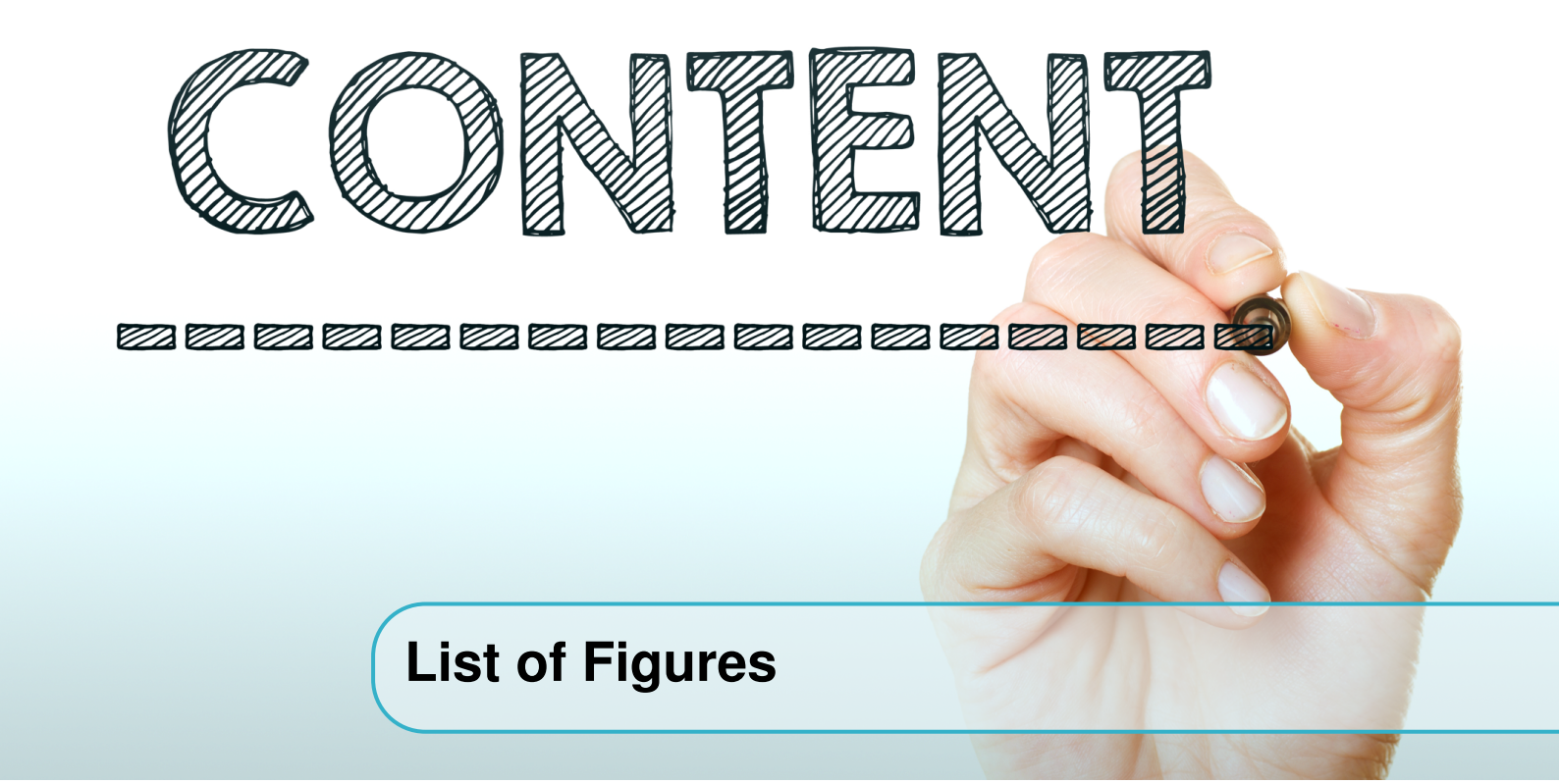
Artificial Intelligence Project

# SMART AI COVID SECURITY CAMERA

Realised by :
BELFADLA Fatima ezzahra
ELGUERRAOUI Elmehdi
RACHIDI Rabia

Supervised by :
Pr. EL KAFI Jamal


# Contents




# List of Figures

# Acknowledgment

This work would not have been possible without the support of our supervisor MR. EL KAFI Jamal. His friendly guidance and expert advice have been invaluable throughout all stages of the work.

As well as for his dynamism, vision, sincerity and motivation that have deeply inspired us. He has taught us the methodology to carry out the research and to present the research works as clearly as possible. It was a great privilege and honor to work and study under his guidance. we are extremely grateful for what he has offered us.

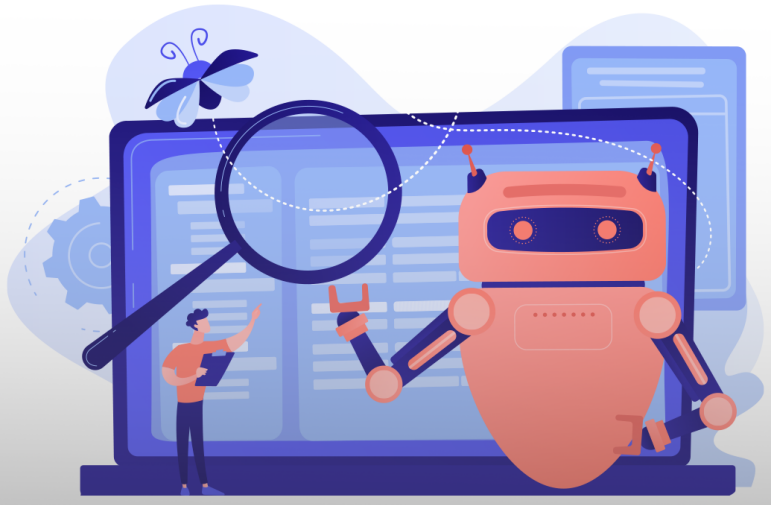
We are grateful to all of those with whom we have had the pleasure to work with during this and other related projects. Each of the members of the project's team has provided an extensive personal and professional guidance and taught us a great deal about both scientific research and life in general.

Last but not the least, We would like to thank everyone who is involved in the project directly or indirectly.


# Abstract

The trend of wearing face mask publicly is rising due to a contagious epidemic called Covid-19. This disease has affected the whole world in severe way, and the major protection method is by wearing a mask to prevent people from having this deadly disease. The main goal of this project is to reveal the presence of a face mask on human faces through a web cams and notifying the admin if individuals are not wearing the mask. Various packages of Machine Learning were used for this project such as TensorFlow, Keras, Scikit-Learn and OpenCV. Meanwhile we also adopted an architecture of deep learning to develop our face detector model. The architecture used in this context is the sequential convolution Neural Networks which is used to train the models used for this project. The proposed architecture achieved an accuracy up to 90%.

**Keywords:** Face protection masks, Convolution Neural Networks.



# 1. Introduction

The year of 2020 was quite the year with everything that went through starting by the pandemic and the lock down. *Covid*19 was the last pandemic that hit the human health within the last century.

The fast spreading of this epidemic forced all authorities to take action as fast as possible. This contagious epidemic was causing a health crisis in the world wide and his first symptoms are similar to the flu, and the effective protection method was and still is by wearing a face mask, people wear face masks once they step out of their homes.

This epidemic urged everyone in every field to step up and find a solution to fight it and prevent people's death, starting by doctors, nurses, even programmers.

Although many people are already convinced of the interest for wearing face protection mask such as suggested by the world health organization and scientific studies, we can observe that many individuals do not correctly wear their masks (see various mask wearing configurations in Fig 1.1).

To make sure that people are following the basic safety principal, A strategy should be developed, one of these strategies we have the face mask detector, which is a system that can be implemented in security cameras to check whether people are wearing their masks or not. The application should detect in real time whether the mask is worn or not.

Figure 1.1: Various configurations to the mask wearing.

Additionally, if the mask is not worn, the system will capture that person's image and send it to the admin mailbox. (figure 1.2)

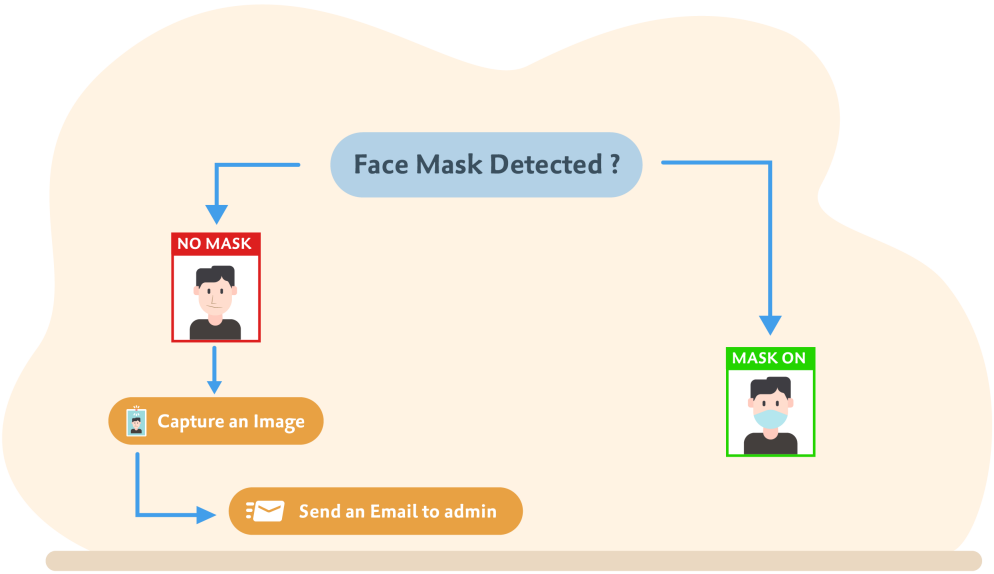



Figure 1.2: Face Mask detection process

The rest of the project report is organized as follows: **chapter 2** explores the relevant work associated with face mask detection. **chapter 3** discusses the artificial intelligence,also there is a **chapter 4** that explains neural networks, genetic algorithms and their relationship with face detection system.
**chapter 5** presents the dataset, the packages details used to build the model and an overview of the proposed model. The results and experimental analyzes are represented in **chapter 6**. and the last **chapter 7** contains conclusion & Future work.

# 2. Related work

In general, Various researchers papers handled the detection of face masks using camera acquisition systems. In this project the highlight is on the detection of the presence of mask or not since several specialized sources confirmed that wearing a mask reduces the spread of the virus .

In [1] the hanwhas were able to develop an application for detecting face masks placed in open areas, they succeeded in making a system that analyzes the flow of the camera to detect if people are wearing a face mask. For the unmasked person, the system captures an image and video, generates an alarm and reports to the manager or security personnel. For [2] ISS SecurOS created its own app aims to facilitate to owners know who enters their their buildings without a face mask required, they have focused on advanced neural network algorithms to detect a person's face in the scene and then verify if they are wearing a face ,otherwise an automatic security alert will be generated. Nizam et al [3] proposed a completely unique GAN-based network, which will automatically remove the mask covering the face area and regenerate the image by building the missing hole. Shaik et al[4] used deep learning real-time face emotion classification and recognition. Authors in [5] think occlusive face detection comes with two major challenges: 1) unavailability of sizably voluminous datasets containing both masked and unmasked faces, and 2) exclusion of facial expression in the covered area. Utilizing the locally linear embedding (LLE) algorithm and the dictionaries trained on an immensely colossal pool of masked faces, synthesized mundane faces, several mislaid expressions can be recuperated and the ascendancy of facial cues can be mitigated to great extent.

# 3. Ai & Face Mask detection system

Before getting deeper into this topic, it is very important to define some of the AI basics, starting by defining the meaning of the artificial intelligence. Artificial intelligence (AI) is wide-ranging branch of computer science concerned with building smart machines capable of performing tasks that typically require human intelligence. AI Simulate the human intelligence in machines. While addressing a crowd at the Japan AI Experience in 2017, Data Robot CEO Jeremy Achin began his speech by offering the following definition of how AI is used today «AI is a computer system able to perform tasks that ordinarily require human intelligence. . . Many of these artificial intelligent systems are powered by machine learning ,some are powered by deep learning ,and some of them are powered by very boring things like rules »

AI is an extremely powerful and exciting field. ... Artificial neural networks (ANNs) and the more complex deep learning technique are some of the most capable AI tools for solving very complex problems, and will continue to be developed and leveraged in the future.

Deep learning is a subfield of machine learning, and neural networks make up the backbone of deep learning algorithms. In fact, it is the number of node layers, or depth, of neural networks that distinguishes a single neural network from a deep learning algorithm, which must have more than three.

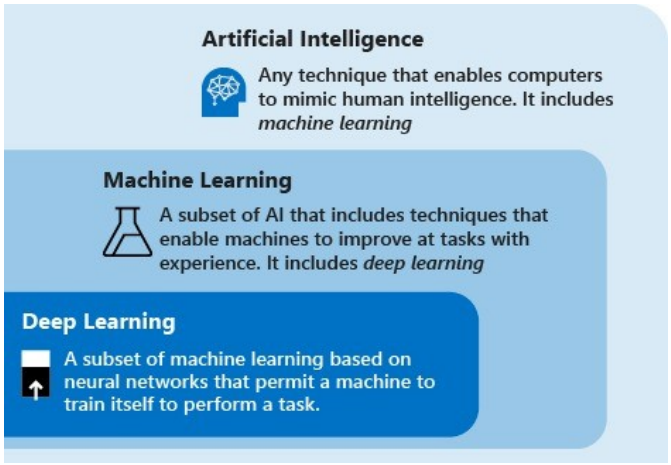
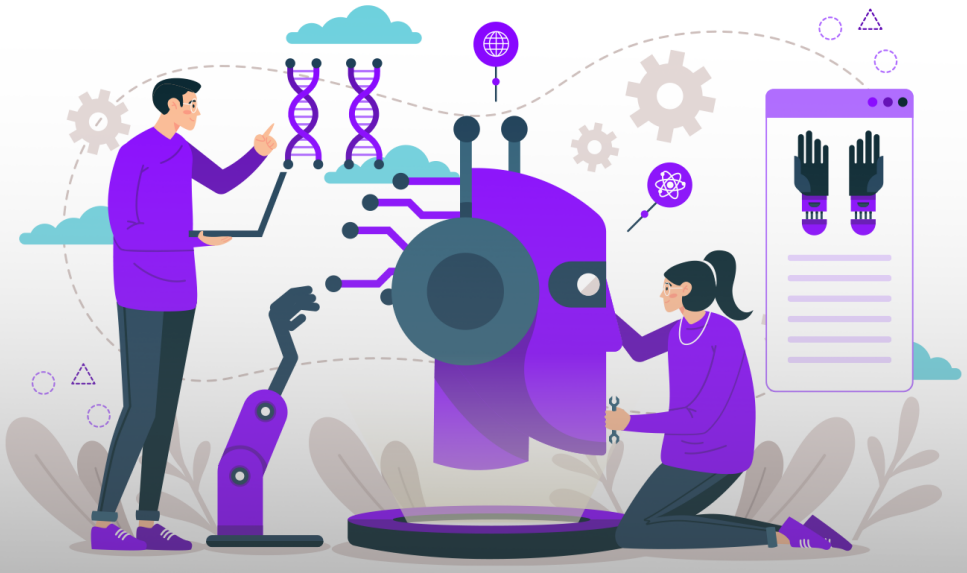



Figure 3.1: Artificial intelligence vs Machine Learning vs Deep learning

The role of Artificial intelligence which plays in the future of apps development is quite significant. It provides app developers with numerous innovative opportunities. Today, with everyone looking for ways to incorporate AI technology in the development of their apps, it has changed the natural language processing between machines and humans.

AI aids the collection and sorting of app user's data. Thereafter, it analyzes these data for behavioral and usage, and enables machines to develop human-like intelligence.

Face Mask Detection system uses Artificial Network to recognize if a person captured on camera is wearing a mask or not. If the camera captures a face without any mask, an email will be sent to the admin containing these captured images. This solution works irrespective of Gender, Age, Close view, longer view, sitting or standing positions, different types of masks, etc., this solution can be integrated with existing solutions like CCTV, access management, Drones, etc.



# 4. Neural networks & genetic algorithms

## 4.1 Neural networks

A neural network is a system that mimics the functionality of a human brain, while the brain is a neural network made up of multiple neurons, An Artificial Neural Network (ANN) is made up of multiple perceptrons.

it consists of three important layers:

- **Input Layer:** it accepts all the inputs provided by the programmer.
- **Hidden Layer:** it's a layer between the input and the output layer, where computations are performed which result in the output.
- **Output Layer:** it delivers the output performed by the hidden layer.

Considering that we want to classify images into two classes:

**Class D** containing images of diseased leaves.

**Class ND** Containing images of non-diseased leaves.

We will break down each leaf image into pixels depending on the dimension of the image. those pixels will be represented as matrices which are then fed into the input layer of our Neural Network.

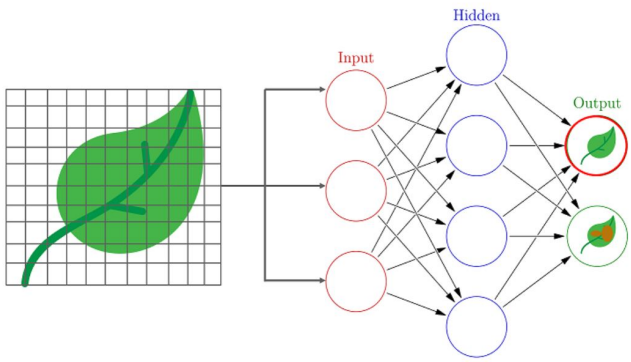



Figure 4.1: Structure of a neural network.

the input is passed from the input layer to the hidden layer, an initial random weight is assigned to each input.

the sum of the inputs multiplied with their weights is calculated and sent as an input to the hidden layer. each perceptron is passed through a transformation function that determines whether a particular perceptron gets activated or not.

At the output layer, a probability is derived which decides whether the data belongs to class D or class ND.

There is so many types of neural networks, we're mentioning some of them that are in our interest since our project is based on image(2D) and face recognition, there are some neural networks created for the analysis of such as CNN , Feed-Forward NN ...

### Image classification and CNNs

Convolutional Neural Networks (CNN) are neural networks most commonly used to analyze images. A CNN receives an image as an input in the form of a 3D matrix. The first two dimensions corresponds to the width and height of the image in pixels while the third one corresponds to the RGB values of each pixel.

CNNs consist of five sequential modules, each one may contain more than one layer :

1. Convolution
2. ReLu activation function
3. Pooling
4. Fully connected layers
5. Output layer

#### Why CNN ?

The architecture of CNN compared to other neural networks performs a better fitting to the image dataset due to the reduction in the number of parameters involved and the reusability of weights. In other words, the network can be trained to understand the sophistication of the image better.

## 4.2 Genetic Algorithms

A genetic algorithm is an heuristic search method used in artificial intelligence and computing. It is used for finding optimized solutions to search problems based on the theory of natural selection(adaptation) and evolutionary biology. It is the best way for searching through large and complex data sets. Also, it is capable of finding reasonable solutions to complex issues as it is highly capable of solving unconstrained and constrained optimization issues.

Genetic algorithms are widely used in many fields such as robotics, automotive design, optimized telecommunications routing, engineering design and computer-aided molecular design. In our case, The Genetic Algorithm

(GA) based approach is proposed for face mask recognition. It recognizes an unknown image by comparing it with other known training images stored in a database and gives information regarding whether the person wearing his mask or not. Then the genetic algorithm is compared with other known face mask recognition algorithms, and then the one with the best recognition rate is the genetic algorithm used for the application.

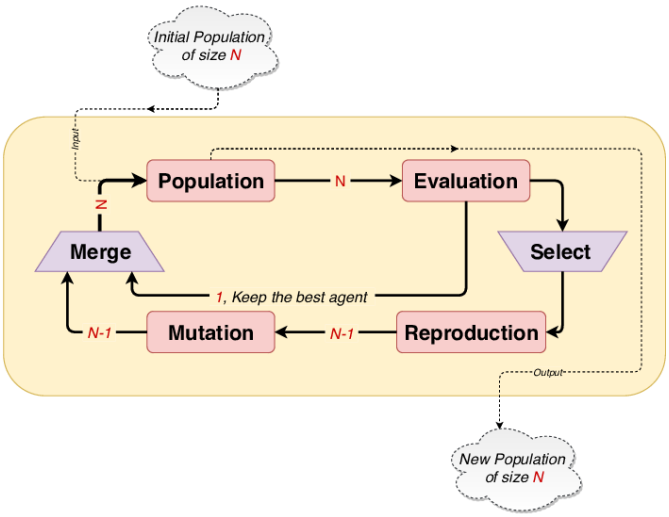



Figure 4.2: Fundamental components of a genetic algorithm.

Before start using a genetic algorithm, we have to determine how to properly encode the candidate solutions as chromosomes. Then the process starts with the random creation of the initial population which is composed of $N$ chromosomes as shown in Figure. 1. Then there is the rest of the algorithm, which is composed of the following steps applied in a loop:

1. Evaluation: quantifying the fitness of each candidate chromosome.
2. Selection of parents: reproduction based on them fitness values.
3. Usage of genetic operators (mutation, recombination) to create $N-1$ new offspring which are slightly modified versions of the mixture of their parents' chromosomes.
4. Merging $N-1$ new offspring with the previous best chromosome, in order not to decrease the overall performance due to the inherent randomness in the process.
5. If any termination criteria is not met, starting over from step 1 with the generated new population, whose size is $N$ again.

# 5. Materials and Methods

## 5.1 Dataset

The data used was provided by Prajna Bhandary. As a result, the dataset contained 1376 images belonging to two classes: **MASK FOUND** and **WEAR MASK**. Otherwise, the first class contains 690 images of people wearing masks, while the second contains 686 images of people do not wear masks. As Shown in Fig 5.1 and 5.2 the dataset mostly contains front face pose with single face in the frame and with same type of mask having different color.


Figure 5.1: Samples from Dataset including faces with masks

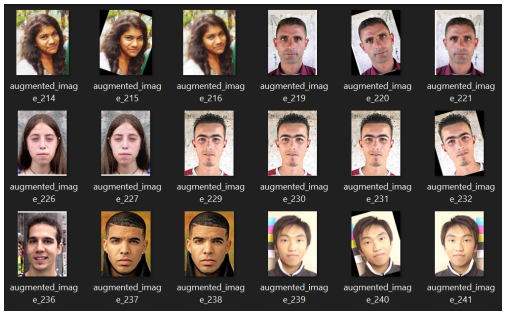



Figure 5.2: Samples from Dataset including faces without masks

## 5.2 Tools

### 5.2.1 TensorFlow

is an open-source software library for machine learning applications. TensorFlow can be used to implement neural networks and other deep learning algorithms.

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 over a bunch of areas of computer science including sentiment analysis, voice recognition, geographic information extraction, computer vision, text summarization, information retrieval, computational drug discovery and flaw detection to pursue research.

In the proposed model, the whole Sequential CNN architecture ,which consists of several layers, uses TensorFlow at backend. It is also used to reshape the data (image) in the data processing.

### 5.2.2 Keras

gives fundamental reflections and building units for creation and transportation of ML arrangements with high iteration velocity. It takes full advantage of the scalability and cross-platform capabilities of TensorFlow. The core data structures of Keras are layers and models[7] .All the layers used in the CNN model are implemented using Keras. Along with the conversion of the class vector to the binary class matrix in data processing, it helps to compile the overall model.

### 5.2.3 OpenCV

(Open Source Computer Vision Library), an open-source computer vision and ML software library, is utilized to differentiate and recognize faces, recognize objects, group movements in recordings, trace progressive modules, follow eye gesture, track camera actions, expel red eyes from pictures taken utilizing flash, find comparative pictures from an image database, perceive landscape and set up markers to overlay it with increased reality and so forth [8]. The proposed method makes use of these features of OpenCV in resizing and color conversion of data images.

### 5.2.4 Scikit-learn

(formerly scikits.learn and also known as sklearn) is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

## 5.3 Face Mask Detection Method

Before starting a bit of theoretical background first. Convolutional Neural Networks (CNN) are neural networks most commonly used to analyse images.

A CNN receives an image as an input in the form of a 3D matrix. The first two dimensions corresponds to the width and height of the image in pixels while the third one corresponds to the RGB values of each pixel. E.g., An image of 6 x 6 x 3 array of matrix of RGB (3 refers to RGB values) and an image of 4 x 4 x 1 array of matrix of grayscale image.

The adopted facial mask detection method combines between two things a cascading classifier and a pre-trained CNN containing two 2D convolutional layers linked by dense neural layers. The face mask detection algorithm is as follows:

```
Algorithm 1: Face Mask Detection
 Input: Dataset including faces with and without masks
 Output: Categorized image depicting the presence of face mask
 for each image in the dataset do
     Visualize the image in two categories and label them
     Convert the RGB image to Gray-scale image
     Resize the gray-scale image into 100 x 100
     Normalize the image and convert it into 4 dimensional array
 end
 for building the CNN model do
     Add a Convolution layer of 200-filters
     Add the second Convolution layer of 100 filter
     Insert a Flatten layer to the network classifier
     Add a Dense layer of 64 neurones
     Add the final Dense Layer with 2 outputs for 2 categories
 end
 Split the data and train the model
```

Figure 5.3: Algorithm: Face Mask Detection

### 5.3.1 Data Pre-processing

Data preprocessing is a data mining technique that involves transforming raw data into an understandable format. Real-world data is often incomplete, inconsistent, and/or lacking in certain behaviors or trends, and is likely to contain many errors. Data preprocessing is a proven method of resolving such issues. The applied model manages image and video data using Numpy and OpenCV.

#### a. data visualization

Data visualization is the graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data

The total number of images in the dataset is visualized in both categories – ‘mask found’ and ‘wear mask. The fig as follow shows how we visualized our data:

```
In [2]: import cv2,os

#path for dataset location
data_path='dataset'

#loading the folder names into CATEGORIES
categories=os.listdir(data_path)
#creating arrays for labels 0 and 1
labels=[i for i in range(len(categories))]
#dictionary for labels
label_dict=dict(zip(categories,labels))
#keys as folder names
#lables as 0 and 1
print(label_dict)
print(categories)
print(labels)
```

```
{'mask found': 0, 'wear mask': 1}
['mask found', 'wear mask']
[0, 1]
```

Figure 5.4: Data visualization

### b. Conversion of RGB image to Gray image

Based upon on all of our research, We realized that almost all of the previous Work performed on this Subject used a Grayscale representation.

So, in Conclusion, We applied this representation on our images instead of RGB Color because it simplifies the Algorithm and reduces Computational requirements.

During this event, We used cv2.cvtColor (img,code) method .This method is used to convert an image from one Color space to another. Here Code determines the type of Conversion following to our work We choose *cv2.COLOR_BGR2GRAY*.

Since we are going to use CNN architecture all images must be in the same size. Therefore, we need a fixed common size for all the images in the dataset. Using *cv2.resize()* the Gray scale image is resized into 100 x 100.


Figure 5.5: Conversion of a RGB image to a Gray Scale image of 100x100 size

#### c. Image Reshaping

Most convolutional neural networks are designed in a way so that they can only accept images of a fixed size. This creates several challenges during data acquisition and model deployment. The common practice to overcome this limitation is to reshape the input images so that they can be fed into the networks.

The images are normalized to a pixel range convergence between 0 and 1. Then they are converted to 4D matrices using:
$data = np.reshape(data, (data.shape[0], img\_size, img\_size, 1))$
where 1 represents grayscale image . Since the last layer of the neural network has 2 output MASK FOUND and WEAR MASK, that is, it contains a categorical representation, the data is transformed into categorical labels.

### 5.3.2 Model training

#### a. Building the model using CNN architecture

Convolutional neural network (CNN) is a class of deep learning methods which has become dominant in various computer vision tasks and is attracting interest across a variety of domains [12]. The current method makes use of Sequential CNN.

In most cases, a convolutional layer is followed by rectified linear units (ReLUs) and MaxPooling layers. The first convolutional layer is used to extract the different characteristics of the input images. Since the input images size are smaller than 128 x 128 so by default must be taken kernel size of 3x3 or 1x1. So, in our case we adopted 3x3.In this layer, the mathematical convolution operation is performed between the input images and 200 filters of a particular 3x3 kernel size. By dragging the filter over the input image, the dot product is caught between the filter and the parts of the input images relative to the kernel size of the 3x3 filter.

The following layers can perform a spontaneous calculation of the shape. In this case, input_shape is specified as $data.shape[1:]$ which returns the dimensions of the data array from index 1.

The padding parameter to the Keras Conv2D class can take on one of two values: "valid" or "same". With the "valid" parameter the input volume is not zero-padded and the spatial dimensions are allowed to reduce via the natural application of convolution.

If we instead want to preserve the spatial dimensions of the volume such that the output volume size matches the input volume size, then we would want to supply a value of same for the padding. While the default Keras Conv2D value is "valid", so we don't change that.

The activation parameter to the Conv2D class is simply a convenience parameter, allowing us to supply a string specifying the name of the activation function we want to apply after performing the convolution. In the following work we performed convolution and then applied a "ReLU" activation function.

Max Pooling is used to reduce the spatial dimensions of the output volume. Pool_size is set to 3 x 3 and the resulting output has a shape (number of rows

or columns) of: $shape\_of\_output = (input\_shape \check{} pool\_size + 1)/strides)$, where strides has default value $(1,1)$[13].

As shown in figure 5.6, the second convolutional layer contains 100 filters and the kernel size is fixed at 3x3. It is followed by "ReLu" and "MaxPooling" layers. For making a classification model, which means these processed data should be good input to the model. It needs to be in the form of a 1-dimensional linear vector. matrix of features can't be direct inputs. And this is why we need flattening and fully-connected layers. We flatten the output of the convolutional layers to create a single long feature vector. And it is connected to the final classification model, which is called a fully-connected layer. In other words, we put all the pixel data in one line and make connections with the final layer [14].

Dropout is used to prevent overfitting and the idea is very simple. During training time, at each iteration, a neuron is temporarily "dropped" or disabled with probability p. This means all the inputs and outputs to this neuron will be disabled at the current iteration. The dropped-out neurons are resampled with probability p at every training step, so a dropped-out neuron at one step can be active at the next one. The hyperparameter p is called the dropout-rate and it's typically a number around 0.5, corresponding to 50% of the neurons being dropped out. After flatten these convolutionals that we got from the above 2 convolutional layers. And these are finally connected to dense layer of 50 neurons with a ReLu activation function is added .then the output layers which has 2 neurons with a SoftMax activation function is added, MASK FOUND & WEAR MASK one for each.

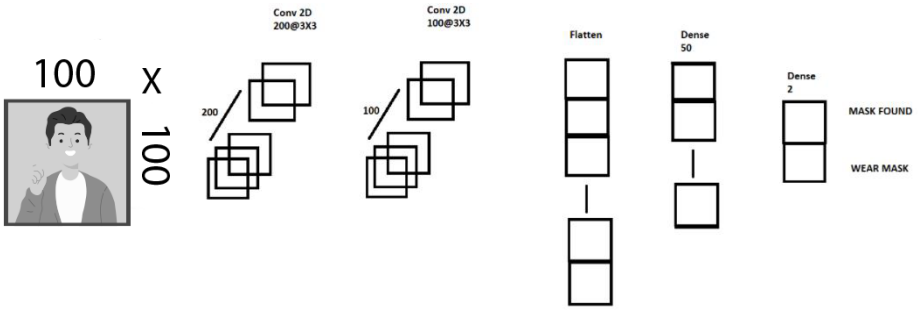



Figure 5.6: Convolutional Neural Network architecture

### b. Splitting the data and training the CNN model

Now that we have implemented a CNN architecture, our model needs to be trained and tested on a specific set of data. A proper model and an optimized *train_test_split* help produce accurate results while making a prediction. The *test_size* is set to 0.1, that is, 90% of the data in the dataset undergoes training and the remaining 10% is for testing. ModelCheckpoint callback is used in conjunction with training using model.fit() to save a model in a checkpoint file at some interval, so the model can be loaded later to continue the training from the state saved. For us the model checkpoints will be saved with the epoch number 20 and the validation loss which set in 20% in the filename.

The directory of the file path should not be reused by any other callbacks to avoid conflicts.

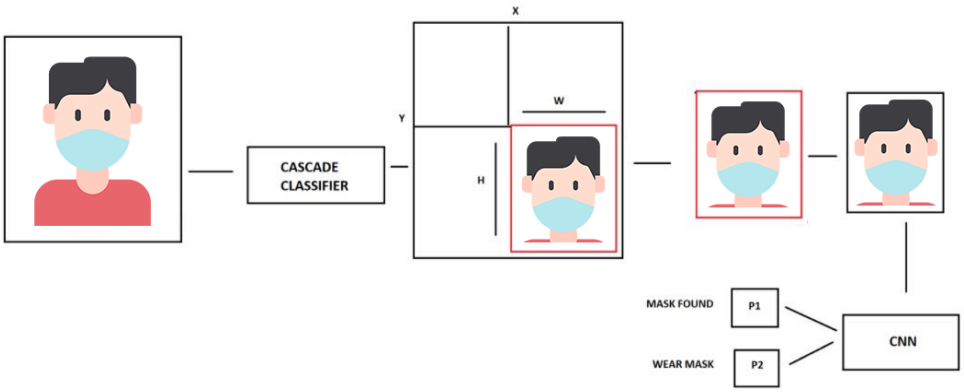
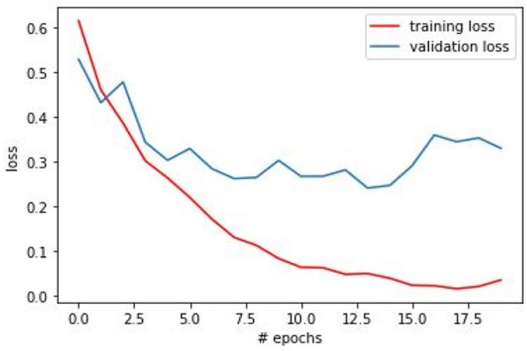



Figure 5.7: Overview of the Model

# 6. Experiments & results

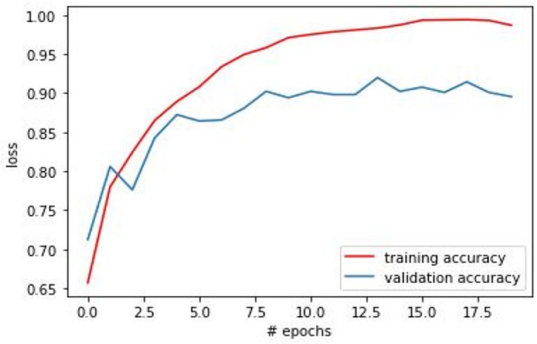
## 6.1 Performance Analysis

Let's now analyze the performance of our model. We will take a look at loss and accuracy curves, comparing training set performance against the validation set.




```
Train_loss: 0.0938 - validation_loss: 0.9805
```

Figure 6.1: epochs vs loss corresponding to our dataset




```
Train_accuracy: 0.2950 - validation_accuracy: 0.9317
```

Figure 6.2: epochs vs accuracy corresponding to our dataset

Training loss keeps going down but the validation loss starts increasing after around epoch 6. This is the textbook definition of overfitting. The model is memorizing the training data, but it's failing to generalize to new instances, and that's why the validation performance goes worse.

We are overfitting despite the fact that we are using dropout. The reason is we are training on very few examples, 1000 images per category. Usually we need at least 100K training examples to start thinking about deep learning. No matter which regularization technique we use, we will overfit on such a small dataset. But fortunately, there is a solution to this problem which enables us to train deep models on small datasets, and it's called data augmentation.

## 6.2 Results

Testing is conducted to a number of images and videos, we use our saved model from the previous training execution, and process each frame of images using cascade classifier *haarcascade_frontalface_default.xml*:

1. Extract the faces from videos using *haarcascade_frontalface_default.xml* to extract images.
2. Pass them to our face mask detector model.
3. Draw a bounding box around the detected faces, based on predictions computed by previous model.
4. if the person not wearing a mask the admin will be notified.

The following table shows the few videos and testing results. As we can see that there are 2 colors of face mask detected, face frontal. There is one failure or inaccurate detection during face heads left / right of camera, and this makes 90% or more away from camera resulting failure to recognize face mask. We assume that the training dataset does not provide enough data for non-frontal face camera images, thus leads to inaccurate trained model and prediction.


Table 6.1: Result of video testing

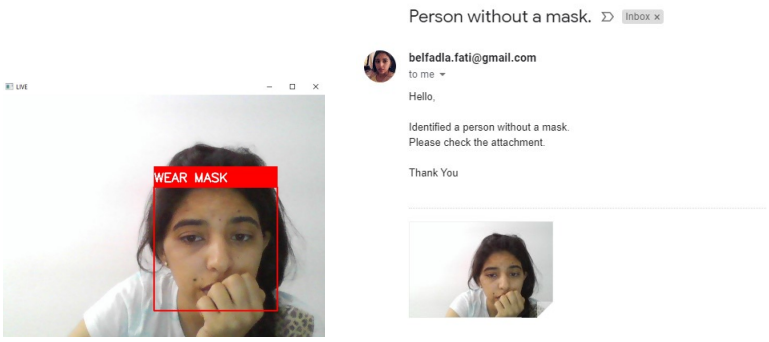



Table 6.2: WEAR MASK: Sending alert email with captured image

# 7. Conclusions & Future work

In this paper, we studied the problem of face-mask detection relevant in the scope of monitoring applications for the COVID-19 pandemic.

The model is trained on an authentic dataset. We used OpenCV, tensor flow, keras and CNN to detect whether people were wearing face masks or not. The models were tested with images and real-time video.

The accuracy of the model has achieved reasonably high score with the use of basic ML tools and simplified techniques. The deployed model will contribute immensely to the public health care system.

By the developing this system, we can detect if the person is wearing a face mask and allow their entry and their avail of public services , and it would be a great help to the society.

In future it can be extended to detect if a person is wearing the mask properly or not. The model can be further improved to detect if the mask is virus prone or not i.e. the type of the mask is surgical, N95 or not.

# References


[1] SecurOS™ Face Mask Detection , https://issivs.com/facemask/

[2] The a2 Face Mask Detection Application, https://www.hanwhasecurity.com/op-a2fmd-01.html

[3] N. Ud Din, K. Javed, S. Bae, J. YiA novel GAN-based network for unmasking of masked face IEEE Access, 8 (2020), pp. 4427644287, 10.1109/ACCESS.2020.2977386

[4] S. A. Hussain, A.S.A.A. Balushi, A real time face emotion classification and recognition using deep learning model, J. Phys.: Conf. Ser. 1432 (2020) 012087, doi: 10.1088/1742- 6596/1432/1/012087.

[5] D. Meena and R. Sharan, "An approach to face detection and recognition", 2016 International Conference on Recent Advances and Innovations in Engineering (ICRAIE), pp. 1-6, 2016.

[6] "TensorFlow White Papers", TensorFlow, 2020, [online] Available: https://www.tensorflow.org/about/bib.

[7] "Keras documentation: About Keras", 2020, [online] Available: Keras.io.

[8] "OpenCV", 2020, [online] Available: Opencv.org.

[9] https://www.intel.com/content/www/us/en/artificial-intelligence/posts/difference-between-ai-machine-learning-deep-learning.html

[10] https://www.ibm.com/cloud/blog/ai-vs-machine-learning-vs-deep-learning-vs-neural-networks

[11] https://blogs.oracle.com/bigdata/post/whatx27s-the-difference-between-ai-machine-learning-and-deep-learning

[12] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6108980/

[13] https://www.pyimagesearch.com/2018/12/31/keras-conv2d-and-convolutional-layers/

[14] https://towardsdatascience.com/the-most-intuitive-and-easiest-guide-for-convolutional-neural-network-3607be47480

[15] https://keras.io/api/callbacks/model_checkpoint/