bulk of the world’s governments. It's true that COVID-19 could be a world pandemic and have an effect on many domains. The importance of sporting masks be reducing vulnerability of risk from a pestilent individual throughout the “pre-symptomatic” amount to restrain the spreading of the virus. More than five million cases were infected by COVID- 19 in less than 6 months across 188 countries. The virus spreads through close contact and in crowded and overcrowded areas.

We can tackle and predict new diseases by the help of new Technologies such as artificial intelligence, IOT, Big data, and Machine learning.

It created a path for researchers in engineering. We've got seen multiple analysis topics, like making new automatic detection ways of COVID-19 and detection individuals with or while not masks. Before corona virus, some individuals place masks to shield themselves from pollution, whereas others place face masks to cover their faces and their emotions from others. Protection against corona virus could be a necessary counter live, per the WHO [1]. Indeed, sporting a mask is a good methodology of obstruction 80 of all metastasis infections [2]. Several organizations enforce mask rules for the non-public protection. Checking manually if people coming into a corporation are sporting masks is cumbersome and probably conflicting. It's crucial to watch mask usage across numerous regions to adequately offer info to policy manufacturers and epidemiologists UN agency project the progress of the irruption. As a results of COVID-19, the necessity has arisen to develop associate degree economical mask detection algorithmic rule to trace mask usage in inhabited areas. A way to see mask usage while not any spreading the virus is to watch the publicly accessible webcams in bulk and examine the faces for masks.

A mask detector system is enforced to envision this. Mask detection means that to spot whether or not someone is sporting a mask or not. the primary step to acknowledge the presence of a mask on the face is to observe the face, that makes the strategy divided into 2 parts: to observe faces and to observe masks on those faces.

Deep learning has been won’t to establish UN agency isn't sporting the facial mask mistreatment Convolutional neural network. It's various applications, like autonomous driving, education, police work, and so on . The approach is ascendable, safe to execute, and provides a much bigger image of mask usage within the world. There are several detector systems developed round the world and being enforced. However, all this science desires optimization; a stronger, a lot of precise Detector, as a result of the world cannot afford to any extent further increase in corona cases. Considering AI legal issues and advantages in combating COVID-19 pandemic, AI technique-based solutions are still associate degree open window for development and legal interpretation. The sphere of AI (AI) analysis has advanced considerably in recent years, particularly within the space of machine learning. Any fresh developed technology is indivisible from the term AI. While not AI it's terribly tough today to form any vital progress in terms of technical innovation. AI is being thought of because the next huge issue that may amendment the world hugely.

**Project Motivation**

There has been a tremendous amount of interest in deep learning in the past few years, notably in fields like machine vision, text analytics, object recognition, and other information processing aspects. The majority of the past research in object detection has been conducted using convolutional neural network models. Using deep learning architectures, convolutional neural networks (CNNs) have become more popular in recent years for a variety of tasks, such as picture identification, speech synthesis, object tracking, and image thresholding. When it comes to the abovementioned domains, CNN exhibits an excellent capacity to retrieve features from images. A growing number of research methods are replacing traditional classification methods with CNNs in order to more effectively capture image information and achieve improved classification performance. Due to energy limitations, numerous deep neural networks are unsuited for mobile-based facial image classification since their evaluation phase is time consuming and expensive. We describe a MobileNet-based facial image classification model that uses a Depthwise separable convolution technique to handle this problem. DSC (Depthwise separable convolution) was first presented in and is commonly used in image processing for classification tasks. The Depthwise separable convolution is a quantized version of the ordinary convolution. Convolutions are often separated into Depthwise and 1×1 pointwise convolutions. Rather than applying each filter to all input channels as in traditional convolution, the Depthwise convolution layer applies one filtering to one pulse and then uses a 1×1 pointwise convolution to combine the Depthwise convolution results. Depthwise separable convolution reduces the number of learnable parameters and the expense of test and train computations.

### Problem Statement

COVID-19 is a highly contagious disease, and the WHO and other health agencies have recommended that people use face masks to prevent its transmission. All governments are attempting to guarantee that face masks are worn in public places, but it is difficult to manually identify those who are not wearing face masks in crowded places. Scientists are working on developing automatic methods to identify and enforce the use of face masks in public locations. The problem may be summarized as follows: given a face picture as an input, the classification model must categorize the facial image in a mask detection task using the classification model. Using Depthwise Separable Convolutions with MobileNet data, we provide a method for mask detection-driven face picture classification that is both fast and accurate, as demonstrated in this work. We employed Depthwise separable convolution layers instead of traditional convolutional layers to successfully develop the model with a smaller number of learnable parameters and a smaller number of learnable parameters.

### Project Contributions

The following is a list of the project's most significant contributions:

• We describe an effective face mask-based facial image classification system using a MobileNet-based deep learning model with a Depthwise separable convolution approach.

• Faster training with fewer parameters is possible with the proposed multilayer MobileNet-based model.

• With Depthwise separable convolution units, we synthesize mobile-based input patterns using their internal memory layouts and resource consumption.

• When evaluated on a publicly accessible dataset, the suggested strategy outperforms current state-of-the-art image classification methods.

## Related work

Face mask detection is a subset of object recognition that uses image processing algorithms. Digital image processing may be divided into two broad categories: classical image processing and deep learning-based image analysis. As opposed to classical image analysis, which uses complex formulas to recognize and interpret pictures, deep learning-based approaches utilize models that mimic the workings of the human brain. Deep Learning models have been used in the majority of past research. After correctly recognizing the face in the picture or video, the CNN-based approach by Kaur et al. Evaluates if the face has been disguised. It is also capable of identifying a moving face and a mask in a video as a surveillance job performance. Accuracy is great with this method. An algorithm called YOLO-v3 was developed by Bhuiyan et al. To identify face masks in public spaces. They trained the YOLO-v3 model on their own custom dataset of photos with people labelled as “mask and no-mask.” The model's performance was enhanced by Mata via data augmentation. It is necessary to create a CNN model that can distinguish between ROIs with and those without a face mask in order to extract the facial area as a ROI. With the use of Mobile NetV2, Toppo et al. Developed a method for detecting face masks that incorporates three distinct face detector models in order to test the model's correctness and evaluate its performance. The trained model's outcome allows for implementation on low-power devices, making the mask detection method's inclusion faster than previous strategies. To recognize people who were not wearing face masks in government workplaces, Balaji et al. Utilized a VGG-16 CNN model developed in Keras/TensorFlow and Open-CV to detect people who were not wearing face masks. To compensate for the model's light weight, Fan et al. Offered two additional methods. A unique residual contextual awareness module for crucial face mask regions Two-stage synthetic Gaussian heat map regression is used to identify better mask discrimination features. Ablation research has found that these strategies can improve feature engineering and, as a result, the effectiveness of numeric identification. For AIZOO and Moxa3K, the suggested model outperforms prior models.

Conventional deep learning algorithms for lightweight facial image classification alone do not give a good discriminating feature space, as shown by the research covered above, and they complicate the model and greatly increase the number of parameters and necessary computational resources.

In this study, a Depthwise Separable Convolution Neural Network-based MobileNet for the detection of face masks by classifying facial images is developed in this study in an effort to answer the shortfalls of previous research in this area. Our technique improves the work performed by replacing the conventional convolution with a depth-wise separable convolution in the neural network.

**Software and Libraries Used**

**1. Anaconda (Python Distribution)**

Anaconda is a [distribution](https://en.wikipedia.org/wiki/Software_distribution" \o "Software distribution) of the [Python](https://en.wikipedia.org/wiki/Python_(programming_language)" \o "Python (programming language)) and [R](https://en.wikipedia.org/wiki/R_(programming_language)" \o "R (programming language)) [programming languages](https://en.wikipedia.org/wiki/Programming_language" \o "Programming language) for [scientific computing](https://en.wikipedia.org/wiki/Scientific_computing" \o "Scientific computing) ([data science](https://en.wikipedia.org/wiki/Data_science" \o "Data science), [machine learning](https://en.wikipedia.org/wiki/Machine_learning" \o "Machine learning) applications, large-scale [data processing](https://en.wikipedia.org/wiki/Data_processing" \o "Data processing), [predictive analytics](https://en.wikipedia.org/wiki/Predictive_analytics" \o "Predictive analytics), etc.), that aims to simplify [package management](https://en.wikipedia.org/wiki/Package_management" \o "Package management) and [deployment](https://en.wikipedia.org/wiki/Deployment_environment" \o "Deployment environment). The distribution includes data-science packages suitable for [Windows](https://en.wikipedia.org/wiki/Microsoft_Windows" \o "Microsoft Windows), [Linux](https://en.wikipedia.org/wiki/Linux" \o "Linux), and [macOS](https://en.wikipedia.org/wiki/MacOS" \o "MacOS). It is developed and maintained by Anaconda, Inc., which was founded by Peter Wang and [Travis Oliphant](https://en.wikipedia.org/wiki/Travis_Oliphant" \o "Travis Oliphant) in 2012. As an Anaconda, Inc. product, it is also known as Anaconda Distribution or Anaconda Individual Edition, while other products from the company are Anaconda Team Edition and Anaconda Enterprise Edition, both of which are not free.

Package versions in Anaconda are managed by the package management system *[conda](https://en.wikipedia.org/wiki/Conda_(package_manager)" \o "Conda (package manager))*. This package manager was spun out as a separate [open-source](https://en.wikipedia.org/wiki/Open_source" \o "Open source) package as it ended up being useful on its own and for things other than Python. There is also a small, [bootstrap](https://en.wikipedia.org/wiki/Bootstrapping" \o "Bootstrapping) version of Anaconda called Miniconda, which includes only conda, Python, the packages they depend on, and a small number of other packages.

Anaconda distribution comes with over 250 packages automatically installed, and over 7,500 additional open-source packages can be installed from [PyPI](https://en.wikipedia.org/wiki/Python_Package_Index" \o "Python Package Index) as well as the conda package and [virtual environment](https://en.wikipedia.org/wiki/Virtual_environment_software" \o "Virtual environment software) manager. It also includes a [GUI](https://en.wikipedia.org/wiki/Graphical_user_interface" \o "Graphical user interface), Anaconda Navigator,  as a graphical alternative to the [command-line interface](https://en.wikipedia.org/wiki/Command-line_interface" \o "Command-line interface) (CLI).

The big difference between conda and the [pip package manager](https://en.wikipedia.org/wiki/Pip_(package_manager)" \o "Pip (package manager)) is in how package dependencies are managed, which is a significant challenge for Python data science and the reason conda exists.

Before version 20.3, when pip installed a package, it automatically installed any dependent Python packages without checking if these conflict with previously installed packages. It would install a package and any of its dependencies regardless of the state of the existing installation. Because of this, a user with a working installation of, for example, [TensorFlow](https://en.wikipedia.org/wiki/TensorFlow" \o "TensorFlow), could find that it stopped working having used pip

to install a different package that requires a different version of the dependent [numpy](https://en.wikipedia.org/wiki/NumPy" \o "NumPy) library than the one used by TensorFlow. In some cases, the package would appear to work but produce different results in detail. While pip has since implemented consistent dependency resolution. this difference accounts for a historical differentiation of the conda package manager.

In contrast, conda analyses the current environment including everything currently installed, and, together with any version limitations specified (e.g. the user may wish to have TensorFlow version 2,0 or higher), works out how to install a compatible set of dependencies, and shows a warning if this cannot be done.

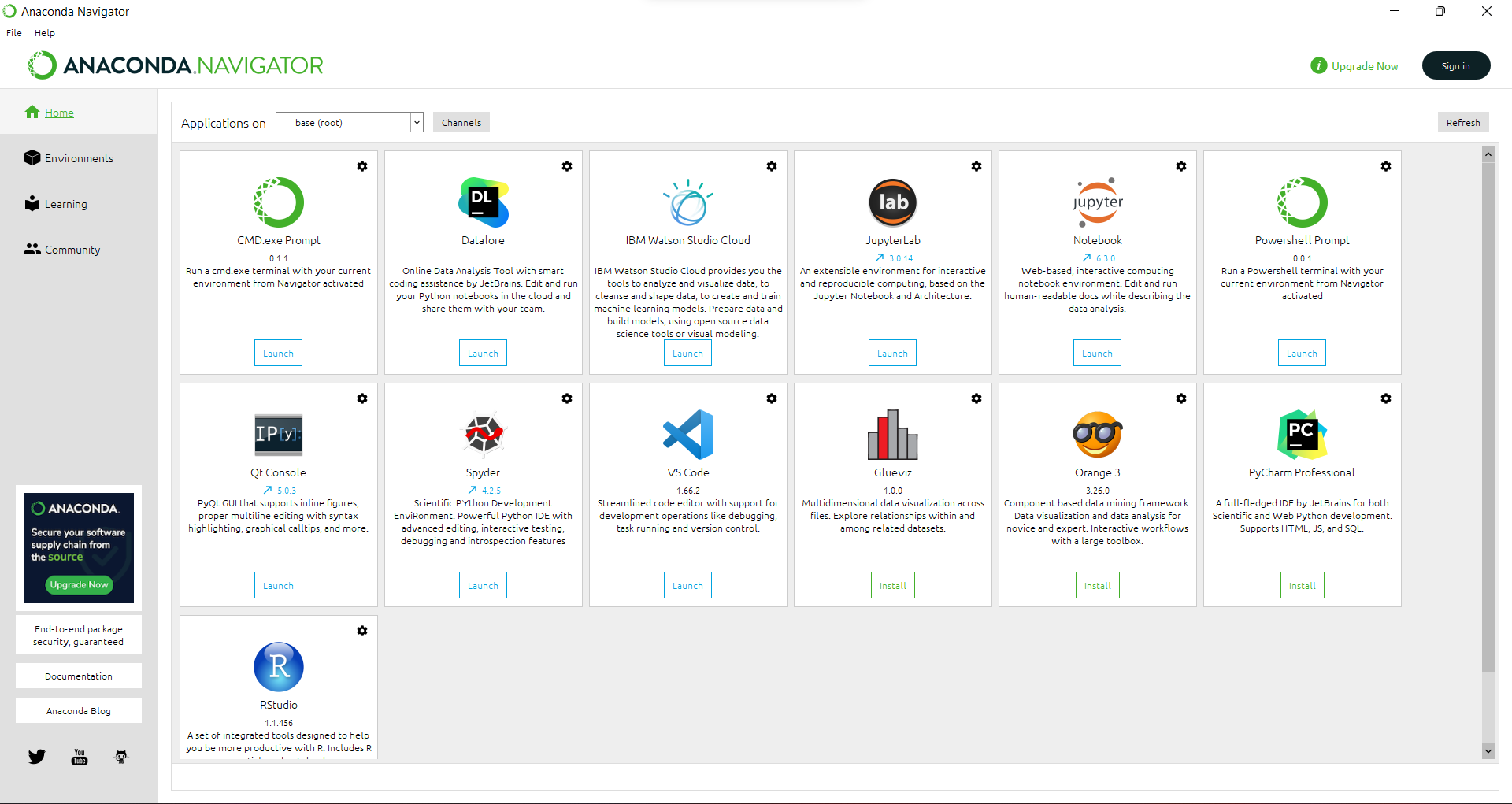
Open source packages can be individually installed from the Anaconda repository, Anaconda Cloud (anaconda.org), or the user's own private repository or mirror, using the conda install command. Anaconda, Inc. [compiles](https://en.wikipedia.org/wiki/Compiler" \o "Compiler) and builds the packages available in the Anaconda [repository](https://en.wikipedia.org/wiki/Repository_(version_control)" \o "Repository (version control)) itself, and provides [binaries](https://en.wikipedia.org/wiki/Binary_file" \o "Binary file) for Windows [32](https://en.wikipedia.org/wiki/32-bit_computing" \o "32-bit computing)/[64 bit](https://en.wikipedia.org/wiki/64-bit_computing" \o "64-bit computing), Linux 64 bit and MacOS 64-bit. Anything available on PyPI may be installed into a conda environment using pip, and conda will keep track of what it has installed itself and what pip has installed.

Custom packages can be made using the conda build command, and can be shared with others by uploading them to Anaconda Cloud,  PyPI or other repositories.

The default installation of Anaconda2 includes Python 2.7 and Anaconda3 includes Python 3.7. However, it is possible to create new environments that include any version of Python packaged with conda.



Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows users to launch applications and manage conda packages, environments and channels without using [command-line commands](https://en.wikipedia.org/wiki/Command-line_interface" \o "Command-line interface). Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository, install them in an environment, run the packages and update them. It is available for Windows, macOS and Linux.



The following applications are available by default in Navigator:

* [JupyterLab](https://en.wikipedia.org/wiki/Project_Jupyter" \l "JupyterLab" \o "Project Jupyter)
* [Jupyter Notebook](https://en.wikipedia.org/wiki/Project_Jupyter" \l "Jupyter_Notebook" \o "Project Jupyter)
* QtConsole
* [Spyder](https://en.wikipedia.org/wiki/Spyder_(software)" \o "Spyder (software))
* [Glue](https://en.wikipedia.org/wiki/Glue_(software)" \o "Glue (software))
* [Orange](https://en.wikipedia.org/wiki/Orange_(software)" \o "Orange (software))
* [RStudio](https://en.wikipedia.org/wiki/RStudio" \o "RStudio)
* [Visual Studio Code](https://en.wikipedia.org/wiki/Visual_Studio_Code" \o "Visual Studio Code)

**2. Jupyter Notebook**

****

Jupyter Notebook was created to make it easier to show one’s programming work, and to let others join in. Jupyter Notebook allows you to combine code, comments, multimedia, and visualizations in an interactive document called a notebook, naturally that can be shared, re-used, and re-worked.

And because Jupyter Notebook runs via a web browser, the notebook itself could be hosted on your local machine or on a remote server.

Originally developed for data science applications written in Python, R, and Julia, Jupyter Notebook is useful in all kinds of ways for all kinds of projects:

* **Data visualizations.** Most people have their first exposure to Jupyter Notebook by way of a data visualization, a shared notebook that includes a rendering of some data set as a graphic. Jupyter Notebook lets you author visualizations, but also share them and allow interactive changes to the shared code and data set.
* **Code sharing.** Cloud services like GitHub and Pastebin provide ways to share code, but they’re largely non-interactive. With a Jupyter Notebook, you can view code, execute it, and display the results directly in your web browser.
* **Live interactions with code.** Jupyter Notebook code isn’t static; it can be edited and re-run incrementally in real time, with feedback provided directly in the browser. Notebooks can also embed user controls (e.g., sliders or text input fields) that can be used as input sources for code.
* **Documenting code samples.** If you have a piece of code and you want to explain line-by-line how it works, with live feedback all along the way, you could embed it in a Jupyter Notebook. Best of all, the code will remain fully functional—you can add interactivity along with the explanation, showing and telling at the same time.

Jupyter Notebooks can include several kinds of ingredients, each organized into discrete blocks:

* **Text and HTML.** Plain text, or text annotated in the Markdown syntax to generate HTML, can be inserted into the document at any point. CSS styling can also be included inline or added to the template used to generate the notebook.
* **Code and output.** The code in Jupyter Notebook notebooks is typically Python code, although you may add support in your Jupyter environment for other languages such as R or Julia. The results of executed code appear immediately after the code blocks, and the code blocks can be executed and re-executed in any order you like, as often as you like.

* **Visualizations.**Graphics and charts can be generated from code, by way of modules like [Matplotlib](https://matplotlib.org/), [Plotly](https://plot.ly/d3-js-for-python-and-pandas-charts/), or [Bokeh](https://bokeh.pydata.org/en/latest/). Like output, these visualizations appear inline next to the code that generates them. However, code can also be configured to write them out to external files if needed.
* **Multimedia.**Because Jupyter Notebook is built on web technology, it can display all the types of multimedia supported in a web page. You can include them in a notebook as HTML elements, or you can generate them programmatically by way of the IPython.display module.
* **Data.** Data can be provided in a separate file alongside the .ipynb file that constitutes a Jupyter Notebook notebook, or it can be imported programmatically—for instance, by including code in the notebook to download the data from a public Internet repository or to access it via a database connection.

The most common use cases for Jupyter Notebook are data science, mathematics, and other research projects that involve visualizations of data or formulas. Apart from those, though, there are plenty of other use cases:

* **Sharing a visualization, with or without interactivity.** People often share the results of a data visualization as a static image, but that’s useful only up to a point. By sharing a Jupyter notebook, you allow your target audience to dive in and play around. They can gain a thorough understanding of the data, interactively.
* **Documenting a process with code.** Many programmers who blog about their programming experiences write up their posts in a Jupyter notebook. Others can download their notebook and recreate the exercise.
* **Live documentation for a library or module.** Most documentation for Python modules is static; a Jupyter notebook can be used as an interactive sandbox for learning how a module works. Any Python module that runs well in a notebook interface (essentially, anything that writes to stdout as part of its behavior) is a good candidate for this.
* **Sharing code and data generally.** All you need to do to share a Jupyter notebook and its associated data files is pack it up into an archive.

A next-generation user interface for Jupyter Notebook, called JupyterLab, is now available and considered ready for production use.

As explained in the [blog post](https://blog.jupyter.org/jupyterlab-is-ready-for-users-5a6f039b8906) announcing general availability, JupyterLab is more malleable than a conventional Jupyter Notebook, allowing users to drag-and-drop cells within and between notebooks and to arrange the

workspace into separate tabs and subsections. Code can run directly from text files as well as Jupyter Notebook files, and many common file formats for both code and data can be rendered with live previews.

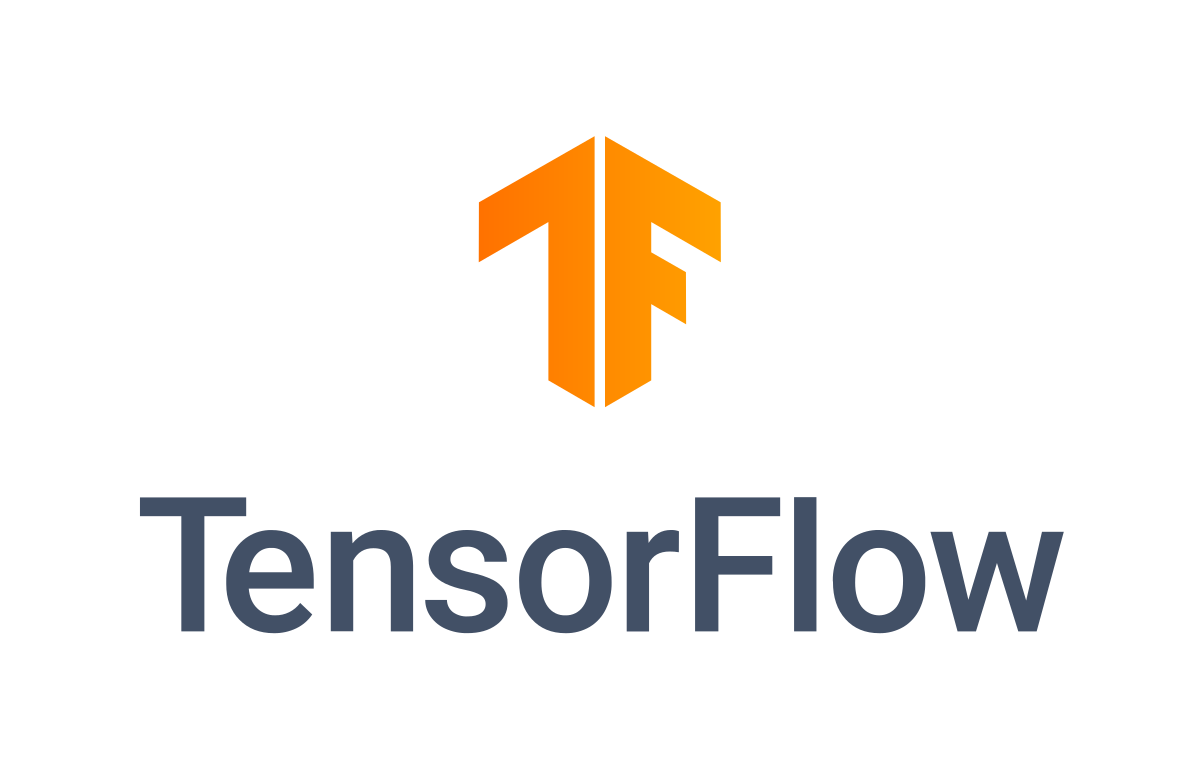
JupyterLab can also be customized with extensions to add support for new file formats, to enrich the interface, or to provide any number of other additional features, enabling a much broader range of notebook applications than Jupyter Notebook. The long-term plan is to replace the current Jupyter Notebook interface with JupyterLab, but only after JupyterLab has proved sufficiently stable and reliable.

As powerful and useful as Jupyter Notebook can be, it does have some limitations that need to be taken into account.

* **Notebooks aren’t self-contained.** This is the single biggest drawback of using Jupyter Notebook: Notebooks require the Jupyter runtime, along with any libraries you plan on using. [A few strategies exist](https://blog.ouseful.info/2014/12/12/seven-ways-of-running-ipython-notebooks/) for creating self-contained Jupyter Notebooks, but none of them is officially supported. You’re best off distributing notebooks to people who already have infrastructure in place to run them, or don’t mind the setup (by way of [Anaconda](https://www.infoworld.com/article/3245814/get-started-with-anaconda-python-the-distro-for-data-science.html), for instance).
* **Session state cannot be saved easily.** The state of any code running in a Jupyter notebook cannot be preserved and restored with Jupyter Notebook’s default toolset. Every time you load the notebook, you will need to re-run the code in it to restore its state.
* **No interactive debugging or other IDE features.** Jupyter Notebook is not a full-blown development environment for Python. Many of the features you would expect to find in an IDE—e.g., interactive debugging, code completion, and module management—aren’t available there.

**3. Tensorflow:**

TensorFlow is an end-to-end open-source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML powered applications.



### Easy model building

TensorFlow offers multiple levels of abstraction so you can choose the right one for your needs. Build and train models by using the high-level Keras API, which makes getting started with TensorFlow and machine learning easy.

If you need more flexibility, eager execution allows for immediate iteration and intuitive debugging. For large ML training tasks, use the Distribution Strategy API for distributed training on different hardware configurations without changing the model definition.

### Robust ML production anywhere

TensorFlow has always provided a direct path to production. Whether it's on servers, edge devices, or the web, TensorFlow lets you train and deploy your model easily, no matter what language or platform you use.

Use TensorFlow Extended (TFX) if you need a full production ML pipeline. For running inference on mobile and edge devices, use TensorFlow Lite. Train and deploy models in JavaScript environments using TensorFlow.js.

### Powerful experimentation for research

Build and train state-of-the-art models without sacrificing speed or performance. TensorFlow gives you the flexibility and control with features like the Keras Functional API and Model Subclassing API for creation of complex topologies. For easy prototyping and fast debugging, use eager execution.

TensorFlow also supports an ecosystem of powerful add-on libraries and models to experiment with, including Ragged Tensors, TensorFlow Probability, Tensor2Tensor and BERT.

**3.1 Tensorflow – Keras:**

Keras is a deep learning API written in Python, running on top of the machine learning platform [TensorFlow](https://github.com/tensorflow/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.

****

Keras is:

* **Simple** -- but not simplistic. Keras reduces developer *cognitive load* to free you to focus on the parts of the problem that really matter.
* **Flexible** -- Keras adopts the principle of *progressive disclosure of complexity*: simple workflows should be quick and easy, while arbitrarily advanced workflows should be *possible* via a clear path that builds upon what you've already learned.
* **Powerful** -- Keras provides industry-strength performance and scalability: it is used by organizations and companies including NASA, YouTube, or Waymo.

## First contact with Keras

The core data structures of Keras are **layers** and **models**. The simplest type of model is the [Sequential model](https://keras.io/guides/sequential_model/), a linear stack of layers. For more complex architectures, you should use the [Keras functional API](https://keras.io/guides/functional_api/), which allows to build arbitrary graphs of layers, or [write models entirely from scratch via subclasssing](https://keras.io/guides/making_new_layers_and_models_via_subclassing/).

## Installation & compatibility

Keras comes packaged with TensorFlow 2 as tensorflow.keras. To start using Keras, simply [install TensorFlow 2](https://www.tensorflow.org/install).

Keras/TensorFlow are compatible with:

* Python 3.7–3.10
* Ubuntu 16.04 or later
* Windows 7 or later
* macOS 10.12.6 (Sierra) or later.

## Why this name, Keras?

Keras (κέρας) means horn in Greek. It is a reference to a literary image from ancient Greek and Latin literature, first found in the Odyssey, where dream spirits (Oneiroi, singular Oneiros) are divided between those who deceive dreamers with false visions, who arrive to Earth through a gate of ivory, and those who announce a future that will come to pass, who arrive through a gate of horn. It's a play on the words κέρας (horn) / κραίνω (fulfill), and ἐλέφας (ivory) / ἐλεφαίρομαι (deceive).

Keras was initially developed as part of the research effort of project ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System).

**4. IMUTILS:**

A series of convenience functions to make basic image processing functions such as translation, rotation, resizing, skeletonization, displaying Matplotlib images, sorting contours, detecting edges, and much more easier with OpenCV and both Python 2.7 and Python 3.



**5. NUMPY:**

****

## What is NumPy?

NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices.

NumPy was created in 2005 by Travis Oliphant. It is an open source project and you can use it freely.

NumPy stands for Numerical Python.

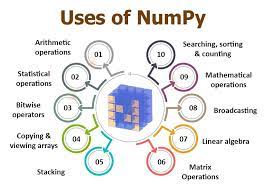
## Why Use NumPy?

In Python we have lists that serve the purpose of arrays, but they are slow to process.

NumPy aims to provide an array object that is up to 50x faster than traditional Python lists.

The array object in NumPy is called ndarray, it provides a lot of supporting functions that make working with ndarray very easy.

Arrays are very frequently used in data science, where speed and resources are very important.



## Why is NumPy Faster Than Lists?

NumPy arrays are stored at one continuous place in memory unlike lists, so processes can access and manipulate them very efficiently.

This behavior is called locality of reference in computer science.

This is the main reason why NumPy is faster than lists. Also it is optimized to work with latest CPU architectures.

## Which Language is NumPy written in?

NumPy is a Python library and is written partially in Python, but most of the parts that require fast computation are written in C or C++.

## Where is the NumPy Codebase?

The source code for NumPy is located at this github repository [https://github.com/numpy/numpy](https://github.com/numpy/numpy" \t "_blank)

1. **SKLEARN:**

**Scikit-learn** (formerly **scikits.learn** and also known as **sklearn**) is a [free software](https://en.wikipedia.org/wiki/Free_software" \o "Free software) [machine learning](https://en.wikipedia.org/wiki/Machine_learning" \o "Machine learning) [library](https://en.wikipedia.org/wiki/Library_(computing)" \o "Library (computing)) for the [Python](https://en.wikipedia.org/wiki/Python_(programming_language)" \o "Python (programming language)) [programming language](https://en.wikipedia.org/wiki/Programming_language" \o "Programming language). It features various [classification](https://en.wikipedia.org/wiki/Statistical_classification" \o "Statistical classification), [regression](https://en.wikipedia.org/wiki/Regression_analysis" \o "Regression analysis) and [clustering](https://en.wikipedia.org/wiki/Cluster_analysis" \o "Cluster analysis) algorithms including [support-vector machines](https://en.wikipedia.org/wiki/Support_vector_machine" \o "Support vector machine), [random forests](https://en.wikipedia.org/wiki/Random_forests" \o "Random forests), [gradient boosting](https://en.wikipedia.org/wiki/Gradient_boosting" \o "Gradient boosting), *[k](https://en.wikipedia.org/wiki/K-means_clustering" \o "K-means clustering)*[-means](https://en.wikipedia.org/wiki/K-means_clustering" \o "K-means clustering) and [DBSCAN](https://en.wikipedia.org/wiki/DBSCAN" \o "DBSCAN), and is designed to interoperate with the Python numerical and scientific libraries [NumPy](https://en.wikipedia.org/wiki/NumPy" \o "NumPy) and [SciPy](https://en.wikipedia.org/wiki/SciPy" \o "SciPy). Scikit-learn is a [NumFOCUS](https://en.wikipedia.org/w/index.php?title=NumFOCUS&action=edit&redlink=1" \o "NumFOCUS (page does not exist)) fiscally sponsored project.

****

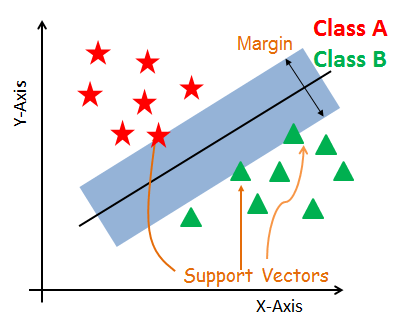
The scikit-learn project started as scikits.learn, a [Google Summer of Code](https://en.wikipedia.org/wiki/Google_Summer_of_Code" \o "Google Summer of Code) project by French [data scientist](https://en.wikipedia.org/wiki/Data_scientist" \o "Data scientist) [David Cournapeau](https://en.wikipedia.org/wiki/David_Cournapeau" \o "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](https://en.wikipedia.org/wiki/SciPy" \o "SciPy). The original [codebase](https://en.wikipedia.org/wiki/Codebase" \o "Codebase) was later rewritten by other developers. In 2010 Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort and Vincent Michel, all from the [French Institute for Research in Computer Science and Automation](https://en.wikipedia.org/wiki/French_Institute_for_Research_in_Computer_Science_and_Automation" \o "French Institute for Research in Computer Science and Automation) in [Rocquencourt](https://en.wikipedia.org/wiki/Rocquencourt,_Yvelines" \o "Rocquencourt, Yvelines), [France](https://en.wikipedia.org/wiki/France" \o "France), took leadership of the project and made the first public release on February the 1st 2010. Of the various scikits, scikit-learn as well as [scikit-image](https://en.wikipedia.org/wiki/Scikit-image" \o "Scikit-image) were described as "well-maintained and popular" in November 2012. Scikit-learn is one of the most popular machine learning libraries on [GitHub](https://en.wikipedia.org/wiki/GitHub" \o "GitHub).

## Implementation

Scikit-learn is largely written in Python, and uses [NumPy](https://en.wikipedia.org/wiki/NumPy" \o "NumPy) extensively for high-performance linear algebra and array operations. Furthermore, some core algorithms are written in [Cython](https://en.wikipedia.org/wiki/Cython" \o "Cython) to improve performance. Support vector machines are implemented by a Cython wrapper around [LIBSVM](https://en.wikipedia.org/wiki/LIBSVM" \o "LIBSVM); logistic regression and linear support vector machines by a similar wrapper around [LIBLINEAR](https://en.wikipedia.org/wiki/LIBLINEAR" \o "LIBLINEAR). In such cases, extending these methods with Python may not be possible.

Scikit-learn integrates well with many other Python libraries, such as [Matplotlib](https://en.wikipedia.org/wiki/Matplotlib" \o "Matplotlib) and [plotly](https://en.wikipedia.org/wiki/Plotly" \o "Plotly) for plotting, [NumPy](https://en.wikipedia.org/wiki/NumPy" \o "NumPy) for array vectorization, [Pandas](https://en.wikipedia.org/wiki/Pandas_(software)" \o "Pandas (software)) dataframes, [SciPy](https://en.wikipedia.org/wiki/SciPy" \o "SciPy), and many more.\

**6. SKlearn – SVM:**



In [machine learning](https://en.wikipedia.org/wiki/Machine_learning" \o "Machine learning), **support-vector machines** (**SVMs**, also **support-vector networks**[[1]](https://en.wikipedia.org/wiki/Support-vector_machine" \l "cite_note-CorinnaCortes-1)) are [supervised learning](https://en.wikipedia.org/wiki/Supervised_learning" \o "Supervised learning) models with associated learning [algorithms](https://en.wikipedia.org/wiki/Algorithm" \o "Algorithm) that analyze data for [classification](https://en.wikipedia.org/wiki/Statistical_classification" \o "Statistical classification) and [regression analysis](https://en.wikipedia.org/wiki/Regression_analysis" \o "Regression analysis). Developed at [AT&T Bell Laboratories](https://en.wikipedia.org/wiki/AT%26T_Bell_Laboratories" \o "AT&T Bell Laboratories) by [Vladimir Vapnik](https://en.wikipedia.org/wiki/Vladimir_Vapnik" \o "Vladimir Vapnik) with colleagues (Boser et al., 1992, [Guyon](https://en.wikipedia.org/wiki/Isabelle_Guyon" \o "Isabelle Guyon) et al., 1993, [Cortes](https://en.wikipedia.org/wiki/Corinna_Cortes" \o "Corinna Cortes) and Vapnik, 1995,[[2]](https://en.wikipedia.org/wiki/Support-vector_machine" \l "cite_note-article1995-2) Vapnik et al., 1997[[citation needed](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed" \o "Wikipedia:Citation needed)]) SVMs are one of the most robust prediction methods, being based on statistical learning frameworks or [VC theory](https://en.wikipedia.org/wiki/VC_theory" \o "VC theory) proposed by Vapnik (1982, 1995) and Chervonenkis (1974). Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-[probabilistic](https://en.wikipedia.org/wiki/Probabilistic_classification" \o "Probabilistic classification) [binary](https://en.wikipedia.org/wiki/Binary_classifier" \o "Binary classifier) [linear classifier](https://en.wikipedia.org/wiki/Linear_classifier" \o "Linear classifier) (although methods such as [Platt scaling](https://en.wikipedia.org/wiki/Platt_scaling" \o "Platt scaling) exist to use SVM in a probabilistic classification setting). SVM maps training examples to points in space so as to maximise the width of the gap between the two categories. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

In addition to performing [linear classification](https://en.wikipedia.org/wiki/Linear_classifier" \o "Linear classifier), SVMs can efficiently perform a non-linear classification using what is called the [kernel trick](https://en.wikipedia.org/wiki/Kernel_method" \l "Mathematics:_the_kernel_trick" \o "Kernel method), implicitly mapping their inputs into high-dimensional feature spaces.

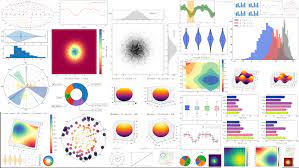
When data are unlabelled, supervised learning is not possible, and an [unsupervised learning](https://en.wikipedia.org/wiki/Unsupervised_learning" \o "Unsupervised learning) approach is required, which attempts to find natural [clustering of the data](https://en.wikipedia.org/wiki/Cluster_analysis" \o "Cluster analysis) to groups, and then map new data to these formed groups. The **support-vector clustering**algorithm, created by [Hava Siegelmann](https://en.wikipedia.org/wiki/Hava_Siegelmann" \o "Hava Siegelmann) and [Vladimir Vapnik](https://en.wikipedia.org/wiki/Vladimir_Vapnik" \o "Vladimir Vapnik), applies the statistics of support vectors, developed in the support vector machines algorithm, to categorize unlabeled data.

1. **MatplotLib:**

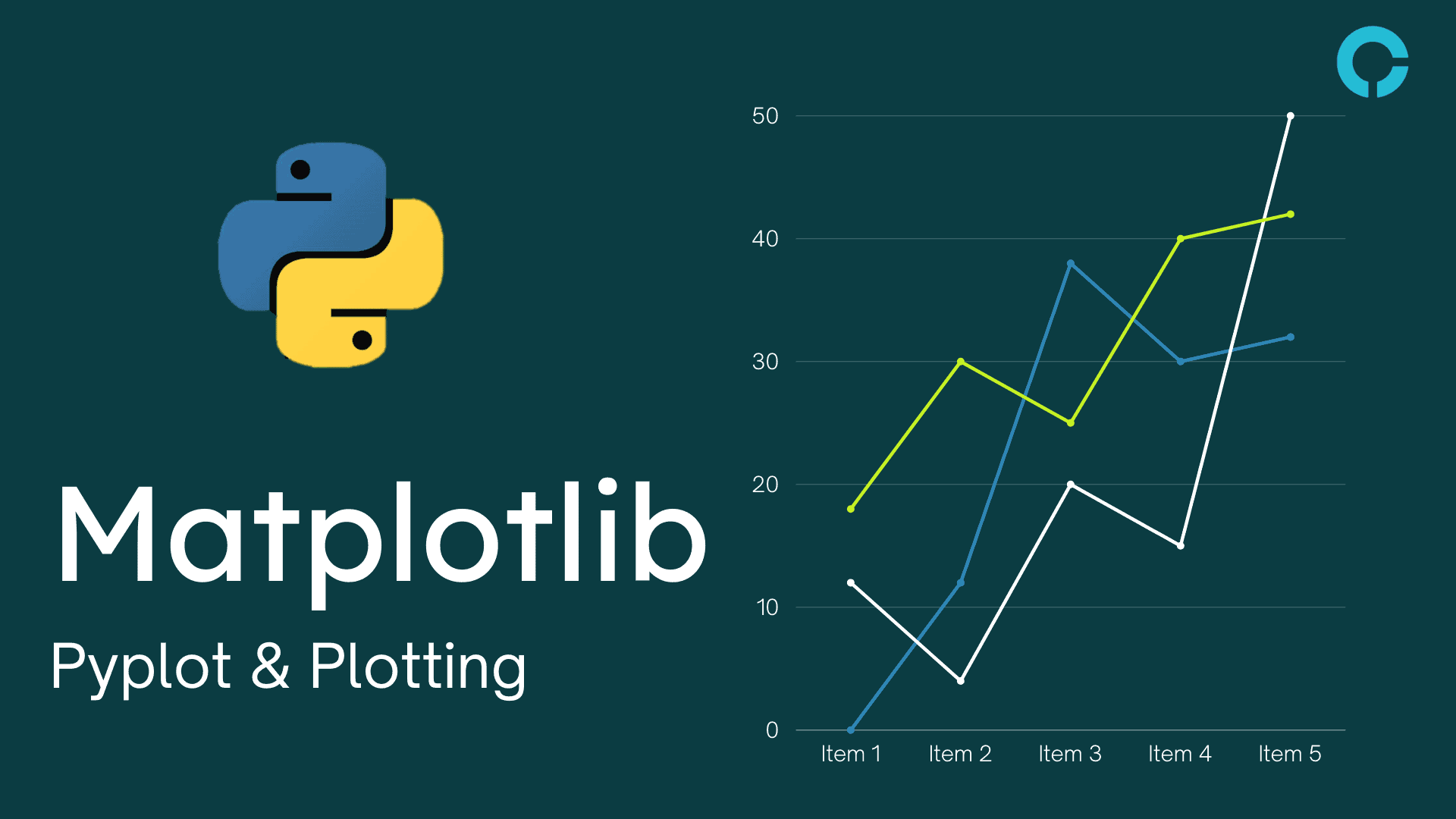
**Matplotlib** is a [plotting](https://en.wikipedia.org/wiki/Plotter" \o "Plotter) [library](https://en.wikipedia.org/wiki/Library_(computer_science)" \o "Library (computer science)) for the [Python](https://en.wikipedia.org/wiki/Python_(programming_language)" \o "Python (programming language)) programming language and its numerical mathematics extension [NumPy](https://en.wikipedia.org/wiki/NumPy" \o "NumPy). It provides an [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming" \o "Object-oriented programming) [API](https://en.wikipedia.org/wiki/API" \o "API) for embedding plots into applications using general-purpose [GUI toolkits](https://en.wikipedia.org/wiki/GUI_toolkit" \o "GUI toolkit) like [Tkinter](https://en.wikipedia.org/wiki/Tkinter" \o "Tkinter), [wxPython](https://en.wikipedia.org/wiki/WxPython" \o "WxPython), [Qt](https://en.wikipedia.org/wiki/Qt_(software)" \o "Qt (software)), or [GTK](https://en.wikipedia.org/wiki/GTK" \o "GTK). There is also a [procedural](https://en.wikipedia.org/wiki/Procedural_programming" \o "Procedural programming) "pylab" interface based on a [state machine](https://en.wikipedia.org/wiki/State_machine" \o "State machine) (like [OpenGL](https://en.wikipedia.org/wiki/OpenGL" \o "OpenGL)), designed to closely resemble that of [MATLAB](https://en.wikipedia.org/wiki/MATLAB" \o "MATLAB), though its use is discouraged.[SciPy](https://en.wikipedia.org/wiki/SciPy" \o "SciPy) makes use of Matplotlib.

Matplotlib was originally written by [John D. Hunter](https://en.wikipedia.org/wiki/John_D._Hunter" \o "John D. Hunter). Since then it has an active development community and is distributed under a [BSD-style license](https://en.wikipedia.org/wiki/BSD_licenses" \o "BSD licenses). Michael Droettboom was nominated as matplotlib's lead developer shortly before John Hunter's death in August 2012 and was further joined by Thomas Caswell. Matplotlib is a [NumFOCUS](https://en.wikipedia.org/w/index.php?title=NumFOCUS&action=edit&redlink=1" \o "NumFOCUS (page does not exist)) fiscally sponsored project.

Matplotlib 2.0.x supports Python versions 2.7 through 3.10. Python 3 support started with Matplotlib 1.2. Matplotlib 1.4 is the last version to support Python 2.6. Matplotlib has pledged not to support Python 2 past 2020 by signing the Python 3 Statement.



**7.1 MatplotLib – pyplot:**

****

**Pyplot is an API (Application Programming Interface) for Python’s matplotlib that effectively makes matplotlib a viable open source alternative to MATLAB. Matplotlib is a library for data visualization, typically in the form of plots, graphs and charts.**

## Pyplot API Structure

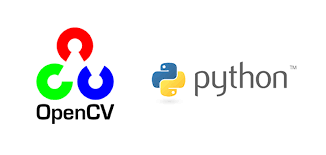
**Pyplot provides matplotlib with two key features:**

* A MATLAB-style interface, which allows those familiar with MATLAB to adopt Python more easily
* Statefulness, which means that pyplot stores the state of an object when you first plot it. This is essential for use in the same loop or session state until plt.close() is encountered in the code. State can also be important when creating several plots continuously.

**The pyplot API consists of a hierarchy of Python code objects, and includes numerous functions topped by**matplotlib.pyplot.**This stack can be viewed as having three interdependent layers:**

* Scripting layer – used to define a figure, which contains one or more plots, which consist of axes (i.e., x axis ,y axis, and possibly z axis)
* Artist Layer – used to manipulate elements of a plot, such as adding labels, drawing lines, etc
* Backend Layer – used to format the plot for display in a specific target application, such as a Jupyter Notebook

1. **OpenCV – CV2:**

****

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code.

The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc. OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding [18 million](https://sourceforge.net/projects/opencvlibrary/files/stats/timeline?dates=2001-09-20+to+2019-01-30" \t "_blank). The library is used extensively in companies, research groups and by governmental bodies.

Along with well-established companies like Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota that employ the library, there are many startups such as Applied Minds, VideoSurf, and Zeitera, that make extensive use of OpenCV. OpenCV’s deployed uses span the range from stitching streetview images together, detecting intrusions in surveillance video in Israel, monitoring mine equipment in China, helping robots navigate and pick up objects at Willow Garage, detection of swimming pool drowning accidents in Europe, running interactive art in Spain and New York, checking runways for debris in Turkey, inspecting labels on products in factories around the world on to rapid face detection in Japan.

It has C++, Python, Java and MATLAB interfaces and supports Windows, Linux, [Android](https://opencv.org/opencv/android/) and Mac OS. OpenCV leans mostly towards real-time vision applications and takes advantage of MMX and SSE instructions when available. A full-featured [CUDA](https://opencv.org/opencv/cuda/)and [OpenCL](https://opencv.org/opencv/opencl/) interfaces are being actively developed right now. There are over 500 algorithms and about 10 times as many functions that compose or support those algorithms. OpenCV is written natively in C++ and has a templated interface that works seamlessly with STL containers.

**2. LITERATURE REVIEW**

Gagandeep Kaur, Ritesh Sinha, Puneet Kumar Tiwari, Srijan Kumar Yadav, Prabhash Pandey, Rohit Raj, Anshu Vashisth, Manik Rakhra (2021) has revealed a paper on mask recognition system using CNN model [7]. This technique is often enforced within the retail retailers and therefore the result is often seen on the digital and promotional screens. Though many case studies are listed to demonstrate the period situation of the COVID-19 issue, the preparation of the systems in period is extraordinarily tough. Developing a system that's adaptative to any or all contexts and surroundings is turning into a problem.  
If we have a tendency to contemplate the price estimation for implementing the project, it'll be virtually of no price as most of the metropolitan cities have already got cameras put in publicly places. Camera; that is the solely main demand of the planned model is already on the market. Their model is predicated on neural networks. A neural network may be a network OR circuit of neurons, that is additionally referred to as an artificial neural network and is formed of artificial neurons or nodes. This model is often used for various functions associated with image process in neurobiology mistreatment dataset containing pictures associated with that task. This approach provides not solely helps in achieving high exactness however additionally enhance the face detection tempo significantly. The system is often applied in several areas like subway stations, markets, schools, railway stations and lots of alternative huddled places to observe the group and to confirm that each one is sporting mask. Finally, this work is often used for future researchers and enthusiasts. Firstly, this model is often employed in any high-definition camcorders, this can check that that this model isn't restricted to solely mask detection system.

Secondly, this will be used for biometric scans with a mask on the face.  
  
Abd El-Aziz, Nesrine A. Azim, Mahmood A. Mahmood and Hamoud Alshammari has revealed a paper on deep learning model for mask detection (2021) [8]. The system will expeditiously discover faces that area unit partly occluded (either with a mask or hair or hand). Supported the occlusion degree of 4 regions (nose, mouth, chin and eye) it differentiates between annotated mask and face lined by hand. Therefore, a mask covering the face absolutely together with nose and chin can solely be treated as “with mask” by the model. The most challenges two-faced by the tactic primarily comprise of variable angles and lack of clarity. The movement of blurry faces within the video stream makes it tougher. However, following the trajectories of many frames of the video helps to form a far better call – “with mask” or “without mask”.  
In this paper, they in brief explained the motivation of the work 1st. Then, they illustrated the educational and performance task of the model. Using basic ML tools and simplified techniques the tactic has achieved fairly high accuracy. In future, the model is often extended to discover if an individual can wear the mask properly (as educated by WHO) and additionally to discover the sort of mask.  
  
Safa Teboulbi, Seifeddine Messaoud, Mohamed Ali Hajjaji and Abdellatif Mtibaa (2021) has planned a paper on period Implementation of AI-Based mask Detection and Social Distancing measuring System for COVID-19 prevention [9]. This work reviewed, firstly, several Analysis works that obtain to surround COVID-19 natural event. Then, it processed the essential ideas of deep CNN models. After that, this paper reproduced the coaching and testing of the foremost used deep pretrained-based CNN models (DenseNet, InceptionV3, MobileNet, MobileNetV2, ResNet-50, VGG-16, and VGG-19) on the mask dataset. Finally, and when evaluated the numerical results, best models are tested on an embedded vision system consisted of Raspberry Pi board and digital camera where efficient real-time deep learning-based techniques are implemented with a social distancing task to change the method of detective work cloaked faces and desecrated or maintained distance between peoples.  
This embedded application are often employed in any operating atmosphere like public place, station, company atmosphere, streets, searching malls, and ex- amination centers, wherever accuracy and exactness are extremely desired to serve the aim. In future works, they're going to exploit this system on good sensors or connected RP nodes that may be thought-about as an Edge Cloud to gather multimedia system knowledge, e.g., an autonomous drone system, which may give capture (by the camera) of the detected objects from totally different angles and send them to the Edge Cloud system to be analyzed.  
  
Eashan Adhikarla and Brian D. Davison (2021) has planned a paper on mask Detection on Real-World Webcam images [10]. They conferred a new webcam-based dataset that reflects real-world complexness. They tested 12 totally different models to know their effectivity. They additionally used 3 models to label the remaining knowledge to match foreseen mask usage trends and with another supply of information. The WFM dataset is efficacious for potential COVID-19 connected studies and offers diversity for AI-related datasets as this is often the primary digital camera dataset with face masks that has been collected. The dataset provides a real-world challenge for developing higher AI models, incorporating real-world masks for face detection and face mask detection tasks, and may be a collection of 10 months of captured pictures, a tiny low portion

of that has been hand-labeled. Normally mask detection

algorithms area unit divided in 2 tasks;

(1) detective work the faces in an exceedingly given image, and (2) then classifying the image as a cloaked or no-masked face that's a binary classification task. They have a tendency to outline a further third category to replicate uncertainty or once the mask isn't worn properly. This work may be a kind of image classification and extraction. Their system is often employed in any public place, restaurant, airport etc.

Mr.Kalla.Kiran, Bokka Vamsi Kiran, Devarapalli Cheswanth Sai , Gaggala Vijay Vamsi, Pitta Rani Salomi (2021) has planned a paper on mask detection using machine learning [11]. The experimental analysis shows that the planned technique will be with success exploited for mask violation detection. It's a true time package application which may be deployed in good cc tv police investigation, public areas like airports, malls, etc. wherever mask is dominant. Simply, the package will be extensible to figure together with different IOT devices to deny allow or closing doors at company workplace. What is more, we have a tendency to highlight that it's operating additionally on device with restricted machine capability and it's able to method in real time pictures and video streams, creating their proposal applicable within the world. Taking in to account higher than mentioned details, they will build the conclusion that the Mask detection project works in real time and be terribly helpful in gift scenario. This application is put up using python, python IDLE.

The project proposed by us use OpenCV, Tensor Flow and deep learning to detect the face mask. Our goal is to create a custom deep learning model to detect whether a person is wearing a mask or not. This system first detect the image of a person. Then it will detect the COVID-19 face mask on the person’s face.

The classifier that obtained is ~98% accurate. Then it will classify whether the person is wearing a mask or not. If not then it will generate an alert. The main purpose of this system is we use this system in primary schools for the security of the children.

**2.1 Literature Review Summary**

Table 2.1: Literature review summary

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Year and citation** | **Article Title** | **Purpose of the study** | **Tools/ Software used** | **Comparison of technique done** | **Source (Journal/ Conference)** | **Findings** | **Data set (if used)** | **Evaluation parameters** |
| 2021 | Face mask recognition system using CNN model | Mask Detection | Python Script, TensorFlow and CNN | Comparison of Deep Neural Network Models of Face Mask Detection in Multi-Angle Head Pose | Kaggle, LinkedIn, Google | They investigate optimal parameter values for the CNN Model to identify of Masks accurately without generating overfitting | Kaggle, Prajna Bhandari’s Data Set | Evaluation done on basis of Face features to tell apart mask wearers. |
|  | A Deep Learning Model for Face Mask Detection |  | TensorFlow, Keras, CNN, OpenCV |  |  | effective model for real-time monitoring using CNN |  |  |
|  | Real-Time Implementation of AI-Based Face Mask Detection and Social Distancing Measuring System for COVID-19 Prevention |  | CNN Model |  |  | implementing a Face Mask and Social Distancing Detection model as an embedded vision system |  |  |
|  | Face Mask Detection on Real-World Webcam Images |  | CNN, R-CNN,  SVM |  |  | implement state-of-the-art object detection algorithms to understand their effectiveness in such a real-world application. |  |  |
|  | FACE MASK DETECTION USING MACHINE LEARNING |  | Pycharm, Notepad++, Jupyter(Python) |  |  | it is working also on device with limited computational capability and it is able to process in real time images and video streams, making our proposal applicable in the real world. |  |  |
|  |  |  |  |  |  |  |  |  |

# PROBLEM FORMULATION

Since the infectious coronavirus sickness (COVID- 19) was initial rumored in urban center. The COVID-19 pandemic has been one in every of the most important health crisis. COVID-19 epidemic has fleetly discontinued our every-day lives’ moving the international trade and movements. However, carrying a mask that stops the transmission of droplets within the air associated maintaining an acceptable physical distance between folks, and reducing shut contact with one another will still be useful in combating this pandemic. However, guaranteeing all folks wear facemask isn't a straightforward task. So, for the same issue we came up with the idea of making this project. The purpose of the project “Detecting Face Mask using Artificial Intelligence, Machine Learning and Deep Learning for COVID-19 prevention” is to create a tool that identifies the image of a human that can calculate the probability that he/she wearing a mask or not, using tools like TensorFlow, Kera, OpenCV and Scikit-Learn.

From the literature review, it is observed that studies highlight the need of efficient and scalable approach for detecting mask on peoples face. The existing techniques are not precise enough for doing the same. We proposed our project with Machine Learning, Artificial Intelligence and Deep Learning.

# RESEARCH OBJECTIVES

The proposed work is aimed to carry out work leading to the development of an approach for Detection of Mask. The proposed aim will be achieved by dividing the work into following objectives:

1. To load mask dataset and train facemask classifier with TensorFlow then serializing it to disk followed by loading it up from disk to detect face in image.
2. Extracting each face ROI to determine “Mask” or “No Mask”.
3. To learn various machine learning, deep learning techniques. And to enhance our knowledge in this aspect.
4. To use clean data for more accurate results.
5. To analyze our data without any bias.
6. To gain knowledge of various python libraries and show Result.

# METHODOLOGY

The following is a quick summary of the suggested approach.

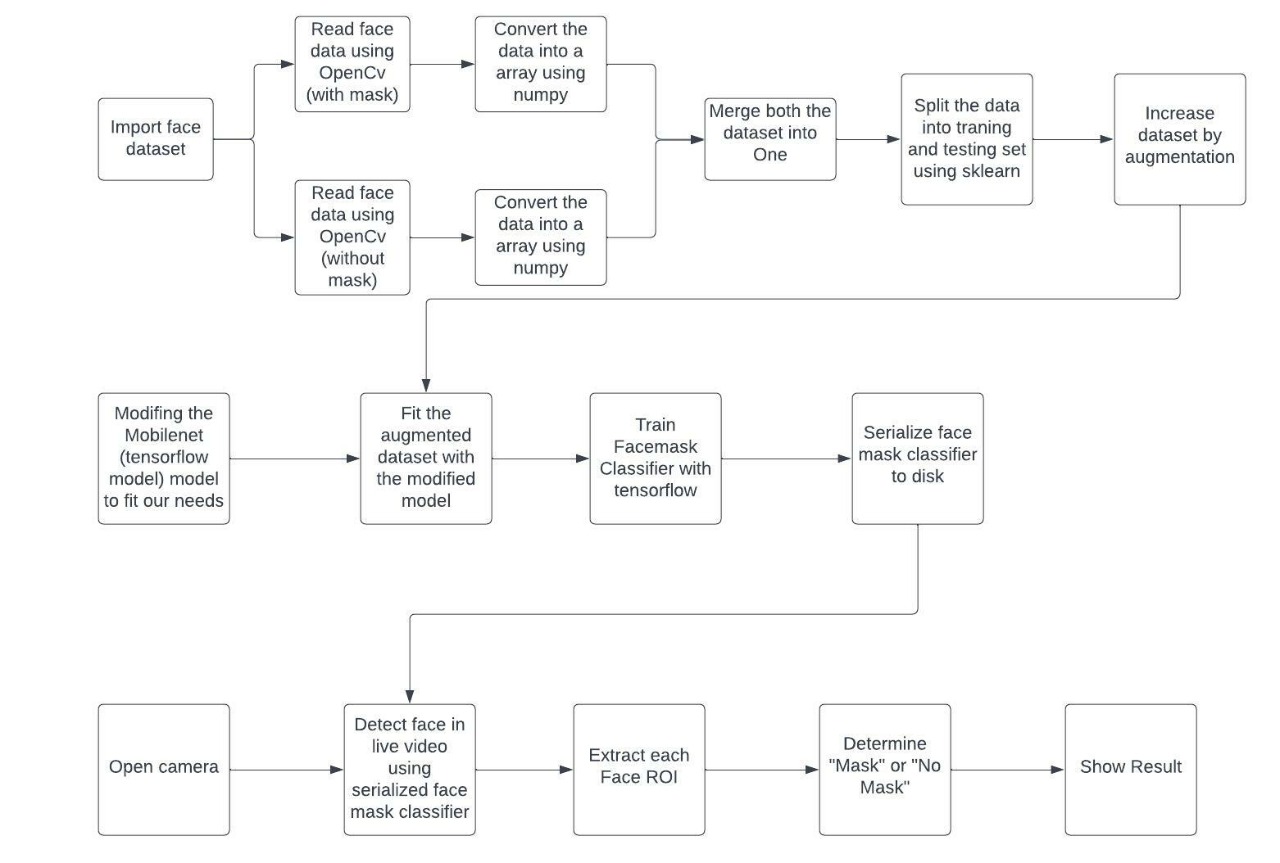
***Dataset Collection:*** Images from various sources are used to build a dataset. The size of datasets can be expanded by the application of data enhancement techniques. The photographs are stored in two files, “training dataset” and “test dataset,” each of which comprises 80 and 20% of the images, respectively. Bounding boxes, sometimes known as “data annotations,” are created around an area of interest using a variety of methods. Labeling pictures as “mask” or “NO mask” will be done using the LabelImg tool in the proposed system.

***Image Enhancement***: To draw attention to the foreground elements, the image is improved through preprocessing methods and segmentation techniques.

***Model Implementation***: We ran the tests on an Intel Core i7 processor with an Nvidia GTX 1,080 graphics card and Windows 10. Python 3.5 was used as the programming language in this project.

***Training the model:*** To distinguish between those wearing “masks” and those who aren't, the model is trained in an online GPU environment called Google Colab. A folder referred to as “the trained folder” is used for training purposes.

***Prediction***: Using the test folder, the model is tested for its ability to identify and classify masks and no-masks that were found in the original photos.



The above figure depicts the training and deployment phases of our face detection model. The dataset is loaded first in the training phase. Training and modeling are streamlined during the training phase. After serializing face mask classifier to the disk, model is loaded to detect the face mask on the images or real-time video. The model will calculate the ROI (Region of Interest) for the determination. We then compute bounding box value for a particular face and ensure that the box falls within the boundaries of the image. We then determine the class label based on predictions returned by the mask detector model and colors are assigned for interpretation. Once all detection is executed, we will display the output.

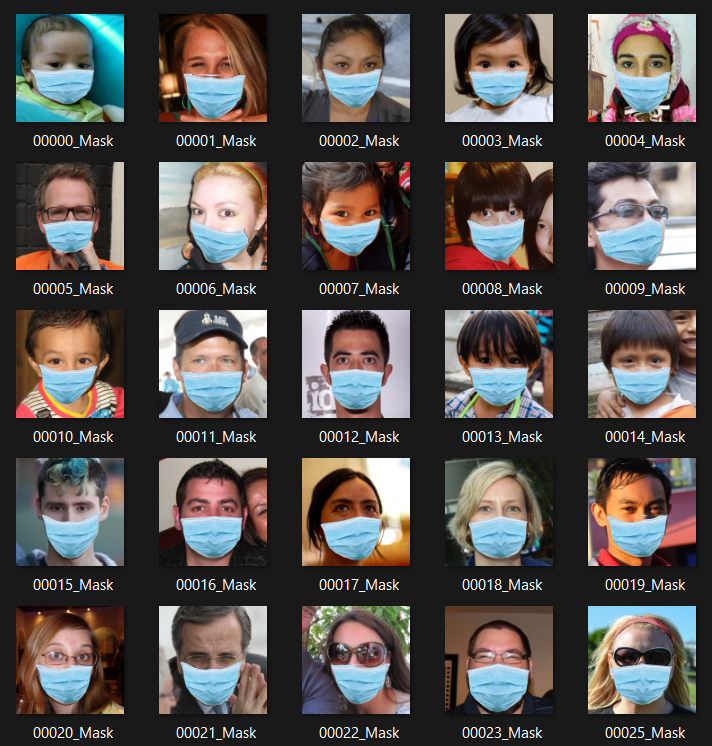
#### Data Collection

All phases of image analysis research necessitate the use of data, from training algorithms to assessing their performance. The following datasets were employed in this study:

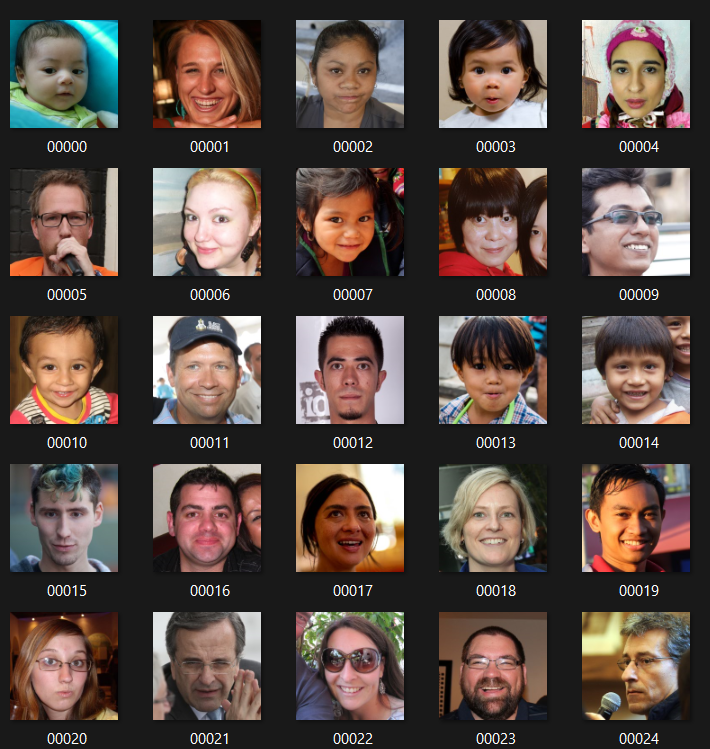
The dataset is taken from arxiv.org and consists of 2,309 images belonging to two classes:

* with\_mask: 1125 images
* without\_mask: 1184 images

***Train/Test Subsets****:* In order to develop a CNN model, you need a large amount of data. As the number of photos in the dataset grows, so does the model's accuracy. A training dataset and a test dataset each comprise 80 and 20% of the photos, respectively. A sample listing is shown in figure A and figure B.



**Figure A**



**Figure B**

#### Preprocessing

Pre-processing methods and picture segmentation are used to improve the input image in order to draw attention to the foreground items. For this, we pre-process all of the photographs in the folders and adjust the height and width dimensions to 224 × 224, respectively, to make our data more consistent and also because it is the dimension recognized by MobileNet. The photos were saved in an array format from the “Keras.preprocessing.image” module, which is necessary while using MobileNet designs. One-hot encoding was accomplished by utilizing LabelBinarizer for the attributes (tags) “mask” and “no-mask,” which were needed while using MobileNet models.

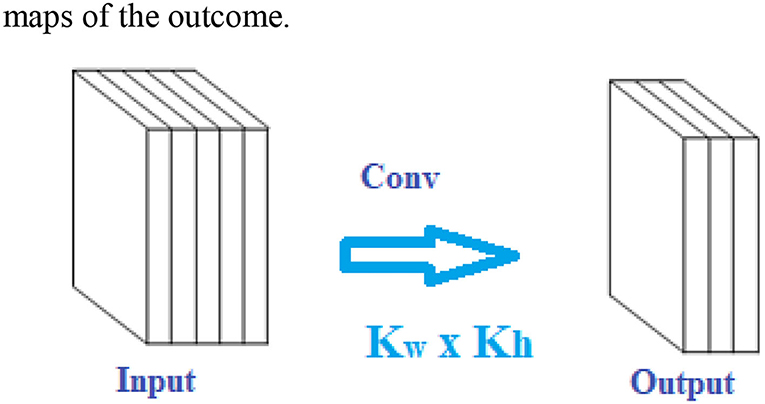
***Image Labelling***: Dataset labelling is a process that involves drawing rectangles over a region of investigation using a range of tools. The LabelImg tool is used in the proposed system to identify the pictures as “Mask” or “NO Mask,” depending on their content.

#### Classic vs. Depthwise Separable Convolutions

This work makes use of a Depthwise Separable Convolutional Neural Network based on MobileNet for classification. To construct the Depthwise Separable Convolutional Neural Network based on MobileNet, we'll go over the techniques employed in this part. Depthwise separable convolutions are being proposed to replace the currently expensive convolutional layers used in image recognition software. Weights and calculation time are both reduced using Depthwise separable convolution. There is an overview of the formulas and fundamental components of the approaches, followed by a detailed description of the proposed approach for effective classification of face photos in the COVID-19 scenario.

#### Classic Convolution

In deep learning, the traditional convolution is often referred as the standard or classic convolution. Figure C  depicts the fundamental operations of standard convolution.



**Figure C**

Classic convolutional comprises of two steps: first, a depthwise convolution layer filters the input, and then a 1×1 (or pointwise) convolution operation integrates the filtered values to create innovative features.

In a typical convolutional layer with *Xin* input channels and *Xout* output channels, each output feature map is the sum of the *Xin* input feature maps twisted by the *Xin* corresponding kernel.

A standard convolution has the following weights:

     Wstd=Cin × Kw × KH× Cout  Wgtstd=Xin × Kw × KH× Xout    (1)

Wstd=Cin × Kw × KH× Cout  Wgtstd=Xin × Kw × KH× Xout    (1)

The kernel size is denoted by the symbols *Kw* × *KH*.

Generating outcome feature maps of dimension *fw* × *fH* has a computational load of:

CCoststd=Xin × Kw × KH× Xout ×  fw × fH    (2)

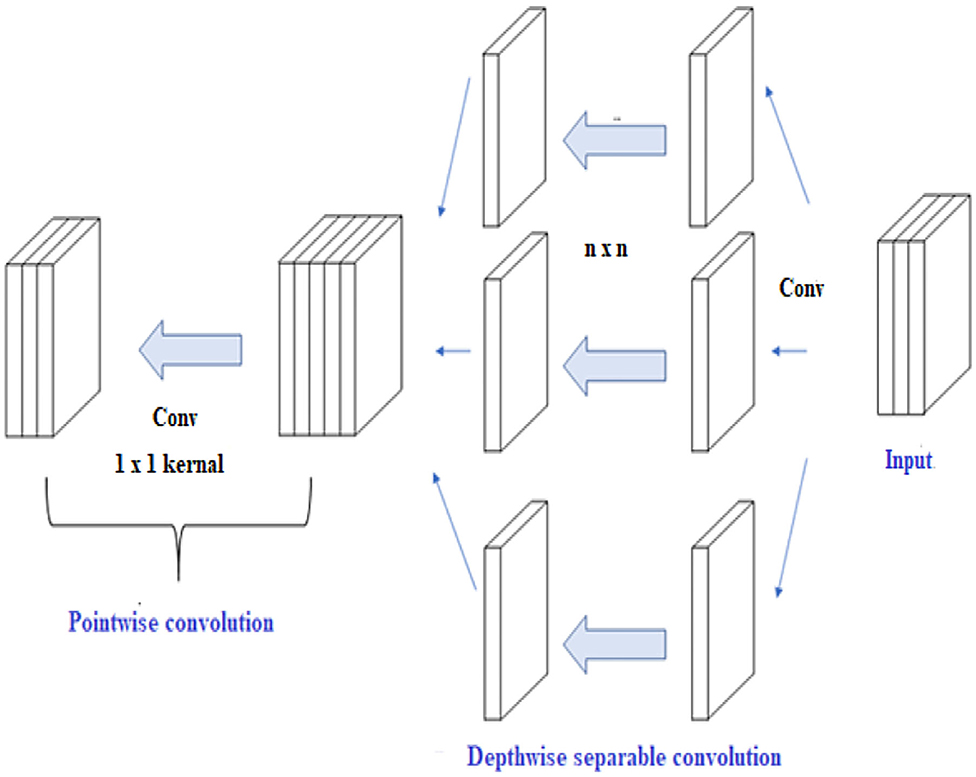
CCoststd=Xin × Kw × KH× Xout ×  fw × fH    (2)

Where Kw and KH are the spatial dimensions (height and width) of the kernels, Xin and Xout are the count of input and output streams, and fw and fH are the spatial measurements maps of the outcome.

#### Depthwise Separable Convolutional Neural Network Based on Mobile Net Architecture

In this section, we'll go over the depth-separable filters that form the foundation of MobileNet. The Depthwise Separable Convolutional Neural Network based on MobileNet is next described in detail. The MobileNet CNN ([1](https://www.frontiersin.org/articles/10.3389/fpubh.2022.855254/full" \l "B1)) design is a form of the CNN model which can be used for developing deep neural networks in cellular systems. In terms of efficiency, it is a very effective way of building convolutional neural networks. One of the things that distinguishes it from other similar products is the use of Depthwise separable convolution.

***Depthwise and Pointwise Convolutions:*** The multi-layered features, as well as the contrast between conventional and depth-wise separable, are depicted in Figure D. As illustrated in Figure D, the Depthwise (dw) and pointwise (pw) convolutions are merged to create a “Depthwise separable” convoluted structure. The Depthwise separable convolutional structure provides a function comparable to traditional convolution but at a much faster rate. Because the frames are Depthwise separable, there's also no pooling layer between them in the given technique. A stride of two is included in a couple of the depth-wise layers to reduce spatial dimension. The collection of output channels is also included in the following pointwise layer in this case.



**Figure D**

Figure D shows the fundamental methods of Depthwise convolution and Depthwise separable convolution. In contrast to conventional convolution, Depthwise convolution creates only one output feature space from a single input matrix modified by a single convolution operation.

Wgtdws=Kw × KH× Xout    (3)Wgtdws=Kw × KH× Xout    (3)

The expense of computing a Depthwise convolution layer is as follows:

CCostdws= Kw × KH× Xin× fw× fH +Xin × Xout                  ×  fw × fH    (4)

CCostdws= Kw × KH× Xin× fw× fH +Xin × Xout                  ×  fw × fH    (4)

Employing Depthwise convolution, the weighting and calculation cost are decreased by Xin times.

A DWS convolution is similar to a traditional convolution because it decreases the computational complexity by a proportion of α, which would be denoted by Equation (5).

α=CCostreduced=CCostdwsCCoststd    (5)

α=CCostreduced=CCostdwsCCoststd    (5)

α=Kw×KH×Xout×fw×fH+Xin×Xout×fw×fHXin×Kw×KH×Xout×fw×fH    (6)

α=Kw×KH×Xout×fw×fH+Xin×Xout×fw×fHXin×Kw×KH×Xout×fw×fH    (6)

α=1Xin+1Kw × KH    (7)

α=1Xin+1Kw × KH    (7)

The calculations can be further illustrated as follows:

The depth-wise separable convolution has two parts: depth-wise and point-wise convolution. It uses depth-wise convolution to apply a singular filtering on all transmissions of input vectors. The depth-wise convolution is expressed by Equation (8).

F(a,b,i)=∑v=1m∑v=1mM(v,v,i)× N(a+v−1, v−1,i)    (8)

F(a,b,i)=∑v=1m∑v=1mM(v,v,i)× N(a+v-1, v-1,i)    (8)

wherein m represents depth-wise convolutional kernels of size m x m x cin and cin symbolizes convolutional kernels of dimension m x m x cin. The nth filter in M is deployed to the nth channel in N to create the nth channel of the filtration outcome feature vector F.

A point-wise convolution uses 1x1 convolution to determine the linearly separable clustering of the depth-wise convolution outcome for generating new features. A point-wise convolution is expressed by Equation (9).

P(a,b,j)=∑i=1cinM(v,v,i)× Q(i,j)    (9)P(a,b,j)=∑i=1cinM(v,v,i)× Q(i,j)    (9)

The size of the 1x1 convolutional kernel is **1 × 1 ×** ***Xin*** **×** ***Xout***. Changing m changes the total number of channels in the output feature vector. The dense 1×1 convolutional function, like the m x m (m > 1) convolutional functions, has no requirement for being close to the vicinity, therefore changing the configuration in memory is not required. After then, the operation is carried out horizontally utilizing very efficient fundamental matrix multiplication algorithms. A DWS convolution computation is expressed by Equation (10):

Cdws=m2 . cin .  h.  w+xin. xout.  h . w     (10)Cdws=m2 . cin .  h.  w+xin. xout.  h . w     (10)

It expresses the expense of convolution layer and 1x1 point-wise convolutional computations.

The connecting strengths in a Depthwise convolution are as follows: The percentage n is generally equivalent to 1/k2 since the magnitude of m is frequently rather big. Since this study employs 3×3 DWS convolution layers, the computational cost and parametric densities of comparable convolution operation are 7 to 8 times smaller than conventional convolution operation.

**RESULTS AND DISCUSSION**

* The Model is complete using OpenCV, TensorFlow and able to distinguish person wearing mask or not.
* Accuracy:98%
* We have made a web application linked to the model to show in runtime whether a person is wearing a mask or not.

Measures must be taken to control the spread of the COVID19 pandemic. This face mask recognition system is a very good and efficient way to do so. The system will separate the people from the crowd who are not wearing mask. The identification of people, violating the COVID norms increases the adaptability of the face mask detection system for the public sake. If applied in a correct way, the face mask detection system could be used to make sure our safety and for others too. This approach gives not only helps in achieving high precision but also enhance the face detection tempo considerably. The system can be applied in many areas like metro stations, markets, schools, railway stations and many other crowded places to monitor the crowd and to ensure that everyone is wearing mask. Finally, this work can be used for future researchers and enthusiasts. Firstly, this model can be used in any high-definition camcorders, this will make sure that this model is not limited to only face mask detection system. Secondly, this can be used for biometric scans with a mask on the face.

# REFERENCES

[1] M.Loey, G.Manogaran, M.H.N.Taha, and N.E.M.Khalifa, “A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of theCOVID-19 pandemic,” *Measurement*, vol.167,ArticleID108288,2021.

[2] Z. Wang, P. Wang, P. C. Louis, L. E. Wheless, and Y. Huo, “Wearmask: fast In-browser face mask detection with serv- erless edge computing for COVID-19,” 2021, https://arxiv. org/abs/2101.00784.

[3] S.K.Dey, A.Howlader, and C.Deb, “Mobile Net mask: a multi-phase face mask detection model to prevent person-to- person transmission of SARS-CoV-2,” in *Proceedings of International Conference on Trends in Computational and Cognitive Engineering*, pp.603–613,Springer,Dhaka,Bangladesh, December2021.

[4] “Coronavirus Disease 2019 (COVID-19) – Symptoms”, Centers for Disease Control and Prevention, 2020. [Online]. Available:

<https://www.cdc.gov/coronavirus/2019-ncov/symptomstesting/symptoms.html.2020>

[5] M. Jiang, X. Fan and H. Yan, “RetinaMask: A Face Mask detector”, arXiv.org, 2020. [Online]. Available: <https://arxiv.org/abs/2005.03950.2020>

[6] M.Perc, M.Ozer, andJ.Hojnik, “Social and juristic challenges of artificial intelligence,” *Palgrave Communications*, vol.5, no. 1,pp.1–7,2019.

[7] Gagandeep Kaur ∗, Ritesh Sinha, Puneet Kumar Tiwari, Srijan Kumar Yadav, Prabhash Pandey, Rohit Raj, Anshu Vashisth, Manik Rakhra “Face mask recognition system using CNN model”, 2021.

[8] A. A. Abd El-Aziz††, Nesrine A. Azim††, Mahmood A. Mahmood†, ††, and Hamoud Alshammari† “A Deep Learning Model for Face Mask Detection”, 2021.

[9] Safa Teboulbi , Seifeddine Messaoud , Mohamed Ali Hajjaji and Abdellatif Mtibaa “Real-Time Implementation of AI-Based Face Mask Detection and Social Distancing Measuring System for COVID-19 Prevention” Article id- Volume 2021, Article ID 8340779, 21 pages, 2021.

[10] Eashan Adhikarla Brian D. Davison “Face Mask Detection on Real-World Webcam Images”, 2021.

[11] Mr.Kalla.Kiran , Bokka Vamsi Kiran , Devarapalli Cheswanth Sai , Gaggala Vijay Vamsi Pitta Rani Salomi “FACE MASK DETECTION USING MACHINE LEARNING” , 2021.

**RESEARCH PAPER**

Detecting Face Mask using AI, ML and Deep Learning for COVID-19 Prevention

Toshiba Ansari (20BCS6671), Abhishek Singh (20BCS6673), Rahul Goyal (20BCS6713) and Pushp Jain (20BCS6706), AIT-CSE Dept., Chandigarh University, Mohali, Punjab, 140413

Abstract

Since the infectious corona virus sickness (COVID- 19) was initial rumored in urban center. COVID-19 epidemic has fleetly discontinuous our every-day lives moving the international trade and movements. However, carrying a mask that stops the transmission of droplets within the air associated maintaining an acceptable physical distance between folks, and reducing shut contact with one another will still be useful in combating this pandemic. However, guaranteeing all folks wear facemask isn't a straightforward task. The purpose of the proposed model “Detecting Face Mask using Artificial Intelligence, Machine Learning and Deep Learning for COVID-19 prevention” is to create a tool that identifies the image of a human that can calculate the probability that he/she wearing a mask or not using tools like TensorFlow, Keras, OpenCV and Scikit-Learn. If there’s a violation within the scene or public places, it will generate an alarm. The system performance is evaluated in terms of precision-recall, F1-score, support, sensitivity, specificity and accuracy that demonstrate the sensible pertinency. The system performs with F1-score of 99%, sensitivity of 99%, specificity of 99% associated an accuracy of 100%. This may be used with the prevailing embedded camera infrastructure to alter these analytics which may be applied to numerous verticals, furthermore as in associate building or at airports terminals/gates.

***Keywords- Artificial Intelligence, Machine Learning, Deep Learning, Face mask, Tenser Flow, COVID-19, F1-score, Open CV, Scikit-Learn.***

Introduction

The COVID-19 pandemic emerged in December 2019 in metropolis town within the Hubei province of central China. Perceptive the virus’s growth and unfold among humans, the World Health Organization declared the corona virus (i.e., Sars-CoV-2) to be a world pandemic in March 2020. This pandemic has devastating effects on societies and economies round the world inflicting a world health crisis [1]. It's associate degree rising metastasis communicable disease caused by Severe Acute metastasis Syndrome Coronavirus2. Everywhere on the planet, particularly within the third wave, COVID-19 has been a major health care challenge [2]. Several shutdowns in numerous industries are caused by this pandemic. According to the centers for Disease Control and Prevention (CDC), corona viral infection is transmitted predominantely by respiratory droplets made once individuals breathe, talk, cough, or sneeze [2] with common drop size 5–10’m however aerosol emission will increase once humans speak and shout loudly [3]. People with COVID-19 have had a good scope of symptoms reported like shortness of breath or issue in respiratory. Elder individuals having respiratory organ unwellness are at higher risk [4] of obtaining corona virus than most. Therefore, to forestall speedy COVID-19 infection, several solutions, like confinement and lockdowns, are recommended by the bulk of the world’s governments. It's true that COVID-19 could be a world pandemic and have an effect on many domains. The importance of sporting masks be reducing vulnerability of risk from a pestilent individual throughout the “pre-symptomatic” amount to restrain the spreading of the virus. More than five million cases were infected by COVID- 19 in less than 6 months across 188 countries. The virus spreads through close contact and in crowded and overcrowded areas. We can tackle and predict new diseases by the help of new Technologies such as artificial intelligence, IOT, Big data, and Machine learning.

It created a path for researchers in engineering. We've got seen multiple analysis topics, like making new automatic detection ways of COVID-19 and detection individuals with or while not masks. Before corona virus, some individuals place masks to shield themselves from pollution, whereas others place face masks to cover their faces and their emotions from others. Protection against corona virus could be a necessary counter live, per the WHO [1]. Indeed, sporting a mask is a good methodology of obstruction 80 of all metastasis infections [2]. Several organizations enforce mask rules for the non-public protection. Checking manually if people coming into a corporation are sporting masks is cumbersome and probably conflicting. It's crucial to watch mask usage across numerous regions to adequately offer info to policy manufacturers and epidemiologists UN agency project the progress of the irruption. As a results of COVID-19, the necessity has arisen to develop associate degree economical mask detection algorithmic rule to trace mask usage in inhabited areas. A way to see mask usage while not any spreading the virus is to watch the publicly accessible webcams in bulk and examine the faces for masks.

A mask detector system is enforced to envision this. Mask detection means that to spot whether or not someone is sporting a mask or not. the primary step to acknowledge the presence of a mask on the face is to observe the face, that makes the strategy divided into 2 parts: to observe faces and to observe masks on those faces. Deep learning has been won’t to establish UN agency isn't sporting the facial mask mistreatment Convolutional neural network. It's various applications, like autonomous driving, education, police work, and so on [5]. The approach is ascendable, safe to execute, and provides a much bigger image of mask usage within the world. There are several detector systems developed round the world and being enforced. However, all this science desires optimization; a stronger, a lot of precise Detector, as a result of the world cannot afford to any extent further increase in corona cases. Considering AI legal issues and advantages in combating COVID-19 pandemic, AI technique-based solutions are still associate degree open window for development and legal interpretation [6]. the sphere of AI (AI) analysis has advanced considerably in recent years, particularly within the space of machine learning. Any fresh developed technology is indivisible from the term AI. While not AI it's terribly tough today to form any vital progress in terms of technical innovation. AI is being thought of because the next huge issue that may amendment the world hugely.

Literature Review

Gagandeep Kaur, Ritesh Sinha, Puneet Kumar Tiwari, Srijan Kumar Yadav, Prabhash Pandey, Rohit Raj, Anshu Vashisth, Manik Rakhra (2021) has revealed a paper on mask recognition system using CNN model. This technique is often enforced within the retail retailers and therefore the result is often seen on the digital and promotional screens. Though many case studies are listed to demonstrate the period situation of the COVID-19 issue, the preparation of the systems in period is extraordinarily tough. Developing a system that's adaptative to any or all contexts and surroundings is turning into a problem.  
If we have a tendency to contemplate the price estimation for implementing the project, it'll be virtually of no price as most of the metropolitan cities have already got cameras put in publicly places. Camera; that is the solely main demand of the planned model is already on the market. Their model is predicated on neural networks. A neural network may be a network OR circuit of neurons, that is additionally referred to as an artificial neural network and is formed of artificial neurons or nodes. This model is often used for various functions associated with image process in neurobiology mistreatment dataset containing pictures associated with that task. This approach provides not solely helps in achieving high exactness however additionally enhance the face detection tempo significantly. The system is often applied in several areas like subway stations, markets, schools, railway stations and lots of alternative huddled places to observe the group and to confirm that each one is sporting mask. Finally, this work is often used for future researchers and enthusiasts. Firstly, this model is often employed in any high-definition camcorders, this can check that that this model isn't restricted to solely mask detection system. Secondly, this will be used for biometric scans with a mask on the face.  
  
Abd El-Aziz, Nesrine A. Azim, Mahmood A. Mahmood and Hamoud Alshammari has revealed a paper on deep learning model for mask detection (2021). The system will expeditiously discover faces that area unit partly occluded (either with a mask or hair or hand). Supported the occlusion degree of 4 regions (nose, mouth, chin and eye) it differentiates between annotated mask and face lined by hand. Therefore, a mask covering the face absolutely together with nose and chin can solely be treated as “with mask” by the model. The most challenges two-faced by the tactic primarily comprise of variable angles and lack of clarity. The movement of blurry faces within the video stream makes it tougher. However, following the trajectories of many frames of the video helps to form a far better call – “with mask” or “without mask”.  
In this paper, they in brief explained the motivation of the work 1st. Then, they illustrated the educational and performance task of the model. Using basic ML tools and simplified techniques the tactic has achieved fairly high accuracy. In future, the model is often extended to discover if an individual can wear the mask properly (as educated by WHO) and additionally to discover the sort of mask.  
  
Safa Teboulbi, Seifeddine Messaoud, Mohamed Ali Hajjaji and Abdellatif Mtibaa (2021) has planned a paper on period Implementation of AI-Based mask Detection and Social Distancing measuring System for COVID-19 prevention [9]. This work reviewed, firstly, several Analysis works that obtain to surround COVID-19 natural event. Then, it processed the essential ideas of deep CNN models. After that, this paper reproduced the coaching and testing of the foremost used deep pretrained-based CNN models (DenseNet, InceptionV3, MobileNet, MobileNetV2, ResNet-50, VGG-16, and VGG-19) on the mask dataset. Finally, and when evaluated the numerical results, best models are tested on an embedded vision system consisted of Raspberry Pi board and digital camera where efficient real-time deep learning-based techniques are implemented with a social distancing task to change the method of detective work cloaked faces and desecrated or maintained distance between peoples.  
This embedded application are often employed in any operating atmosphere like public place, station, company atmosphere, streets, searching malls, and ex- amination centers, wherever accuracy and exactness are extremely desired to serve the aim. In future works, they're going to exploit this system on good sensors or connected RP nodes that may be thought-about as an Edge Cloud to gather multimedia system knowledge, e.g., an autonomous drone system, which may give capture (by the camera) of the detected objects from totally different angles and send them to the Edge Cloud system to be analyzed.  
  
Eashan Adhikarla and Brian D. Davison (2021) has planned a paper on mask Detection on Real-World Webcam images. They conferred a new webcam-based dataset that reflects real-world complexness. They tested 12 totally different models to know their effectivity. They additionally used 3 models to label the remaining knowledge to match foreseen mask usage trends and with another supply of information. The WFM dataset is efficacious for potential COVID-19 connected studies and offers diversity for AI-related datasets as this is often the primary digital camera dataset with face masks that has been collected. The dataset provides a real-world challenge for developing higher AI models, incorporating real-world masks for face detection and face mask detection tasks, and may be a collection of 10 months of captured pictures, a tiny low portion of that has been hand-labeled. Normally mask detection algorithms area unit divided in 2 tasks; (1) detective work the faces in an exceedingly given image, and (2) then classifying the image as a cloaked or no-masked face that's a binary classification task. They have a tendency to outline a further third category to replicate uncertainty or once the mask isn't worn properly. This work may be a kind of image classification and extraction. Their system is often employed in any public place, restaurant, airport etc.

Mr. Kalla. Kiran, Bokka Vamsi Kiran, Devarapalli Cheswanth Sai, Gaggala Vijay Vamsi, Pitta Rani Salomi (2021) has planned a paper on mask detection using machine learning [11]. The experimental analysis shows that the planned technique will be with success exploited for mask violation detection. It's a true time package application which may be deployed in good cc tv police investigation, public areas like airports, malls, etc. wherever mask is dominant. Simply, the package will be extensible to figure together with different IOT devices to deny allow or closing doors at company workplace. What is more, we have a tendency to highlight that it's operating additionally on device with restricted machine capability and it's able to method in real time pictures and video streams, creating their proposal applicable within the world. Taking in to account higher than mentioned details, they will build the conclusion that the Mask detection project works in real time and be terribly helpful in gift scenario. This application is put up using python, python IDLE.

The project proposed by us use OpenCV, Tensor Flow and deep learning to detect the face mask. Our goal is to create a custom deep learning model to detect whether a person is wearing a mask or not. This system first detect the image of a person. Then it will detect the COVID-19 face mask on the person’s face. The classifier that obtained is ~98% accurate. Then it will classify whether the person is wearing a mask or not. If not then it will generate an alert. The main purpose of this system is we use this system in primary schools.

Proposed Work

The proposed model is based on Machine Learning, Artificial Intelligence and Deep Learning.

"AI" or "Artificial intelligence" is about making a computer that can mimic human intelligence and mimic actions. The human-like machine is the ai. AI is rapidly becoming a popular field of computer science and has transformed human life in many ways. AI includes Natural Language Processing (NLP), Robotic, Expert System etc. AI techniques include sensory networks, cognitive intelligence, evolutionary computing and mixed artificial intelligence. Some of the ai apps

• Sick housing

• Finance.

***Some of the challenges of ai***

• Model durability

• Model transparency

• Disclosure of information

• Extrapolation

• Uncertainty

“ML” or “Machine Learning” is a field of computer science in which a machine learns from previous experiences without human intervention. Machine learning is everywhere, from our smartphone to our car. The snapchat filters we use, Alexa, google assistant, google maps are all ML apps.

***Machine learning involves three types of activities:***

• Supervised reading - Input and output system.

• UnSupervised reading - Installed program only.

• Reinforcement learning - A system that is agent-oriented and environmentally friendly.

***Some of the ML applications***

• Filter spam email

• Online fraud detection

• Forecast

• Speech recognition

• Self-driving cars

• Photo recognition etc.

***Some of the challenges of ML are***

• Implementation of state-of-the-art production technology

• Variety, volume, authentication for big data

"Deep Learning" is a subset of Machine Learning. It is a field that captures past knowledge and provides output using advanced algorithms. Deep Learning uses sensory networks designed to mimic human behavior. Deep Reading uses two layers which is why, this name got its name.

***Some of the Advanced Learning applications are***

• Chatbot

• Health care

• Automatic translation

• Voice search

***Some of the challenges of In-depth Learning are***

• Neural Network Disruption

• Ensuring Data Quality

• Data Security

• Production Garde AI

In order to overcome the problems of the existing system, the proposed system has been modified. This project aims to ensure that people adhere to basic safety principles. This is done by creating a face mask search system. The proposed approach is divided into two phases. We will add two more Python texts.

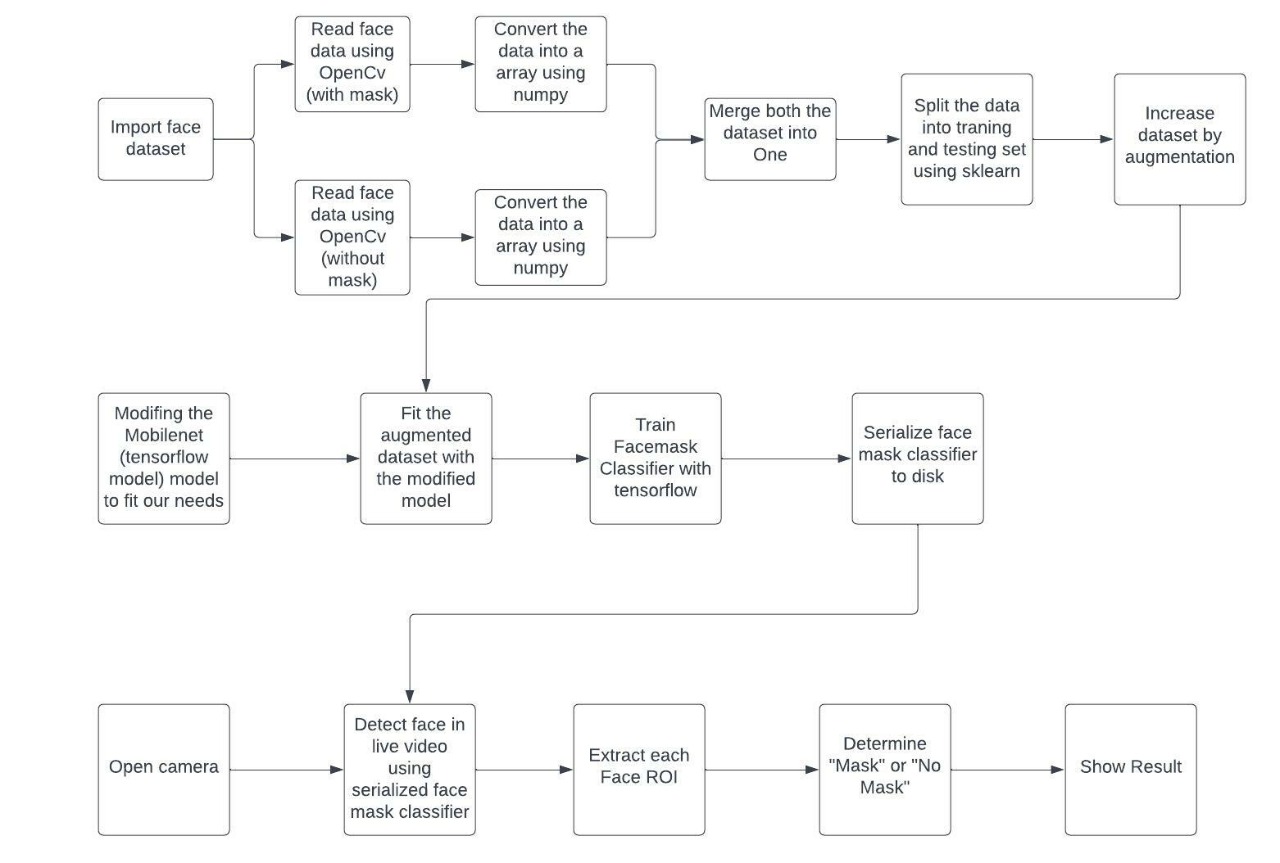
• Get COVID-19 face mask on photos

• Get a COVID-19 face mask in real-time video streaming

Figure 1 shows the whole proposed framework, in this paper, which consists of two main blocks.

**1. Training:** We load our face mask discovery data, train the model using Keras / TensorFlow on this database, and assemble the face mask detector on disk.

**2. Deployment:** Once the mask detector is trained, we can proceed to loading the mask scanner, scan the face, and classify each face as a\_mask with or without a\_ mask.

 The first dataset or Fig1 consists of 1,125 images of Adnane Cabani belonging to the class:

* with\_mask: 1125 images

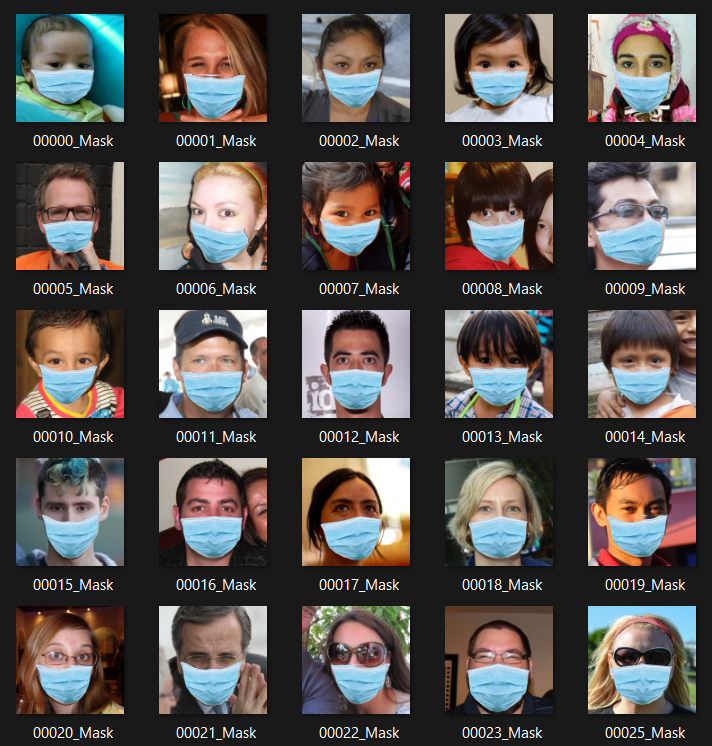


Fig1.

The second dataset or Fig2. is from NV labs consists of 1184 images and its countenances are clarified as without\_mask. In Fig. 2 some face collections are head turn, tilt and slant with multiple faces in the frame with different types and colors of masks.

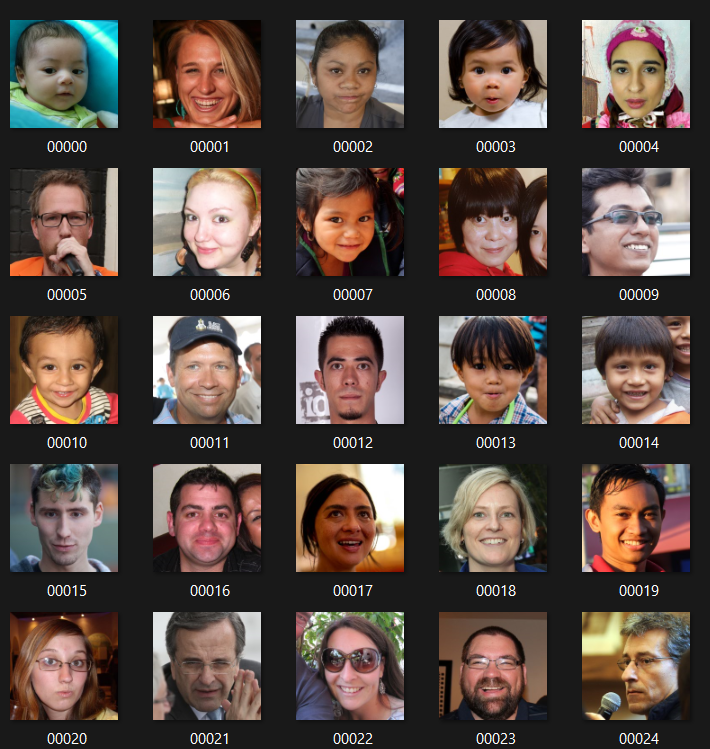


Fig2.

We can take normal images of the people and then by using computer vision we can add face mask to their face. This is easy for doing by using face recognition that helps us to infer the location of facial structures like

* Eyes
* Eyebrows
* Nose
* Mouth
* Jawline

For creating our own dataset, we need an image without mask. Then we apply face detection to get the location of bound box. Then we extract face region of interest. After that we apply facial landmarks. We need an image of a mask and by computing facial landmarks, the mask applied automatically on the image. We create our frame using QT designer and add application that open the camera for the user to detect whether the person is wearing a mask or not. We can also create confusion matrix, plots the graph or bar chart etc. by using matplotlib library that is used for data visualization. In this hard period, where mask is as important as oxygen, this project is very useful for safety of the people.

Comparison Study

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ref** | **Model Used** | **Image Type** | **Year** | **Accuracy** | **Optimization** |
| [1] | CNN Model | RGB Image | 2021 | 99.15% | YES |
| [2] | Deep Learning Model | RGB Image | 2021 | 95.77% | YES |
| [3] | CNN Model | JPEG Image | 2021 | 100% | YES |
| [4] | CNN Model | JPEG Image | 2021 | 92.64% | NO |
| [5] | Deep Learning & Machine Learning | JPEG Image | 2021 | 95% | YES |

Conclusion

Measures must be taken to control the spread of the COVID19 pandemic. This face mask recognition system is a very good and efficient way to do so. The system will separate the people from the crowd who are not wearing mask. The identification of people, violating the COVID norms increases the adaptability of the face mask detection system for the public sake. If applied in a correct way, the face mask detection system could be used to make sure our safety and for others too. This approach gives not only helps in achieving high precision but also enhance the face detection tempo considerably. The system can be applied in many areas like metro stations, markets, schools, railway stations and many other crowded places to monitor the crowd and to ensure that everyone is wearing mask. Finally, this work can be used for future researchers and enthusiasts. Firstly, this model can be used in any high-definition camcorders, this will make sure that this model is not limited to only face mask detection system. Secondly, this can be used for biometric scans with a mask on the face.

References

[1] M.Loey, G.Manogaran, M.H.N.Taha, and N.E.M.Khalifa, “A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of theCOVID-19 pandemic,” *Measurement*, vol.167,ArticleID108288,2021.

[2] Z. Wang, P. Wang, P. C. Louis, L. E. Wheless, and Y. Huo, “Wearmask: fast In-browser face mask detection

with serverless edge computing for COVID-19,” 2021, https://arxiv. org/abs/2101.00784.

[3] S.K.Dey, A.Howlader, and C.Deb, “Mobile Net mask: a multi-phase face mask detection model to prevent person-to- person transmission of SARS-CoV-2,” in *Proceedings of International Conference on Trends in Computational and Cognitive Engineering*, pp.603–613,Springer,Dhaka,Bangladesh, December2021.

[4] “Coronavirus Disease 2019 (COVID-19) – Symptoms”, Centers for Disease Control and Prevention, 2020. [Online]. Available: <https://www.cdc.gov/coronavirus/2019-ncov/symptomstesting/symptoms.html.2020>

[5] M. Jiang, X. Fan and H. Yan, “RetinaMask: A Face Mask detector”, arXiv.org, 2020. [Online]. Available: <https://arxiv.org/abs/2005.03950.2020>

[6] M.Perc, M.Ozer, andJ.Hojnik, “Social and juristic challenges of artificial intelligence,” *Palgrave Communications*, vol.5, no. 1,pp.1–7,2019.

[7] Gagandeep Kaur ∗, Ritesh Sinha, Puneet Kumar Tiwari, Srijan Kumar Yadav, Prabhash Pandey, Rohit Raj, Anshu Vashisth, Manik Rakhra “Face mask recognition system using CNN model”, 2021.

[8] A. A. Abd El-Aziz††, Nesrine A. Azim††, Mahmood A. Mahmood†, ††, and Hamoud Alshammari† “A Deep Learning Model for Face Mask Detection”, 2021.

[9] Safa Teboulbi, Seifeddine Messaoud, Mohamed Ali Hajjaji and Abdellatif Mtibaa “Real-Time Implementation of AI-Based Face Mask Detection and Social Distancing Measuring System for COVID-19 Prevention” Article id- Volume 2021,

Article ID 8340779, 21 pages, 2021.

[10] Eashan Adhikarla Brian D. Davison “Face Mask Detection on Real-World Webcam Images”, 2021.

[11] Mr.Kalla.Kiran , Bokka Vamsi Kiran , Devarapalli Cheswanth Sai , Gaggala Vijay Vamsi Pitta Rani Salomi “FACE MASK DETECTION USING MACHINE LEARNING” , 2021.