**Summer Training Report**

**on**

**Social Distancing and Face Mask Monitor**

**A Project Report submitted in partial fulfilment of**

**the requirements for the award of**

**Bachelor of Engineering**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

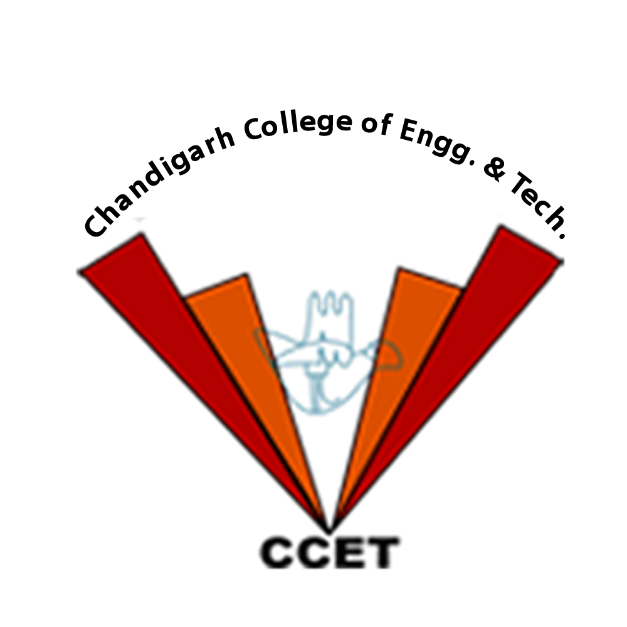
**Submitted by**

**Akhilesh Thapliyal**

**(Roll no: CO17309)**

**Under the supervision of**

**Mr. Anton Konushin and Alexey Artemov**



**CHANDIGARH COLLEGE OF ENGINEERING AND TECHNOLOGY**

**(DEGREE WING)**

Government Institute under Chandigarh (UT) Administration, Affiliated to Punjab University , Chandigarh

Sector-26, Chandigarh. PIN-160019

**August, 2020**

**CANDIDATE’S DECLARATION**

I hereby declare that the work presented in this report entitled “Social Distancing and Face Mask Monitor”, in fulfilment of the requirement for the award of the degree Bachelor of Engineering in Computer Science & Engineering, submitted in CSE Department, Chandigarh College of Engineering & Technology (Degree wing) affiliated to Punjab University, Chandigarh, is an authentic record of my/our own work carried out during my degree under the guidance of Mr. Anton Konushin and Alexey Artemov. The work reported in this has not been submitted by me for award of any other degree or diploma.

Date: August 2, 2020 Akhilesh Thapliyal

Place: Chandigarh CO17309

**CERTIFICATE**

This is to certify that the Project work entitled “Social Distancing and Face Mask Monitoring” submitted by Akhilesh Thapliyal and roll no. CO17309 in fulfilment for the requirements of the award of Bachelor of Engineering Degree in Computer Science & Engineering at Chandigarh College of Engineering and Technology (Degree Wing), Chandigarh is an authentic work carried out by him/her under my supervision and guidance.

To the best of my knowledge, the matter embodied in the project has not been submitted to any other University / Institute for the award of any Degree.

Date: August 2, 2020 Dr. Ankit Gupta

Place: Chandigarh Deptt of CSE

CCET(Degree Wing)

Chandigarh

**ACKNOWLEDGEMENT**

I would like to express My whole hearted gratitude towards my esteemed of Mr. Anton Konushin and Mr. Alexey Artemov (Coursera) for his able guidance, monitoring and encouraging attitude and searching supervision throughout my study which enable me to successfully complete my project. Without his help and valuable suggestion, this report would not have been the light of the day. The blessing, help and guidance given by him time to time shall carry me a long way in the journey of life on which I am about to embark.

I would also like to express a deep sense of gratitude to **Dr. Sunil. K Singh (Head Of Department C.S.E, CCET) and Dr. Ankit Gupta (Assistant Professor, C.S.E, CCET)** for their cordial support, valuable information and guidance which helped me in completing this task through various stages.

Lastly, I would like to thank almighty and my parents for their moral support and my friends with whom I shared my day-to-day experience and received lots of suggestions that improved my quality of work.

**ABSTRACT**

The only way to prevent the spread of COVID-19 is Social Distancing. Keeping a safe distance from each other is the ultimate way to prevent the spread of this disease at least until a vaccine is found.

I build a program that can potentially detect where each person is in real-time and return a green coloured bounding box that turns red if the distance between two people is dangerously close. It also detect whether a person is wearing a face mask or not by returning a green coloured bounding box on its face if he or she is wearing face mask and returning a red coloured bounding box on its face if he or she is not wearing face mask.

Input video stream is taken from web cam and it is processed frame by frame. From video stream single frame is taken by using OpenCV which is processed and in it a rectangle is drawn by using cv2.rectangle function and a text is shown by using cv2.putText function. After editing the frame, it is converted to Tkinter image by using Python Imaging Library and then it is showed in screen by using Label in Tkinter.

This program processes each and every frame of a video from webcam. Processing is done in two phases. First it detect face mask on the frame and draw a green or red coloured box on that frame. Second it detects persons in the frame by using yolo object detection algorithm and calculates the distance between two persons by using Euclidean distance.

This can be used by governments to analyze the movement of people and alert them if the situation turns serious.

**CONTENTS**

Candidate’s declaration…………………………………………………………....1

Certificate by the guide……………………………………………………….…...2

Acknowledgement…………………………………………………………….…...4

Abstract………………………………………………………………………….…5

Contents…………………………………………………………………….……...6

List of Figures...............................................................................................7

Chapter 1: Introduction……………………………………………….…….…......8

1.1 What is computer vision? .........................................................8

1.2 Digital Images .........................................................................9

1.3 Colour Vision and Colour Models ............................................11

Chapter 2: Data Pre-processing.......................................................................13

2.1 Dataset ..................................................................................13

2.2 Data Pre-processing ................................................................13

Chapter 3: Training the CNN ……………………………………………..………15

3.1 Introduction to Convolutional Neural Network............................15

3.2 Input Image.............................................................................15

3.3 Convolution Layer……………………………………………....…16

3.4 Flatten Layer……………………………………………..………...17

3.5 Dense Layer……………………………………………..……….. 17

Chapter 4: YOLO V3………………………………………………….….…..……19

4.1 Introduction……………………………………………..……….....19

4.2 YOLO algorithm……………………………………………..……..29

Chapter 5: Social Distancing and Face Mask Monitoring…….……….……….….20

5.1 Euclidean Distance……………………………………………..…..20

5.2 Risk Indication……………………………………………..……….20

5.3 Application……………………………………………..………......20

References……………………………………………………………………….....23

**List of Figures**

Figure 1.1 A diagram of a pinhole camera. ………………………….........................10

Figure 2.1: Data Pre-processing steps. ……………………………………………......13

Figure 3.1: Architecture of CNN. …………………………………………………..…15

Figure 3.2: Input Image. ………………………………………………………….......15

Figure 3.3: First Convolutional Layer. …………………………………………....….16

Figure 3.4: Second Convolutional Layer. ……………………………………….....…16

Figure 3.5: Flatten Layer. ………………………………………………………..........17

Figure 3.6: First Dense Layer. ……………………………………………….…..........17

Figure 3.1: Rectifier function. …………………………………………...................…17

Figure 3.8: First Dense Layer. ................................................................................…..18

Figure 3.1: Softmax function. ……………………………………………………....…18

Figure 5.1: Euclidean distance in 2D plane. ……………………………………….….20

Figure 5.2: Euclidean distance in R. ……………………………………………….…..20

Figure 5.2: Euclidean distance in 3D space. …………………….................................20

Figure 5.3: Home Page (i.e. first window). ……………………............................…..21

Figure 5.4: Social distancing and face mask monitor (i.e. fourth window). ………….22

**Chapter 1: Introduction**

**1.1 What is computer vision?**

Vision: “To see is to know what is where by looking”.

Computer vision: To design a computer model of vision, it is a part of artificial intelligence (AI).

Computer vision is a field of computer science that works on enabling computers to see, identify and process images in the same way that human vision does, and then provide appropriate output. It is like imparting human intelligence and instincts to a computer. In reality though, it is a difficult task to enable computers to recognize images of different objects.

Computer vision's goal is not only to see, but also process and provide useful results based on the observation. Computer vision is closely linked with artificial intelligence, as the computer must interpret what it sees, and then perform appropriate analysis or act accordingly.

**1.1.1 Human Vision**

We combine observation of image features and prior knowledge for image interpretation.

The human visual system is generally believed to be sensitive to visible light in the range of wavelengths between 370 and 730 nanometers (0.00000037 to 0.00000073 meters) of the electromagnetic spectrum. However, some research suggests that humans can perceive light in wavelengths down to 340 nanometers (UV-A), especially the young.

On an estimate 25% of human brain is devoted to vision.

We can come up with many helpful image features.

**1.1.2 Viola-jones face detector (2001)**

Paul Viola and Michael Jones back in 2001, the Viola-Jones Object Detection Framework can quickly and accurately detect objects in images and works particularly well with the human face (Viola & Jones, 2001). Despite its age, the framework is still a leading player in face detection alongside many of its CNNs counter parts.

**1.1.3 AlexNet (2012)**

AlexNet is the name of a convolutional neural network (CNN), designed by Alex Krizhevsky, and published with Ilya Sutskever and Krizhevsky's doctoral advisor Geoffrey Hinton.

AlexNet competed in the ImageNet Large Scale Visual Recognition Challenge on September 30, 2012.

The network achieved a top-5 error of 15.3%, more than 10.8 percentage points lower than that of the runner up. The original paper's primary result was that the depth of the model was essential for its high performance, which was computationally expensive, but made feasible due to the utilization of graphics processing units (GPUs) during training.

It is used in the followings :

1. Image recognition and annotation
2. Object detection and self-driving cars
3. Biometric methods (e.g. face recognition)
4. Image and video captioning
5. Image generation and image processing

**1.2 Digital Images**

**1.2.1 Optical Image**

Optical image, the apparent reproduction of an object, formed by a lens or mirror system from reflected, refracted, or diffracted light waves. There are two kinds of images, real and virtual. In a real image the light rays actually are brought to a focus at the image position, and the real image may be made visible on a screen e.g., a sheet of paper, whereas a virtual image cannot.

2D representation of a physical object is obtained by propagation of light through optical system, depicting object contour and features.

Different types of optical system

1. Common camera
2. Fisheye camera
3. Tomography

**1.2.2 Pinhole camera model**

Probably the first optical system, which allows human to capture the shape of objects from projected light.

The pinhole camera model describes the mathematical relationship between the coordinates of a point in three-dimensional space and its projection onto the image plane of an ideal pinhole camera, where the camera aperture is described as a point and no lenses are used to focus light.

The model does not include, for example, geometric distortions or blurring of unfocused objects caused by lenses and finite sized apertures. It also does not take into account that most practical cameras have only discrete image coordinates. This means that the pinhole camera model can only be used as a first order approximation of the mapping from a 3D scene to a 2D image

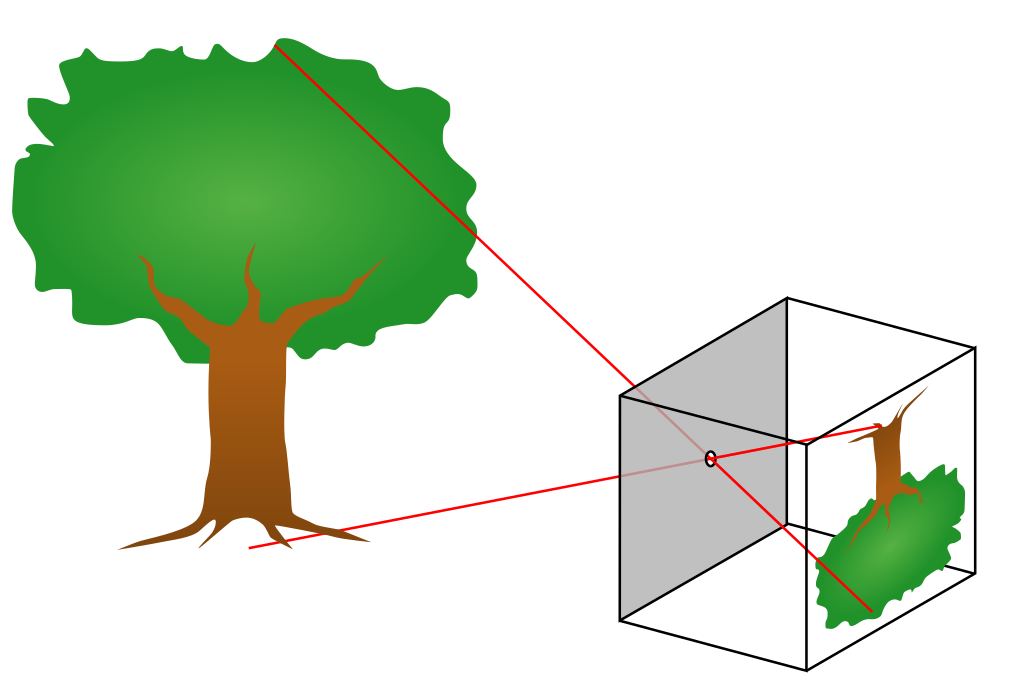


Figure 1.1: A diagram of a pinhole camera.

1. Captures pencil of rays, all rays through a single point.
2. The point is called **centre of** **projection** (focal point).
3. The image is formed on the Image Plane.

**1.2.3 Perspective distortion**

Perspective distortion is a warping or transformation of an object and its surrounding area that differs significantly from what the object would look like with a normal focal length, due to the relative scale of nearby and distant features.

Perspective distortion is determined by the relative distances at which the image is captured and viewed, and is due to the angle of view of the image (as captured) being either wider or narrower than the angle of view at which the image is viewed, hence the apparent relative distances differing from what is expected.

Perspective projection introduces distortions of some properties of real world objects.

1. Converging vertical lines.
2. The exterior columns appear bigger.

The distortion is not due to lens flaws.

**1.2.4 Modern digital camera**

A digital camera uses an electronic image sensor to create still photographs and record video. The optical system of a digital camera works like a film camera, in which a typical lens and diaphragm are used to adjust electronic image sensor lighting.

Digital cameras equip amateur and professional photographers with multiple automated control functions. Advanced digital cameras facilitate manual control of most functions.

They are based on same principle as pinhole camera, but with complex optical system and digital sensor array.

**1.2.5 Digital gray scale image**

A gray scale image is simply one in which the only colours are shades of gray. The reason for differentiating such images from any other sort of colour image is that less information needs to be provided for each pixel. In fact a `gray' colour is one in which the red, green and blue components all have equal intensity in RGB space, and so it is only necessary to specify a single intensity value for each pixel, as opposed to the three intensities needed to specify each pixel in a full colour image.

Often, the gray scale intensity is stored as an 8-bit integer giving 256 possible different shades of gray from black to white. If the levels are evenly spaced then the difference between successive gray levels is significantly better than the gray level resolving power of the human eye.

Gray scale images are very common, in part because much of today's display and image capture hardware can only support 8-bit images. In addition, grayscale images are entirely sufficient for many tasks and so there is no need to use more complicated and harder-to-process color images.

**1.3 Colour Vision and Colour Models**

**1.3.1 What is colour?**

1. Colour is the characteristic of visual perception described through colour categories.
2. This perception of color derives from the stimulation of photoreceptor cells by electromagnetic radiation (in the visible spectrum in the case of humans).
3. Colour categories and physical specifications of colour are associated with objects through the wavelengths of the light that is reflected from them and their intensities.
4. Colour is a psychological property of out visual experiences when we look at objects and lights, it is not a physical property of those objects or lights.
5. Colour is the result of interaction between physical light in the environment and our visual system.

**1.3.2 Electromagnetic spectrum**

The electromagnetic spectrum is the range of frequencies (the spectrum) of electromagnetic radiation and their respective wavelengths and photon energies.

The electromagnetic spectrum covers electromagnetic waves with frequencies ranging from below one hertz to above 1025 hertz, corresponding to wavelengths from thousands of kilometres down to a fraction of the size of an atomic nucleus.

This frequency range is divided into separate bands, and the electromagnetic waves within each frequency band are called by different names; beginning at the low frequency (long wavelength) end of the spectrum these are: radio waves, microwaves, infrared, visible light, ultraviolet, X-rays, and gamma rays at the high-frequency (short wavelength) end

Visible light is in the range of 380nm to 780nm. Visible light pass through main optical window of earth atmosphere (~46% energy). Any source of light can be completely described physically by its spectrum. The amount of energy emitted (per time unit) at each wavelength 380-780nm.

**1.3.3 Colour perception**

Colour perception is a part of the larger visual system and is mediated by a complex process between neurons that begins with differential stimulation of different types of photoreceptors by light entering the eye.

In very low light levels, vision is scotopic: light is detected by rod cells of the retina.

In brighter light, such as daylight, vision is photopic: light is detected by cone cells which are responsible for colour vision.

Rods and cones act as filters on the spectrum. To get the output of a filter, multiply its response curve by the spectrum, integrate over all wavelengths. Each cone yields one number.

**1.3.4 Trichromatic colour theory**

The Young-Helmholtz theory states that within your eye are tiny cells that can receive waves of light and translate them into one of three colors: blue, green, and red. These three colours can then be combined to create the entire visible spectrum of light as we see it.

1. Lets select 3 primary colours P1, P2, P3.
2. Other colours are represented with linear combination of primary colours.
3. Weights are colour “Coordinates”.
4. Trichromatic colour theory says that we can get all visible colours by combining three primaries.
5. We can make colour matching experiments.

**Chapter 2: Data Pre-processing**

**2.1 Dataset**

The dataset consisted of 1376 images, 690 face images with masks and 686 without masks. The original dataset is prepared by Prajna Bhandary and available at Github (<https://github.com/prajnasb/observations/tree/master/experiements/data>).

It contains images of different colours, different sizes and different orientations. Therefore before training the model with this dataset, it is necessary to perform pre-processing on the dataset.

**2.2 Data Pre-processing**

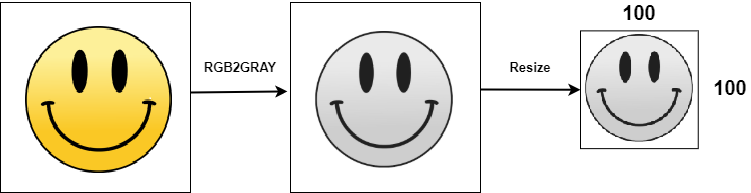
****

Figure 2.1: Data Pre-processing steps

The data pre-processing is done in the following steps.

**2.2.1 Converting the image into gray scale**

Assuming that colour is not a critical feature in detecting mask therefore I convert the image into a gray scale image to decrease the size of data in the image to save the processing time required to process each image. It is done by using cv2.cvtColor function of OpenCV library.

**2.2.2 Resizing the gray scale into 100x100**

The images have different sizes therefore it is needed to fixed common size for all the images in the dataset. It is done by using cv2.resize function of OpenCV library.

**2.2.3 Appending the data in two lists**

Appending the image in a list named data and its label or category into the list named target.

**2.2.4 Exception handling**

In some cases the image may not be available and an exception might be raised, so the exception will be printed and pass to the next image in the dataset.

**2.2.6 Normalising the data**

For normalisation the images the data list is converted into numpy array and divided by 255 to convert the pixel range into 0 and 1.

The target list which contains the label or categories is converted into numpy array. It is then converted into categorical representation by using np\_utils.to\_categorical function of keras library because last layer on convolutional neural network has two neurons, with mask and without mask.

**2.2.5 Saving the data as numpy array**

The data list which contains the images and the target list which contains the label or categories are saved in the form of numpy array, which are loaded to train the convolutional neural network.

**Chapter 3: Training the CNN**

**3.1 Introduction to Convolutional Neural Network**

Computer Vision enable machines to view the world as humans do, perceive it in a similar manner and even use the knowledge for a multitude of tasks such as Image & Video recognition, Image Analysis & Classification, Media Recreation, Recommendation Systems, Natural Language Processing, etc. The advancements in Computer Vision with Deep Learning has been constructed and perfected with time, primarily over one particular algorithm called **Convolutional Neural Network**.

A Convolutional Neural Network is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects in the image and be able to differentiate one from the other. The pre-processing required in a CNN is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, CNN have the ability to learn these filters/characteristics.

The architecture of a CNN is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex.

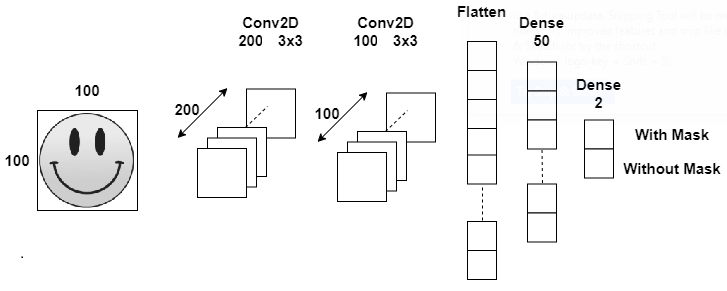


Figure 3.1: Architecture of CNN

**3.2 Input Image**

It would be computationally intensive once the images reach dimensions, say 8K (7680x4320). The role of the CNN is to reduce the images into a form which is easier to process, without losing features which are critical for getting a good prediction. This is important when we are to design an architecture which is not only good at learning features but also is scalable to massive datasets. In the above architecture, the size of input image is reduced to 100x100 pixels.



Figure 3.2: Input Image

**3.3 Convolution Layer**

The convolutional layer is the core building block of a CNN. The layer's parameters consist of a set of learnable filters, which have a small receptive field, but extend through the full depth of the input volume. During the forward pass, each filter is convolved across the width and height of the input volume, computing the dot product between the entries of the filter and the input and producing a 2-dimensional activation map of that filter. As a result, the network learns filters that activate when it detects some specific type of feature at some spatial position in the input.

Stacking the activation maps for all filters along the depth dimension forms the full output volume of the convolution layer. Every entry in the output volume can thus also be interpreted as an output of a neuron that looks at a small region in the input and shares parameters with neurons in the same activation map.

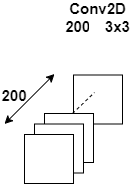


Figure 3.3: First Convolutional Layer

This layer creates a convolution kernel that is convolved with the layer input to produce a tensor of outputs. Its number of output filters in the convolution is 200 and height and width of the 2D convolution window is 3x3.

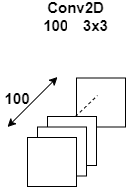


Figure 3.4: Second Convolutional Layer

In this layer the number of output filters in the convolution is 100 and height and width of the 2D convolution window is 3x3.

**3.4 Flatten Layer**

Flattening is converting the data into a 1-dimensional array for inputting it to the next layer. We flatten the output of the convolutional layers to create a single long feature vector. It is connected to the final classification model, which is called a fully-connected layer.



Figure 3.5: Flatten Layer

**3.5 Dense Layer**

Dense layer is the regular deeply connected neural network layer. It is most common and frequently used layer. Dense layer does the below operation on the input and return the output.



Figure 3.6: First Dense Layer

In first dense layer input which represent the input data is 50 and the activation function is rectifier function. Rectifier function is defined as the positive part of its argument.

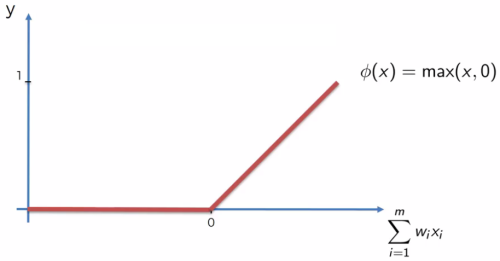


Figure 3.1: Rectifier function

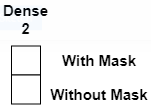


Figure 3.8: First Dense Layer

In second dense layer input which represent the input data is 2 which represent with mask and without mask. The activation function is softmax function. The softmax function, also known as softargmax or normalized exponential function, is a function that takes as input a vector z of K real numbers, and normalizes it into a probability distribution consisting of K probabilities proportional to the exponentials of the input numbers.

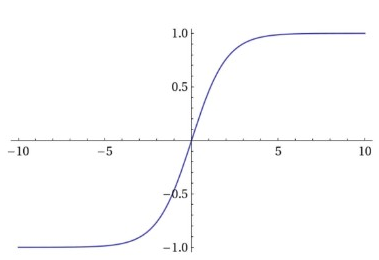
****

Figure 3.1: Softmax function

**Chapter 4: YOLO: Real-Time Object Detection**

**4.1 Introduction**

YOLOv3 predicts an objectness score for each bounding box using logistic regression. This should be 1 if the bounding box prior overlaps a ground truth object by more than any other bounding box prior.

**4.2 YOLO algorithm**

There are a few different algorithms for object detection and they can be split into two groups:

Algorithms based on classification. They are implemented in two stages. First, they select regions of interest in an image. Second, they classify these regions using convolutional neural networks. This solution can be slow because we have to run predictions for every selected region. A widely known example of this type of algorithm is the Region-based convolutional neural network (RCNN) and its cousins Fast-RCNN, Faster-RCNN and the latest addition to the family: Mask-RCNN. Another example is RetinaNet.

Algorithms based on regression – instead of selecting interesting parts of an image, they predict classes and bounding boxes for the whole image in one run of the algorithm. The two best known examples from this group are the YOLO (You Only Look Once) family algorithms and SSD (Single Shot Multibox Detector). They are commonly used for real-time object detection as, in general, they trade a bit of accuracy for large improvements in speed.

The aim of this algorithm is to predict a class of an object and the bounding box specifying object location. Each bounding box can be described using four descriptors:

1. Center of a bounding box (bxby)
2. Width (bw)
3. Height (bh)
4. Value cis corresponding to a class of an object (such as: car, traffic lights, etc.).

**Chapter 5: Social Distancing and Face Mask Monitoring**

**5.1 Euclidean Distance**

The Euclidean distance between two points in either the plane or 3-dimensional space measures the length of a segment connecting the two points. It is the most obvious way of representing distance between two points.

In two- dimensional Euclidean plane, if p = (p1, p2) and q = (q1, q2) then the distance is given by

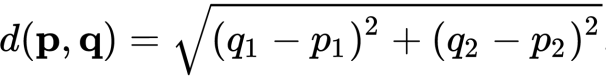


Figure 5.1: Euclidean distance in 2D plane.

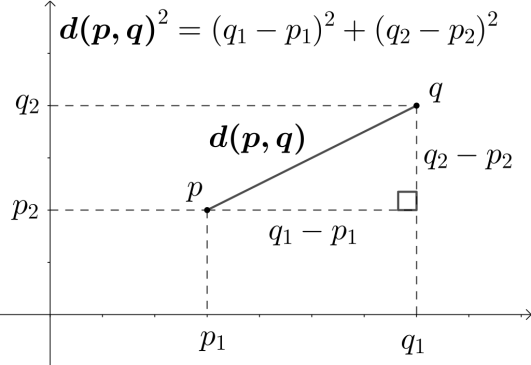
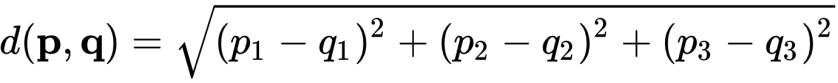


Figure 5.2: Euclidean distance in R.

In three-dimensional Euclidean space, the distance is



5.2: Euclidean distance in 3D space.

**5.2 Risk Indication**

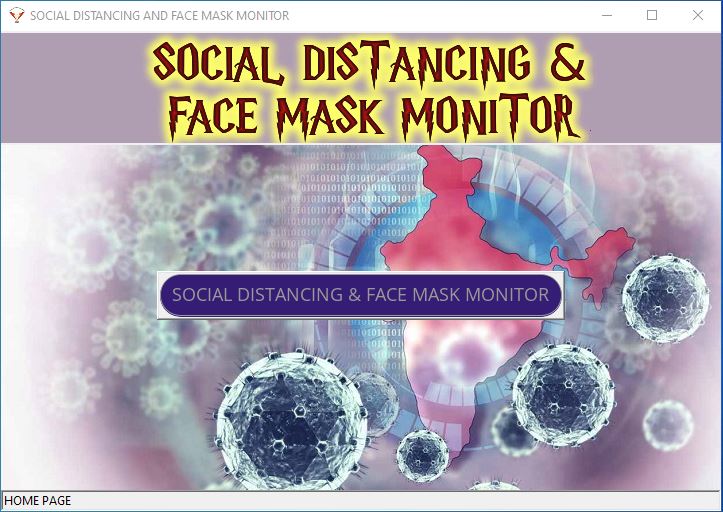
Risk is indicated by using red coloured rectangular boxes. On the top of the box if the social distancing criteria is not satisfied then “UNSAFE” is shown and if the person is not wearing a mask then “WITHOUT MASK” is shown.

If there is no risk then green coloured rectangular box is drawn in image. If the social distancing criterion is satisfied then “SAFE” is shown and if the person is wearing a mask then “WITH MASK” is shown.

**5.3 Application**

The application has four windows. Graphic User Interface of this application is made by using Tkinter library.

**5.3.1 First Window**



By. Akhilesh Thapliyal

Figure 5.3: Home Page (i.e. first window).

The above figure shows the first window of the application. In this window there are five components.

**Heading**: I show the name of project.

**Button 1**: Face mask monitor button switch the page to the second window of application which detects the persons with mask and persons without mask.

**Button 2**: Social distancing monitor button switch the page to the third window of which detects the social distancing between persons.

**Button 3**: Social distancing & face mask monitor button switch the page to the fourth window of which detects the social distancing between persons and the persons with mask and persons without mask.

**Status Bar**: This shows the name of current window.

**5.3.2 Second Window**

****

Figure 5.4: Social distancing and face mask monitor (i.e. fourth window).

The above figure shows the second window of the application which detects the social distancing between persons and the persons with mask and persons without mask. In this there are eight components.

1. **Heading**: It shows the name of current window.
2. **Screen:** Input video stream is taken from web cam and it is processed frame by frame. From video stream single frame is taken by using OpenCV which is processed and in it a rectangle is drawn by using cv2.rectangle function and a text is shown by using cv2.putText function. After editing the frame, it is converted to Tkinter image by using Python Imaging Library and then it is showed in screen by using Label in Tkinter.
3. **Button 1**: Home button, it is shown by using an image of home and it is located at the upper left corner of the screen. It switches the current window to the home page of the application.
4. **Button 2**: Close button, it is shown by using an image of cross and it is located at the upper right corner of the screen. It closes the window and releases the resources that are taken by the application.
5. **Button 3**: Camera button, it is shown by using an image of camera lens and it is located at the lower centre corner of the screen. It saves the image of screen in the current working directory.
6. **Button 4**: Face mask monitor button switch the page to the second window of application which detects the persons with mask and persons without mask.
7. **Button 5**: Social distancing monitor button switch the page to the third window of which detects the social distancing between persons.
8. **Status Bar**: This shows the name of current window.

**References**

* https://en.wikipedia.org/wiki/Convolutional\_neural\_network
* http://rosalind.info/glossary/euclidean-distance/
* https://en.wikipedia.org/wiki/Euclidean\_distance
* https://www.techopedia.com/definition/32309/computer-vision
* https://en.wikipedia.org/wiki/Viola%E2%80%93Jones\_object\_detection\_framework
* https://en.wikipedia.org/wiki/AlexNet
* https://www.britannica.com/technology/optical-image
* https://en.wikipedia.org/wiki/Pinhole\_camera\_model#:~:text=The%20pinhole%20camera%20model%20describes,are%20used%20to%20focus%20light.
* https://en.wikipedia.org/wiki/Color
* https://homepages.inf.ed.ac.uk/rbf/HIPR2/gryimage.htm