Contents

1.INTRODUCTION

The project presents facemask, object, social distance, face emotion characteristics analysis using image processing techniques for automated vision system used at agricultural field. In agriculture research of automatic characteristics detection is essential one in monitoring large fields of crops, and thus automatically detects symptoms of object characteristics as soon as they appear on plant leaves. The proposed decision making system utilizes image content characterization and supervised classifier type of neural network. Image processing techniques for this kind of decision analysis involves pre-processing, feature extraction and classification stage.

1.1 General

At Processing, an input image will be resized and region of interest selection performed if needed. Here, color and texture features are extracted from an input for network training and classification.Color features like mean, standard deviation of HSV color space and texture features like energy, contrast, homogeneity and correlation. The system will be used to classify the test images automatically to decide object characteristics. For this approach, automatic classifier NN be used for classification based on learning with some training samples of that some category. This network uses tangent sigmoid function as kernel function. Finally, the simulated result shows that used network classifier provides minimum error during training and better accuracy in classification.

1.2 Objective

The main objective is to detect the object and count the number of object by using machine learning technique. It can do it as real time video surveillance.

The main objective of this major project is to detect the Multiple Objects in space, Social Distance, Face Mask, Human Emotions detection and count the number of humans by using machine learning (ML) and AI techniques under single software using **PYTHON** programming

Language as It can be done in as real time video surveillance, with one module as android application implementation known as **“ARTIFIC SIGHT INTELLIGENCE APPLICATION,,**

1.3 Scope of Project

The main contributions of this project therefore are

- Data Analysis
- Dataset Pre-processing
- Training the Model
- Testing of Dataset

1.4 Domain Overview

Machine Learning

Machine Learning combines computer science, mathematics, and statistics. Statistics is essential for drawing inferences from the data. Mathematics is useful for developing machine learning models and finally, computer science is used for implementing algorithms.

However, simply building models is not enough. You must also optimize and tune the model appropriately so that it provides you with accurate results. Optimization techniques involve tuning the hyper parameters to reach an optimum result.

The world today is evolving and so are the needs and requirements of people. Furthermore, we are witnessing a fourth industrial revolution of data. In order to derive meaningful insights from this data and learn from the way in which people and the system interface with the data, we need computational algorithms that can churn the data and provide us with results that would benefit us in various ways. Machine Learning has revolutionized industries like medicine, healthcare, manufacturing, banking, and several other industries. Therefore, Machine Learning has become an essential part of modern industry.

Data is expanding exponentially and in order to harness the power of this data, added by the massive increase in computation power, Machine Learning has added another dimension to the way we perceive information. Machine Learning is being utilized everywhere. The electronic devices you use, the applications that are part of your everyday life are powered by powerful machine learning algorithms.

With an exponential increase in data, there is a need for having a system that can handle this massive load of data. Machine Learning models like Deep Learning allow the vast majority of data to be handled with an accurate generation of predictions. Machine Learning has revolutionized the way we perceive information and the various insights we can gain out of it. These machine learning algorithms use the patterns contained in the training data to perform classification and future predictions. Whenever any new input is introduced to the ML model, it applies its learned patterns over the new data to make future predictions. Based on the final accuracy, one can optimize their models using various standardized approaches. In this way, Machine Learning model learns to adapt to new examples and produce better results.

Types of Machine Learning

Machine Learning Algorithms can be classified into 3 types as follows –

1. Supervised Learning
2. Unsupervised Learning
3. Reinforcement Learning

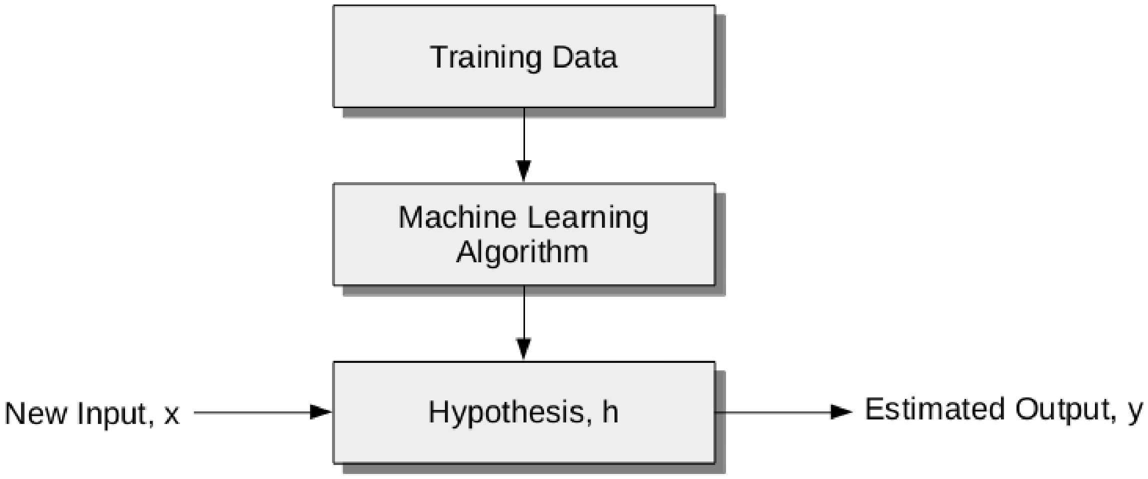


Fig 1.1 Machine Learning Process

Supervised Learning

In the majority of supervised learning applications, the ultimate goal is to develop a finely tuned predictor function h(x) (sometimes called the “hypothesis”). “Learning” consists of using sophisticated mathematical algorithms to optimize this function so that, given input data x about a certain domain (say, square footage of a house), it will accurately predict some interesting value h(x) (say, market price for said house).

$$h(x_1, x_2, x_3, x_4) = \theta_0 + \theta_1 x_1 + \theta_2 x_3^2 + \theta_3 x_3 x_4 + \theta_4 x_1^3 x_2^2 + \theta_5 x_2 x_3^4 x_4^2$$

This function takes input in four dimensions and has a variety of polynomial terms. Deriving a normal equation for this function is a significant challenge. Many modern machine learning problems take thousands or even millions of dimensions of data to build predictions using hundreds of coefficients. Predicting how an organism's genome will be expressed, or what the climate will be like in fifty years, are examples of such complex problems.

Under supervised ML, two major subcategories are:

- Regression machine learning systems: Systems where the value being predicted falls somewhere on a continuous spectrum. These systems help us with questions of "How much?" or "How many?".
- Classification machine learning systems: Systems where we seek a yes-or-no prediction, such as "Is this tumer cancerous?", "Does this cookie meet our quality standards?", and so on.

In practice, x almost always represents multiple data points. So, for example, a housing price predictor might take not only square-footage (x1) but also number of bedrooms (x2), number of bathrooms (x3), number of floors (x4), year built (x5), zip code (x6), and so forth. Determining which inputs to use is an important part of ML design. However, for the sake of explanation, it is easiest to assume a single input value is used.

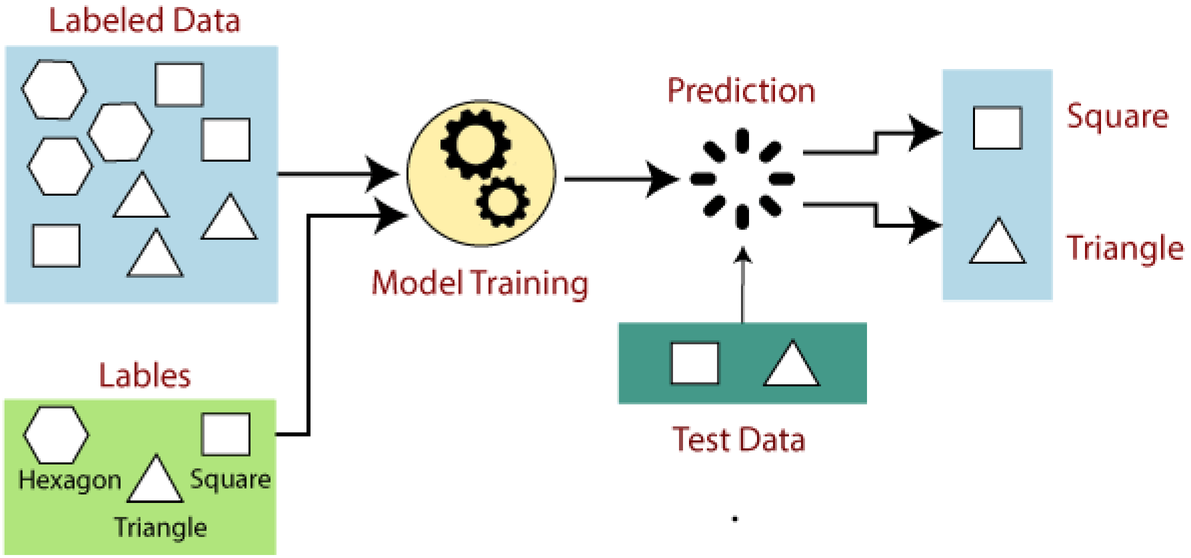


Fig 1.2: Supervised Learning

Steps Involved in Supervised Learning:

- First Determine the type of training dataset
- Collect/Gather the labelled training data.
- Split the training dataset into training dataset, test dataset, and validation dataset.
- Determine the input features of the training dataset, which should have enough knowledge so that the model can accurately predict the output.
- Determine the suitable algorithm for the model, such as support vector machine, decision tree, etc.
- Execute the algorithm on the training dataset. Sometimes we need validation sets as the control parameters, which are the subset of training datasets.
- Evaluate the accuracy of the model by providing the test set. If the model predicts the correct output, which means our model is accurate.

Regression

Regression algorithms are used if there is a relationship between the input variable and the output variable. It is used for the prediction of continuous variables, such as Weather forecasting, Market Trends, etc. Below are some popular Regression algorithms which come under supervised learning:

1. Linear Regression
2. Regression Trees
3. Non-Linear Regression
4. Bayesian Linear Regression
5. Polynomial Regression

Classification

Classification algorithms are used when the output variable is categorical, which means there are two classes such as Yes-No, Male-Female, True-false, etc.

- Spam Filtering,
- Random Forest
- Decision Trees
- Logistic Regression
- Support vector Machines

2.Literature Survey

2.1 Overall Description

Literature survey is the most important step in software development process. Before developing the tool it is necessary to determine the time factor, economy and company strength. Once these things are satisfied, then next step is to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from book or from websites. Before building the system the above considerations are taken into account for developing the proposed system.

A literature review is a body of text that aims to review the critical points of current knowledge including substantive findings as well as theoretical and methodological contributions to a particular topic. Literature reviews are secondary sources, and as such, do not report any new or original experimental work. Also, a literature review can be interpreted as a review of an abstract accomplishment.

Most often associated with academic-oriented literature, such as a thesis, a literature review usually precedes a research proposal and results section. Its main goal is to situate the current study within the body of literature and to provide context for the particular reader.

2.2 Survey Results

S.No	Title	Author Name	Year of Publish	Technology / Algorithm used	Drawbacks
1.	Multiple Real-time object identification using Single shot Multi-Box detection	Kanimozhi S	2019	Machinelearning/CNN	real time system are unlabelled data

2	Multi-object Detection Method based on YOLO and ResNet Hybrid Networks	Zhenyu Lu	2019	Deep Learning/CNN	deep learning theory cannot accurately recognize multiple objects
3	Improve object detection via a multi-feature and multi-task CNN model	Yingxin Lou	2017	Machinelearning/CNN	small-sized objects
4	MULTI-SCALE OBJECT DETECTION IN SATELLITE IMAGERY BASED ON YOLT	Wentong Li	2019	Machinelearning/R-CNN	Performance Low
5	Application of Improved Bhattacharyya Coefficient Based Multi-Object Detection and Tracking Integrated Strategy	Mingbin Zeng	2018	Image processing/Sift Algorithm	redundant detection

6	Cascade multi-scale object detection on high-resolution images	Alexey Novoselov	2019	Computervision/CNN	Low resolution
7	Fast salient object detection based on multi-scale feature aggression	Xiaohu Zhang	2019	Image processing/ CNNAlgorithm	slow speed
8	Research on Moving Object Detection and Matching Technology in Multi-Angle Monitoring Video	Wenqi Zhou	2019	Computervision/HOG Featutr Extraction	Accuracy less
9	MULTI-SCALE OBJECT DETECTION WITH FEATURE FUSION AND	Wenjie Guan	2018	Computervision/DCNN	performance of small objects and eliminating the redundant

	REGION OBJECTNESS NETWORK				
10	Motion Multi-object Detection Method under Complex Environment	Zi-xiao PAN	2016	Imageproessing/subtrsction algorithm	Low complexity
11	MULTI-VIEW FRUSTUM POINTNET FOR OBJECT DETECTION IN AUTONOMOUS DRIVING	Pei Cao	2019	Imageprocessing/CNN(using Hardware LIDAR)	Very much Expensives
12	A Deep Learning Framework Using Convolutional Neural Network for Multi-class Object Recognition	Shaukat Hayat	2018	Computervision/CNN	Low Quality
13	Geospatial Object Detection In Remote Sensing	Qunil Yao	2019	Imageprocessing/CNN	weakened simultaneously. In this paper, to

	Images Based On Multi-Scale Convolutional Neural Networks				tackle these problems
14	Multi-Task Fusion of Object Detection and Semantic Segmentation	Xiaohan Liu	2019	Computervision/RCNN	real-time performance and lower
15	3D Point Cloud Object Detection with Multi-View Convolutional Neural Network	Guan Pang	2016	Imageprocessing/CNN	time-consuming

S.No	**Title**	**Author Name**	**Year of Publish**	**Technology / Algorithm used**	**Drawbacks**

16	Artific Sight Software	Team Asia	2022	SVM/RANDOM FOREST/OPEN CV/YOLO/COCO	Multiple Frame Overlapping

List of modules

- Preprocessing
- RGB color model
- Glcm feature extraction
- Artificial neural network

COLOR CONVERSION

RGB2HSV

1) Color Feature based Retrieval:

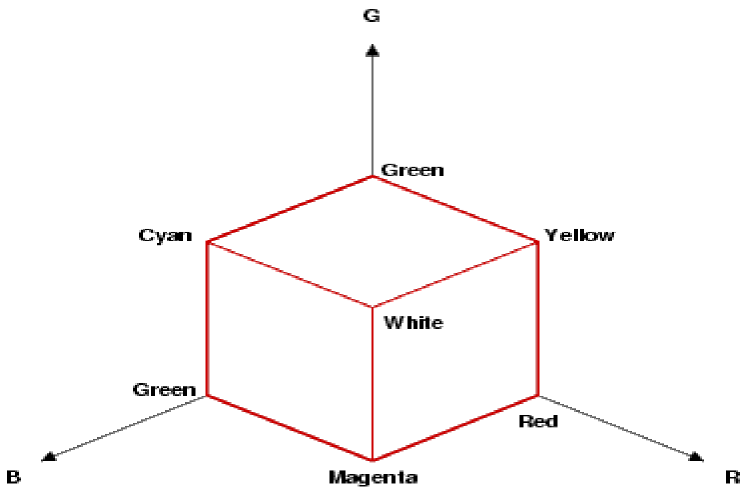
Several ways in which for retrieving photos on the premise of color similarity unit of measurement pictured at intervals the literature, however most are variations on constant basic prepare. Every image other to the gathering is analyzed to calculate a Jim Crow graph, that shows the proportion of pixels of every color at intervals the image. The color chart for every image is then keeping at intervals the information. At search time, the user will either specify the desired proportion of every color (75% inexperienced and twenty fifth red, for example), or submit Associate in Nursing example image from that a Jim Crow graph is calculated. Either way, the matching technique then retrieves those photos whose color histograms match those of the question most closely. The matching technique most typically used, chart intersection, was initial developed by beau and Ballard. Variants of this system unit of measurement presently utilized in a {very} very high proportion of current CBIR systems. ways in which of rising on beau and Ballard's original technique embody the employment of accumulative color histograms, combining chart intersection with some an area of spatial matching, and therefore the use of region-based color querying. The results from style of those systems will look quite spectacular.

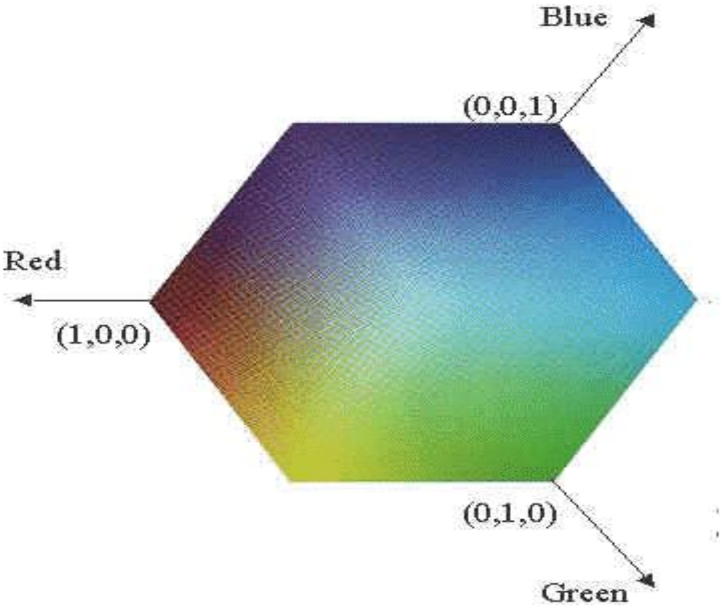
a) **RGB color model**

The RGB color model consists of the first colors Red, Green, and Blue. This system defines the color model that's utilized in most color vacuum tube monitors and color formation graphics. They are thought-about the "additive primaries" since the colors unit of measurement other on to supply the desired color. The RGB model uses the co-ordinate system as shown in Figure one. (a). Notice the diagonal from (0,0,0) black to (1,1,1) white that represents the grey-scale

.

Fig (a)





O

Fig (b)

b) HSV Colour Model

The HSV stands for the Hue, Saturation, and worth supported the artists (Tint, Shade, and Tone). The frame of reference in an extremely hexatone in Figure a combine of. (a).

And Figure a combine of.(b) a browse of the HSV colour model. the price represents intensity of a colour, that's decoupled from the colour information inside the drawn image.

The hue and saturation components area unit intimately related to the means that human eye perceives colour resulting in image method algorithms with physiological basis.

As hue varies from zero to one.0, the corresponding colours vary from red, through yellow, green, cyan, blue, and magenta, back to red, therefore there are actually red values every at zero and one.0.

As saturation varies from zero to one.0, the corresponding colours (hues) vary from unsaturated (shades of gray) to fully saturated (no white component). As value, or brightness, varies from zero to one.0, the corresponding colours become more and more brighter.

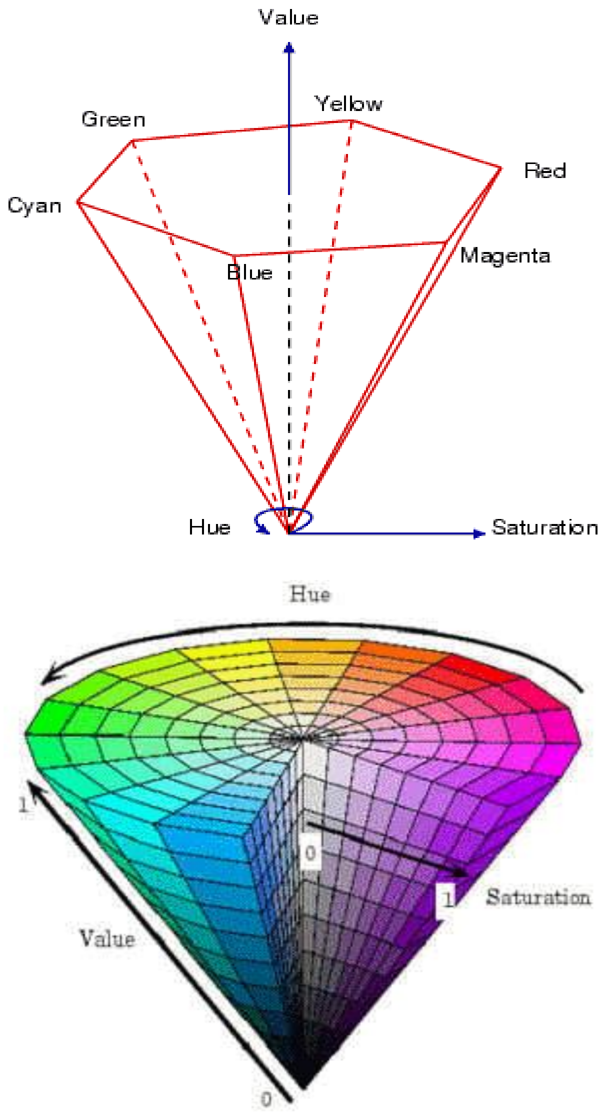


Fig (a) **Fig (b)**

Fig: (a) HSV coordinates system

(b) HSV colour model

c) Color conversion

In order to use a good colour area for a selected application, colour conversion is required between colour areas. The nice colour area for image retrieval system ought to preserve the perceived colour variations. In alternative words, the numerical geometer distinction ought to approximate the human perceived distinction.

d) RGB to HSV conversion

In Fig 4.3, the gettable HSV colours lie among a triangle whose vertices area unit outlined by the 3 primary colours in RGB space:

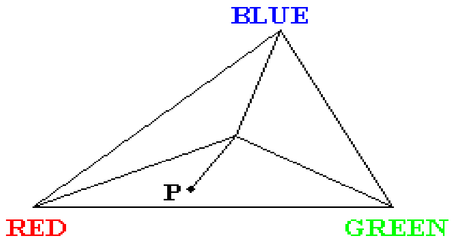


Fig: Obtainable HSV colour from RGB colour space

The hue of purpose| P is that the measured angle between the road connecting P to the Triangle center and line connecting RED point to the Triangle center. The saturation of the purpose P is that the distance between P and triangle center.The value (intensity) of the purpose P is depicted as height on a line perpendicular to the Triangle and spending through its center.

The grayscale points area unit located onto a similar line. and therefore the conversion formula is as follows:

$$H = \cos^{-1}\left\{\frac{\frac{1}{2}[(R-G)+(R-B)]}{\sqrt{(R-G)^2+(R-B)(G-B)}}\right\}, \quad S = 1 - \frac{3}{R+G+B}[\min(R,G,B)],$$

$$V = \frac{1}{3}(R+G+B)$$

e) HSV to RGB conversion

Conversion from HSV area to RGB house is additional complicated. And, given to the character of the hue data, we are going to have a unique formula for every sector of the color triangle.

<u>Red-Green Sector:</u>

$for \quad 0° < H \leq 120°$

$$b = \frac{1}{3}(1-S), \quad r = \frac{1}{3}\left[1 + \frac{S\cos H}{\cos(60° - H)}\right], \quad g = 1-(r+b)$$

Green-Blue Sector:

$for \quad 120° < H \leq 240°$

$$r = \frac{1}{3}(1-S)\ , \quad g = \frac{1}{3}\left[1 + \frac{S\cos H}{\cos(60° - H)}\right], \quad b = 1-(r+b)$$

Blue-Red Sector:

$for \quad 240° < H \leq 360°$

$$g = \frac{1}{3}(1-S)\ , \quad b = \frac{1}{3}\left[1 + \frac{S\cos H}{\cos(60° - H)}\right], \quad r = 1-(r+b)$$

HISTOGRAM-BASED IMAGE SEARCH

The colour bar chart for an image is formed by count the number of pixels of each colour. Retrieval from image databases practice colour histograms has been investigated in [tools, fully, automated]. In these studies the developments of the extraction algorithms follow a similar progression: (1) selection of a colour space, (2) division of the colour space, (3) computation of histograms, (4) derivation of the bar chart distance perform, (5) identification of assortment shortcuts. each of these steps is additionally crucial towards developing a successful algorithm.

There square measure several difficulties with chart based totally retrieval. the first of these is that the high property of the colour histograms. Even with forceful division of the colour space, the image chart feature spaces can occupy over 100 dimensions in real valued area. This high property ensures that ways of feature reduction, pre-filtering and hierarchical categorization ought to be implemented. the huge property collectively can increase the standard and computation of the house perform. It notably complicates `cross' distance functions that embrace the activity distance between chart bins.

Colour histogram Definition:

An image bar chart refers to the chance mass perform of the image intensities. this may be extended for colour footage to capture the joint possibilities of the intensities of the three colour channels. further formally, the colour chart is printed by,

$$h_{A,B,C}(a,b,c) = N \cdot \mathrm{Pr}ob(A = a, B = b, C = c)$$

where A , B and C represent the three colour channels (R,G,B or H,S,V) and N is that the range of pixels among the image. Computationally, the colour bar chart is made by discrediting the colours at intervals an image and enumeration the number of pixels of each colour.

Since the quality pc represents colour pictures with up to 224 colours, this technique usually desires substantial division of the colour house. the foremost issues regarding the employment of colour histograms for assortment involve the choice of colour house and division of the colour house. Once a perceptually uniform colour home is chosen uniform division is additionally acceptable. If a non-uniform colour home is chosen, then non-uniform division is additionally needed. usually smart issues, prefer to be compatible with the electronic computer show, encourage the alternatives of uniform division and RGB colour house. the colour bar chart are thought of as a gaggle of vectors. For gray-scale footage these unit a pair of dimensional vectors. One dimension provides the price of the gray-level and thus the choice the count of pixels at the gray-level. For colour footage the colour histograms unit composed of 4-D vectors. This makes colour histograms very hard to determine. There are a unit several loss approaches for viewing colour histograms, one altogether the most effective is to seem at severally the histograms of the colour channels. this sort of visual image can illustrate variety of the salient choices of the colour bar chart.

Colour Uniformity:

The RGB colour space is way from being perceptually uniform. to urge associate honest colour illustration of the image by uniformly sampling the RGB house it is necessary to select out the division step sizes to be fine enough such distinct colours do not appear to be assigned to constant bin. The drawback is that oversampling at constant time produces a much bigger set of colours than is additionally needed. the increase among the range of bins among the bar chart impacts performance of knowledge retrieval. large sized histograms become computationally unwieldy, notably once distance functions unit computed for many things among the data. Moreover, as we've got a bent to shall see among subsequent section, to possess finer but not perceptually uniform sampling of colours negatively impacts retrieval effectiveness.

However, the HSV colour house mentioned earlier offers improved activity uniformity. It represents with equal stress the three colour variants that characterize colour: Hue, Saturation and worth (Intensity). This separation is participating as a results of colour image method performed severally on the colour channels does not introduce false colours. moreover, it's easier to complete many artefacts and colour distortions. as associate example, lighting and shading artefacts area unit usually be isolated to the lightness channel. but this colour home is typically inconvenient due to the non-linearity in forward and reverse transformation with RGB house.

There are 3 kinds of pix applied in Digital Image Processing. They are

1. Binary Image
2. Gray Scale Image
3. Color Image

Binary picture

A binary photograph is a virtual photo that has most effective viable values for every pixel. Typically the two colorings used for a binary image are black and white despite the truth that any colors may be used. The color used for the item(s) in the photograph is the foreground color while the rest of the picture is the historic past coloration.

Binary pix are also called bi-stage or -stage. This approach that every pixel is stored as a single bit (zero or 1).This name black and white, monochrome or monochromatic are regularly used for this idea, however may also additionally designate any pics that have best one pattern in keeping with pixel, together with grayscale photos

Binary pics often upward push up in virtual image processing as masks or because the result of positive operations collectively with segmentation, thresholding, and dithering.

Gray scale images

A grayscale Image is virtual photo is an picture in which the price of every pixel is a single sample, this is, it consists of satisfactory depth information. Images of this type, also referred to as black-and-white, are composed solely of sun shades of gray(0-255), various from black(0) on the weakest depth to white(255) at the maximum effective.

Grayscale snap shots are extremely good from one-bit black-and-white pix, which in the context of laptop imaging are pix with only the 2 sun sunglasses, black, and white (additionally known as bi-diploma or binary photos). Grayscale snap shots have many sun sun shades of grey in among. Grayscale photos are also referred to as monochromatic, denoting the absence of any chromatic version.

Grayscale pix are frequently the quit end result of measuring the intensity of moderate at every pixel in a single band of the electromagnetic spectrum (e.G. Infrared, seen mild, ultraviolet, and lots of others.), and in such instances they will be monochromatic proper at the same time as simplest a given frequency is captured. But moreover they may be synthesized from a whole color picture; see the phase approximately changing to grayscale.

Colour picture

A (digital) colour image is a virtual photograph that includes shade records for every pixel. Each pixel has a specific rate which determines its appearing shade. This fee is certified by means of manner of 3 numbers giving the decomposition of the color inside the 3 number one sun shades Red, Green and Blue. Any shade seen to human eye may be represented this manner. The decomposition of a colour inside the 3 number one hues is quantified through some of amongst 0 and 255. For instance, white may be coded as R = 255, G = 255, B = 255; black can be referred to as (R,G,B) = (0,0,0); and say, amazing pink may be : (255,0,255).

In great phrases, an photo is an intensive -dimensional array of colour values, pixels, each of them coded on three bytes, representing the 3 primary shades. This allows the image to encompass a complete of 256x256x256 = 16.Eight million particular shades. This approach is likewise referred to as RGB encoding, and is particularly tailor-made to human imaginative and prescient

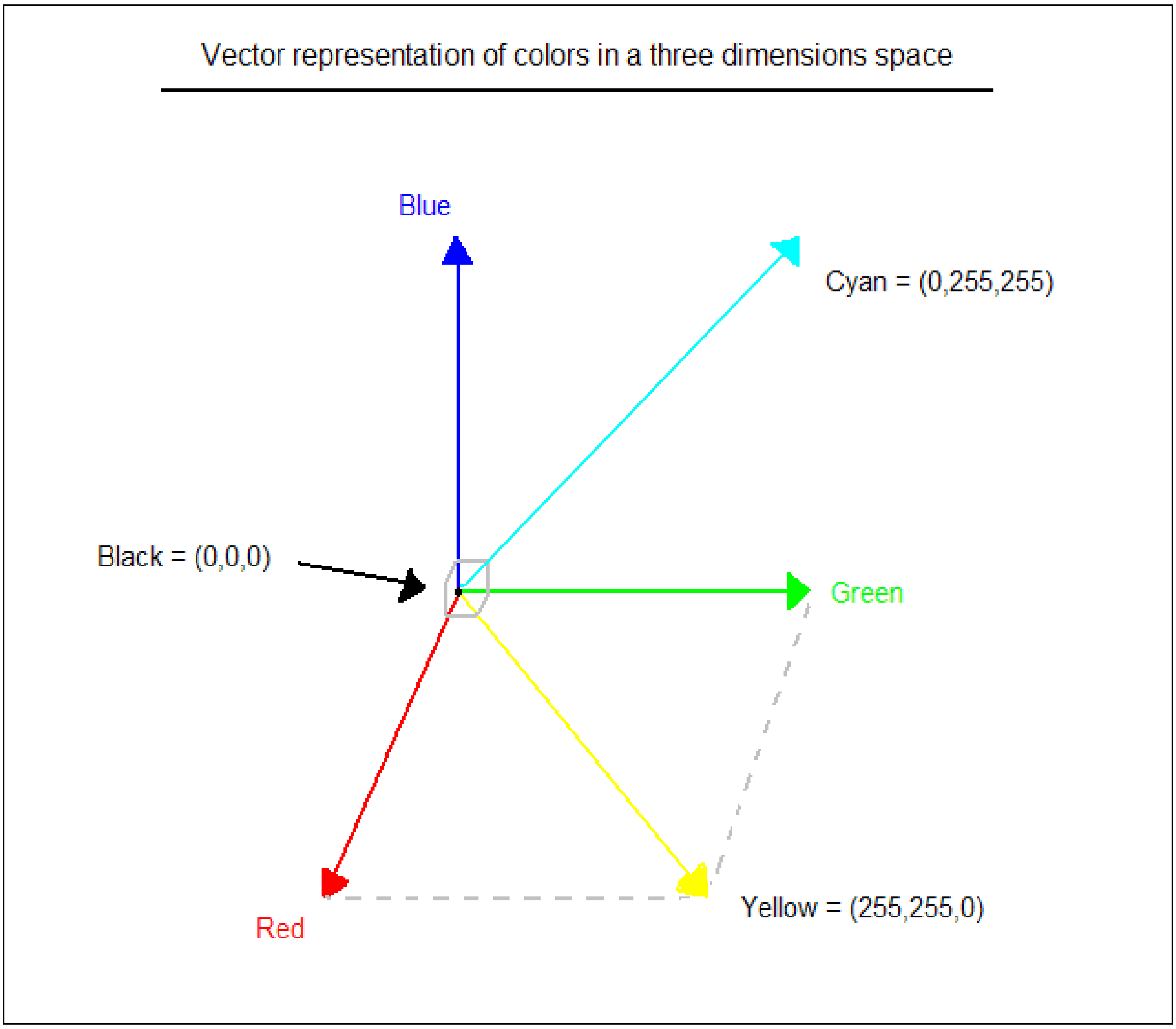


Fig 5.4 Hue Saturation Process of RGB SCALE Image

From the above determine, sun shades are coded on 3 bytes representing their decomposition on the 3 primary colorings. It sounds obvious to a mathematician to right away interpret colors as vectors in a three size area in which every axis stands for one of the number one hues. Therefore we are able to gain of maximum of the geometric mathematical requirements to cope with our colors, which consist of norms, scalar product, projection, rotation or distance.

The identification of devices in an picture and this method may additionally possibly begin with photograph processing techniques collectively with noise removal, observed via the usage of (low-diploma) feature extraction to find out lines, areas and probable regions with sure textures.

The smart bit is to interpret collections of those shapes as single devices, e.G. Vehicles on a avenue, boxes on a conveyor belt or cancerous cells on a microscope slide. One motive this is an AI problem is that an item can seem very one-of-a-type while viewed from unique angles or underneath amazing lighting. Another problem is finding out what skills belong to what item and which can be information or shadows and so on. The human visible device plays these responsibilities specifically unconsciously however a laptop calls for skilful programming and masses of processing strength to method human commonplace overall performance. Manipulation of records inside the shape of an photograph through severa viable strategies. An image is generally interpreted as a -dimensional array of brightness values, and is maximum familiarly represented via way of such patterns as the ones of a photographic print, slide, tv display, or movie show display. An image may be processed optically or digitally with a laptop.

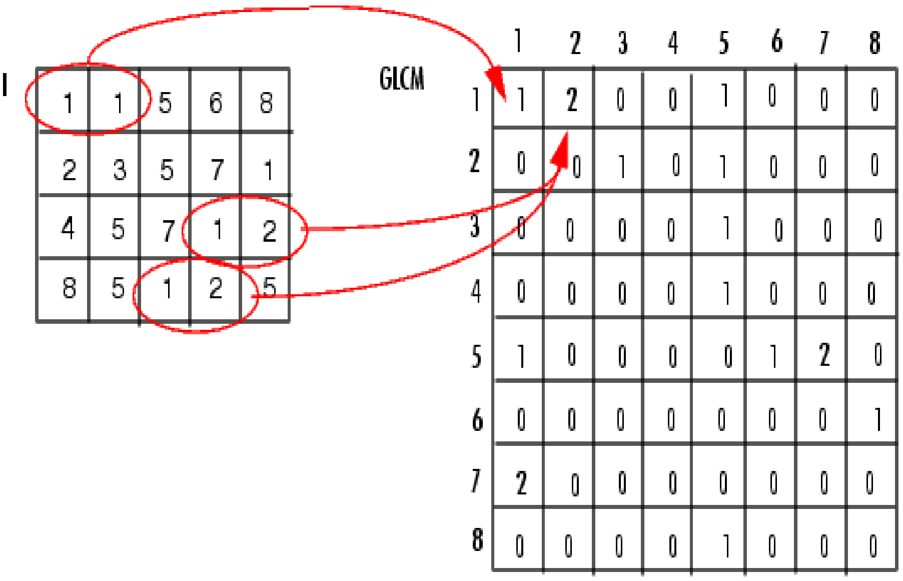
Feature Extraction

Gray-Level Co-Occurrence Matrix:

To create a GLCM, use the graycomatrix function. The graycomatrix function creates a gray-level co-occurrence matrix (GLCM) by calculating how often a pixel with the intensity (gray-level) value *i* occurs in a specific spatial relationship to a pixel with the value *j*. By default, the spatial relationship is defined as the pixel of interest and the pixel to its immediate right (horizontally adjacent), but you can specify other spatial relationships between the two pixels. Each element (*i*,*j*) in the resultant GLCM is simply the sum of the number of times that the pixel with value *i* occurred in the specified spatial relationship to a pixel with value *j* in the input image. Because the processing required to calculate a GLCM for the full dynamic range of an image is prohibitive, graycomatrix scales the input image. By default, graycomatrix uses scaling to reduce the number of intensity values in gray scale image from 256 to eight. The number of gray levels determines the size of the GLCM. To control the number of gray levels in the GLCM and the scaling of intensity values, using the Num Levels and the Gray Limits parameters of the graycomatrix function. See the graycomatrix reference page for more information.

The gray-level co-occurrence matrix can reveal certain properties about the spatial distribution of the gray levels in the texture image. For example, if most of the entries in the GLCM are concentrated along the diagonal, the texture is coarse with respect to the specified offset. To illustrate, the following figure shows how graycomatrix calculates the first three values in a GLCM. In the output GLCM, element (1,1) contains the value 1 because there is only one instance in the input image where two horizontally adjacent pixels have the values 1 and 1, respectively.

GLCM(1,2)contains the value 2 because there are two instances where two horizontally adjacent pixels have the values 1 and 2. Element (1,3) in the GLCM has the value 0 because there are no instances of two horizontally adjacent pixels with the values 1 and 3. graycomatrix continues processing the input image, scanning the image for other pixel pairs (*i*,*j*) and recording the sums in the corresponding elements of the GLCM.



I

1	1	5	6	8
2	3	5	7	1
4	5	7	1	2
8	5	1	2	5

GLCM

	1	2	3	4	5	6	7	8
1	1	2	0	0	1	0	0	0
2	0	0	1	0	1	0	0	0
3	0	0	0	0	1	0	0	0
4	0	0	0	0	1	0	0	0
5	1	0	0	0	0	1	2	0
6	0	0	0	0	0	0	0	1
7	2	0	0	0	0	0	0	0
8	0	0	0	0	1	0	0	0

To create multiple GLCMs, specify an array of offsets to the graycomatrix function. These offsets define pixel relationships of varying direction and distance. For example, you can define an array of offsets that specify four directions (horizontal, vertical, and two diagonals) and four distances. In this case, the input image is represented by 16 GLCMs. When you calculate statistics from these GLCMs, you can take the average.

You specify these offsets as a *p*-by-2 array of integers. Each row in the array is a two-element vector, [row_offset, col_offset], that specifies one offset. Row_offset is the number of rows between the pixel of interest and its neighbour. Col_offset is the number of columns between the pixel of interest and its neighbour. This example creates an offset that specifies four directions and 4 distances for each direction. After you create the GLCMs, you can derive several statistics from them using the graycoprops function. These statistics provide information about the texture of an image.Statistic such a as Contras, Correlation, Energy, Homogeneity gives information about image.

ARTICIAL NEURAL NETWORK

Artificial neural networks (**ANNs**), usually simply called **neural networks** (**NNs**), are computing systems vaguely inspired by the biological neural networks that constitute animal brains.

An ANN is based on a collection of connected units or nodes called artificial neurons, which loosely model the neurons in a biological brain. Each connection, like the synapses in a biological brain, can transmit a signal to other neurons. An artificial neuron that receives a signal then processes it and can signal neurons connected to it. The "signal" at a connection is a real number, and the output of each neuron is computed by some non-linear function of the sum of its inputs. The connections are called *edges*. Neurons and edges typically have a *weight* that adjusts as learning proceeds. The weight increases or decreases the strength of the signal at a connection. Neurons may have a threshold such that a signal is sent only if the aggregate signal crosses that threshold. Typically, neurons are aggregated into layers. Different layers may perform different transformations on their inputs. Signals travel from the first layer (the input layer), to the last layer (the output layer), possibly after traversing the layers multiple times.

Training

Neural networks learn (or are trained) by processing examples, each of which contains a known "input" and "result," forming probability-weighted associations between the two, which are stored within the data structure of the net itself. The training of a neural network from a given example is usually conducted by determining the difference between the processed output of the network (often a prediction) and a target output. This difference is the error. The network then adjusts its weighted associations according to a learning rule and using this error value. Successive adjustments will cause the neural network to produce output which is increasingly similar to the target output. After a sufficient number of these adjustments the training can be terminated based upon certain criteria. This is known as supervised learning.

Such systems "learn" to perform tasks by considering examples, generally without being programmed with task-specific rules. For example, in image recognition, they might learn to identify images that contain cats by analyzing example images that have been manually labeled as "cat" or "no cat" and using the results to identify cats in other images. They do this without any prior knowledge of cats, for example, that they have fur, tails, whiskers, and cat-like faces. Instead, they automatically generate identifying characteristics from the examples that they process.

History[

Warren McCulloch and Walter Pitts[1] (1943) opened the subject by creating a computational model for neural networks.[2] In the late 1940s, D. O. Hebb[3] created a learning hypothesis based on the mechanism of neural plasticity that became known as Hebbian learning. Farley and Wesley A. Clark[4] (1954) first used computational machines, then called "calculators", to simulate a Hebbian network. Rosenblatt[5] (1958) created the perceptron.[6] The first functional networks with many layers were published by Ivakhnenko and Lapa in 1965, as the Group Method of Data Handling.[7][8][9] The basics of continuous backpropagation[7][10][11][12] were derived in the context of control theory by Kelley[13] in 1960 and by Bryson in 1961,[14] using principles of dynamic programming. Thereafter research stagnated following Minsky and Papert (1969),[15] who discovered that basic perceptrons were incapable of processing the exclusive-or circuit and that computers lacked sufficient power to process useful neural networks.

In 1970, Seppo Linnainmaa published the general method for automatic differentiation (AD) of discrete connected networks of nested differentiable functions.[16][17] In 1973, Dreyfus used backpropagation to adapt parameters of controllers in proportion to error gradients.[18] Werbos's (1975) backpropagation algorithm enabled practical training of multi-layer networks. In 1982, he applied Linnainmaa's AD method to neural networks in the way that became widely used.[10][19]

The development of metal–oxide–semiconductor (MOS) very-large-scale integration (VLSI), in the form of complementary MOS (CMOS) technology, enabled increasing MOS transistor counts in digital electronics. This provided more processing power for the development of practical artificial neural networks in the 1980s.[20]

In 1986 Rumelhart, Hinton and Williams showed that backpropagation learned interesting internal representations of words as feature vectors when trained to predict the next word in a sequence.[21]

In 1992, max-pooling was introduced to help with least-shift invariance and tolerance to deformation to aid 3D object recognition.[22][23][24] Schmidhuber adopted a multi-level hierarchy of networks (1992) pre-trained one level at a time by unsupervised learning and fine-tuned by backpropagation.[25]

Geoffrey Hinton et al. (2006) proposed learning a high-level representation using successive layers of binary or real-valued latent variables with a restricted Boltzmann machine[26] to model each layer. In 2012, Ng and Dean created a network that learned to recognize higher-level concepts, such as cats, only from watching unlabeled images.[27] Unsupervised pre-training and increased computing power from GPUs and distributed computing allowed the use of larger networks, particularly in image and visual recognition problems, which became known as "deep learning".[28]

Ciresan and colleagues (2010)[29] showed that despite the vanishing gradient problem, GPUs make backpropagation feasible for many-layered feedforward neural networks.[30] Between 2009 and 2012, ANNs began winning prizes in ANN contests, approaching human level performance on various tasks, initially in pattern recognition and machine learning.[31][32] For example, the bi-directional and multi-dimensional long short-term memory (LSTM)[33][34][35][36] of Graves et al. won three competitions in connected handwriting recognition in 2009 without any prior knowledge about the three languages to be learned.[35][34]

Ciresan and colleagues built the first pattern recognizers to achieve human-competitive/superhuman performance[37] on benchmarks such as traffic sign recognition (IJCNN 2012).

Components of ANN

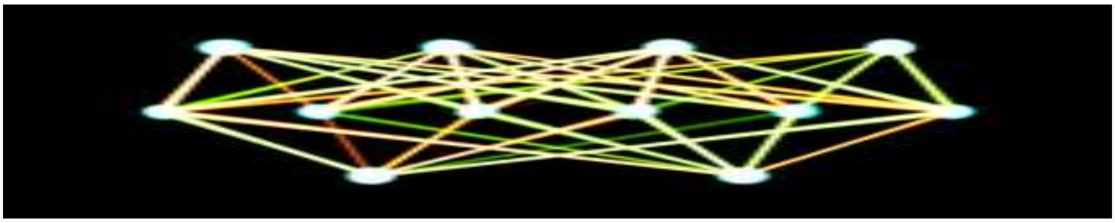
Neurons

ANNs are composed of artificial neurons which are conceptually derived from biological neurons. Each artificial neuron has inputs and produces a single output which can be sent to multiple other neurons. The inputs can be the feature values of a sample of external data, such as images or documents, or they can be the outputs of other neurons. The outputs of the final *output neurons* of the neural net accomplish the task, such as recognizing an object in an image.

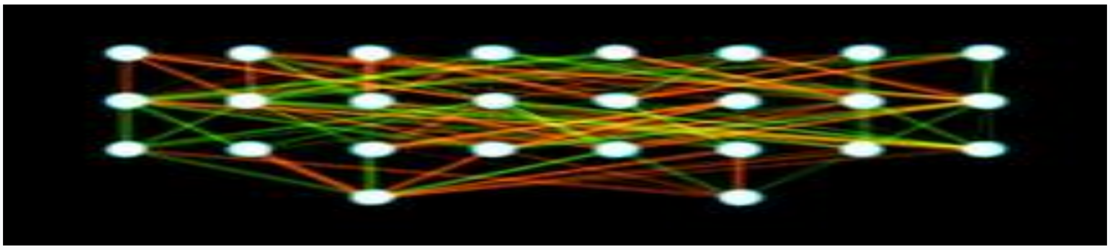
To find the output of the neuron, first we take the weighted sum of all the inputs, weighted by the *weights* of the *connections* from the inputs to the neuron. We add a *bias* term to this sum. This weighted sum is sometimes called the *activation*. This weighted sum is then passed through a (usually nonlinear) activation function to produce the output. The initial inputs are external data, such as images and documents. The ultimate outputs accomplish the task, such as recognizing an object in an image.[40]

Connections and weights

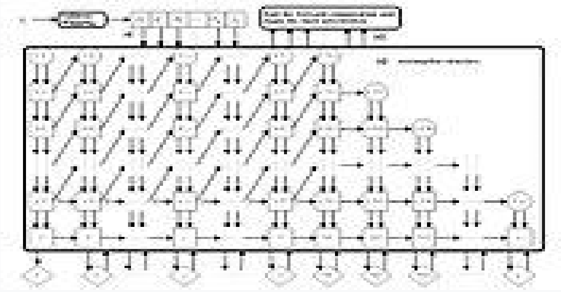
The network consists of connections, each connection providing the output of one neuron as an input to another neuron. Each connection is assigned a weight that represents its relative importance.[38] A given neuron can have multiple input and output connections.

- 

  A single-layer feedforward artificial neural network with 4 inputs, 6 hidden and 2 outputs. Given position state and direction outputs wheel based control values.

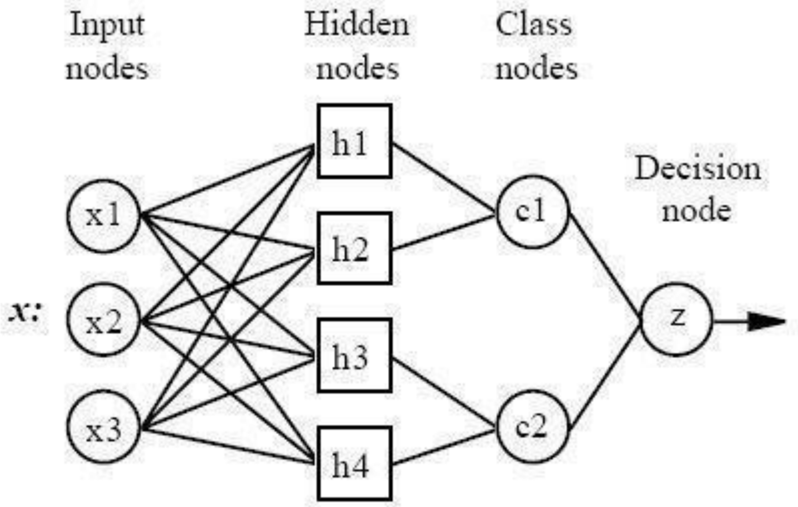
- 

A two-layer feedforward artificial neural network with 8 inputs, 2x8 hidden and 2 outputs. Given position state, direction and other environment values outputs thruster based control values.

- 

Parallel pipeline structure of CMAC neural network. This learning algorithm can converge in one step.

Architecture of a NN:



All NN networks have four layers:

1. **Input layer** — There is one neuron in the input layer for each predictor variable. In the case of categorical variables, *N*-1 neurons are used where *N* is the number of categories. The input neurons (or processing before the input layer) standardizes the range of the

values by subtracting the median and dividing by the interquartile range. The input neurons then feed the values to each of the neurons in the hidden layer.

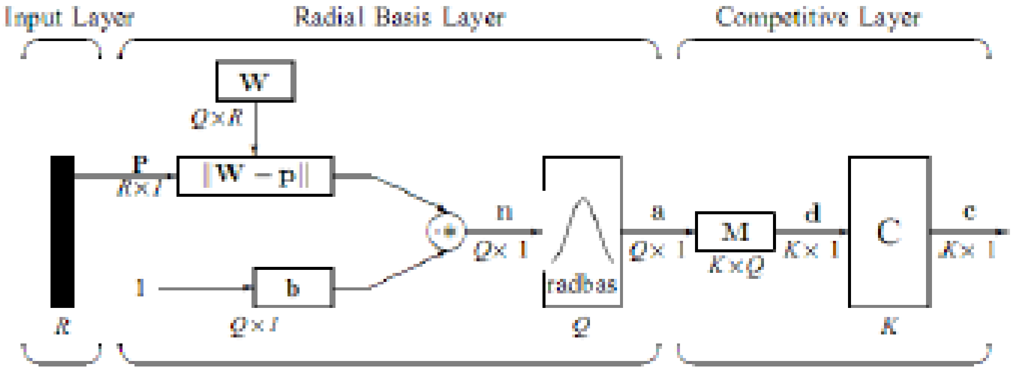
1. **Hidden layer** — This layer has one neuron for each case in the training data set. The neuron stores the values of the predictor variables for the case along with the target value. When presented with the *x* vector of input values from the input layer, a hidden neuron computes the Euclidean distance of the test case from the neuron’s center point and then applies the RBF kernel function using the sigma value(s). The resulting value is passed to the neurons in the pattern layer.

1. **Pattern layer / Summation layer** — The next layer in the network is different for NN networks and for GRNN networks. For NN networks there is one pattern neuron for each category of the target variable. The actual target category of each training case is stored with each hidden neuron; the weighted value coming out of a hidden neuron is fed only to the pattern neuron that corresponds to the hidden neuron’s category. The pattern neurons add the values for the class they represent (hence, it is a weighted vote for that category). For GRNN networks, there are only two neurons in the pattern layer. One neuron is the denominator summation unit the other is the numerator summation unit. The denominator summation unit adds up the weight values coming from each of the hidden neurons. The numerator summation unit adds up the weight values multiplied by the actual target value for each hidden neuron.

1. **Decision layer** — The decision layer is different for NN and GRNN networks. For NN networks, the decision layer compares the weighted votes for each target category accumulated in the pattern layer and uses the largest vote to predict the target category.

   For GRNN networks, the decision layer divides the value accumulated in the numerator summation unit by the value in the denominator summation unit and uses the result as the predicted target value.

The following diagram is actual diagram or propose network used in our project



1) Input Layer:

The input vector, denoted as **p**, is presented as the black vertical bar. Its dimension is *R* × *1*. In this paper, *R* = *3*.

2) Radial Basis Layer:

In Radial Basis Layer, the vector distances between input vector p and the weight vector made of each row of weight matrix **W** are calculated. Here, the vector distance is defined as the dot product between two vectors [8]. Assume the dimension of **W** is *Q*×*R*. The dot product between **p** and the *i*-th row of **W** produces the *i*-th element of the distance vector ||**W-p**||, whose dimension is *Q*×*1*. The minus symbol, "-", indicates that it is the distance between vectors. Then, the bias vector **b** is combined with ||**W- p**|| by an element-by-element multiplication, .The result is denoted as **n** = ||**W- p**|| **..p**. The transfer function in NN has built into a distance criterion with respect to a center. In this paper, it is defined as *radbas(n)* = 2 *n* *e*- (1) Each element of **n** is substituted into Eq. 1 and produces corresponding element of **a**, the output vector of Radial Basis Layer. The *i*-th element of **a** can be represented as ai = *radbas*(||**W**i - **p**|| **..b**i) (2) where **W**i is the vector made of the *i*-th row of **W** and **b**i is the *i*-th element of bias vector **b**.

3.ANALYSIS

3.1 Software Requirement Specification

- **Support Vector Machine**
- **K-Mean Clustering**
- **RANDOM Forest Classifier**

Drawbacks

- Not getting proper frame alignments.
- It can't give 99% accuracy.

SVM

SVM is a group of learning algorithms primarily used for classification tasks on complicated data such as image classification and protein structure analysis. SVM is used in a countless fields in science and industry, including Bio-technology, Medicine, Chemistry and Computer Science. It has also turned out to be ideally suited for categorization of large text repositories such as those housed in virtually all large, modern organizations. Introduced in 1992, SVM quickly became regarded as the state-of-the-art method for classification of complex, high-dimensional data. In particular its ability to capture trends observed in a small training set and to generalize those trends against a broader corpus have made it useful across a large number of applications. SVM uses a supervised learning approach, which means it learns to classify unseen data based on a set of labeled training data, s uch as corporate documents. The initial set of training data is typically identified by domain experts and is used to build a model that can be applied to any other data outside the training set. The effort required to construct a high quality training set is quite modest, particularly when compared to the volume of data that may be ultimately classified against it. This means that learning algorithms such as SVM offer an exceptionally cost effective method of text classification for the massive volumes of documents produced by modern organizations. The balance of this paper covers the inner workings of SVM, its application in science and industry, the legal defensibility of the method as well as classification accuracy compared to manual classification.

Overview

SVM is built upon a solid foundation of statistical learning theory. Early classifiers were proposed by Vladimir Vapnik and Alexey Chervonenkis more than 40 years ago. In 1992 Boser, Guyon and Vapnik proposed an improvement that considerably extended the applicability of SVM. From this point on SVM began to establish its reputation as the stateof-the-art method for data categorization. Starting with handwriting recognition tasks SVM showed results that were superior to all other methods of classification. It was quickly shown that SVM was able to beat even Artificial Neural Networks that were considered to be the strongest categorization algorithms at the time. Thousands of researchers applied SVM to a large number of machine learning problems and the results have contributed to the acceptance of this technology as the state-of-the-art for machine classification. Numerous studies (Joachmis 1998, Dumais et al. 1998, Drucker et al. 1999) have shown the superiority of SVM over other machine learning methods for text categorization problems. For example, Joachmis reported 86% accuracy of SVM on classification of the Reuters news dataset, while the next best method, a significantly slower k-NearestNeighbor algorithm was only able to achieve 82% accuracy. Today SVM is widely accepted in industry as well as in the academia. For example, Health Discovery Corporation uses SVM in a medical image analysis tool currently licensed to Pfizer, Dow Chemical uses SVM in their research for outlier detection and Reuters uses SVM for text classification.

Introduction to K-Means Clustering

K-means clustering is a type of unsupervised learning, which is used when you have unlabeled data (i.e., data without defined categories or groups). The goal of this algorithm is to find groups in the data, with the number of groups represented by the variable *K*. The algorithm works iteratively to assign each data point to one of *K* groups based on the features that are provided. Data points are clustered based on feature similarity. The results of the *K*-means clustering algorithm are:

1. The centroids of the *K* clusters, which can be used to label new data

2. Labels for the training data (each data point is assigned to a single cluster)

Rather than defining groups before looking at the data, clustering allows you to find and analyze the groups that have formed organically. The "Choosing K" section below describes how the number of groups can be determined.

Each centroid of a cluster is a collection of feature values which define the resulting groups. Examining the centroid feature weights can be used to qualitatively interpret what kind of group each cluster represents.

This introduction to the *K*-means clustering algorithm covers:

- Common business cases where K-means is used
- The steps involved in running the algorithm
- A Python example using delivery fleet data

K-Mean Algorithm

The *K*-means clustering algorithm uses iterative refinement to produce a final result. The algorithm inputs are the number of clusters *K* and the data set. The data set is a collection of features for each data point. The algorithms starts with initial estimates for the *K* centroids, which can either be randomly generated or randomly selected from the data set. The algorithm then iterates between two steps:

1. Data Assignment step: Each centroid defines one of the clusters. In this step, each data point is assigned to its nearest centroid, based on the squared Euclidean distance. More formally, if c_i is the collection of centroids in set C, then each data point x is assigned to a cluster based on where $dist(\cdot)$ is the standard (L_2) Euclidean distance. Let the set of data point assignments for each i^{th} cluster centroid be S_i.

2. Centroid update step: In this step, the centroids are recomputed. This is done by taking the mean of all data points assigned to that centroid's cluster.

The algorithm iterates between steps one and two until a stopping criteria is met (i.e., no data points change clusters, the sum of the distances is minimized, or some maximum number of iterations is reached).This algorithm is guaranteed to converge to a result. The result may be a local optimum (i.e. not necessarily the best possible outcome), meaning that assessing more than one run of the algorithm with randomized starting centroids may give a better outcome.

Proposed System

- PREPROCESSING
- RGB COLOR MODEL
- GLCM FEATURE EXTRACTION
- BLOB DETECTION
- Artificial NEURAL NETWORK

Advantages :

- Maximum accuracy in classification
- Real time achievement
- Machine based prediction

Hardware Requirements

- MONITOR
- HDD: 1TB
- RAM: 8GB

Software Requirements

- PYTHON 3.9 VERSION
- Anaconda Navigator
- Python IDLE Editor

PYTHON

Python is an interpreted high-level programming language for programming Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter outputs the fastest and easiest way to create the GUI applications. Creating a GUI using tkinter is an easy task.

PYTHON FEATURES:

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented , imperative, functional and procedural, and has a large and comprehensive library. Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect -oriented programming (including by meta programming and meta objects (magic methods)). Many other paradigms are supported via extensions, including design by contract and logic programming.

PYTHON LIBRARIES

Python's large standard library, commonly cited as one of its greatest strengths, provides tools suited too many tasks. For Internet-facing applications,many standard formats and protocols such as MIME and HTTP are supported. It includes modules for creating graphical user interfaces, connecting to relational databases, generating pseudorandom numbers, arithmetic with arbitrary precision decimals, manipulating regular expressions, and unit testing.

As of March 2018, the Python Package Index (PyPI), the official repository for third-party Python software, contains over 130,000 packages with a wide range of functionality, including:

- Graphical user interfaces
- Web frameworks
- Multimedia
- Databases
- Networking
- Test frameworks
- Automation
- Web scraping
- Documentation
- System administration
- Scientific computing
- Text processing
- Image processing

DEPLOYMENT ENVIRONMENT

Most Python implementations (including CPython) include a read–eval–print loop (REPL), permitting them to function as a command line interpreter for which the user enters statements sequentially and receives results immediately. Other shells, including IDLE and IPython, add further abilities such as auto-completion, session state retention and syntax highlighting.

As well as standard desktop integrated development environments, there are Web browser-based IDEs; Sage Math (intended for developing science and math-related Python programs); Python Anywhere, a browser-based IDE and hosting environment; and Canopy IDE, a commercial Python IDE emphasizing scientific computing.

OPENCV-PYTHON

Python is a general purpose programming language started by Guido van Rossum, which became very popular in short time mainly because of its simplicity and code readability.

It enables the programmer to express his ideas in fewer lines of code without reducing any readability. Compared to other languages like C/C++, Python is slower. But another important feature of Python is that it can be easily extended with C/C++. This feature helps us to write computationally intensive codes in C/C++ and create a Python wrapper for it so that we can use these wrappers as Python modules. This gives us two advantages: first, our code is as fast as original C/C++ code (since it is the actual C++ code working in background) and second, it is very easy to code in Python. This is how OpenCV-Python works, it is a Python wrapper around original C++ implementation. And the support of Numpy makes the task more easier. Numpy is a highly optimized library for numerical operations. It gives a MATLAB-style syntax. All the OpenCV array structures are converted to-and-from Numpy arrays. So whatever operations you can do in Numpy, you can combine it with OpenCV, which increases number of weapons in your arsenal. Besides that, several other libraries like SciPy, Matplotlib which supports Numpy can be used with this. So OpenCV-Python is an appropriate tool for fast prototyping of computer vision problems.

Since OpenCV is an open source initiative, all are welcome to make contributions to this library. And it is same for this tutorial also. So, if you find any mistake in this tutorial (whether it be a small spelling mistake or a big error in code or concepts, whatever), feel free to correct it

And that will be a good task for freshers who begin to contribute to open source projects. Just fork the OpenCV in github, make necessary corrections and send a pull request to OpenCV. OpenCV developers will check your pull request, give you important feedback and once it passes the approval of the reviewer, it will be merged to OpenCV. Then you become a open source contributor. Similar is the case with other tutorials, documentation etc. As new modules are added to OpenCV-Python, this tutorial will have to be expanded. So those who knows about particular algorithm can write up a tutorial which includes a basic theory of the algorithm and a code showing basic usage of the algorithm and submit it to OpenCV. Remember, we together can make this project a great success !!!

ARRAY ATTRIBUTES

Array attributes reflect information that is intrinsic to the array itself. Generally, accessing an array through its attributes allows you to get and sometimes set intrinsic properties of the array without creating a new array. The exposed attributes are the core parts of an array and only some of them can be reset meaningfully without creating a new array.

For a 1-D array, this has no effect. (To change between column and row vectors, first cast the 1-D array into a matrix object.) For a 2-D array, this is the usual matrix transpose. For an n-D array, if axes are given, their order indicates how the axes are permuted (see Examples). If axes are not provided and a.shape = (i[0], i[1], ... i[n-2], i[n-1]), then a.transpose().shape = (i[n-1], i[n-2], ... i[1], i[0]).

SCALARS

Python defines only one type of a particular data class (there is only one integer type, one floating-point type, etc.). This can be convenient in applications that don’t need to be concerned with all the ways data can be represented in a computer. For scientific computing, however, more control is often needed. In NumPy, there are 24 new fundamental Python types to describe different types of scalars. These type descriptors are mostly based on the types available in the C language that CPython is written in, with several additional types compatible with Python’s types.

METHODS

Array scalars have exactly the same methods as arrays. The default behavior of these methods is to internally convert the scalar to an equivalent 0-dimensional array and to call the corresponding array method. In addition, math operations on array scalars are defined so that the same hardware flags are set and used to interpret the results as for ufunc, so that the error state used for ufuncs also carries over to the math on array scalars.

DATA TYPE OBJECTS (DTYPE)

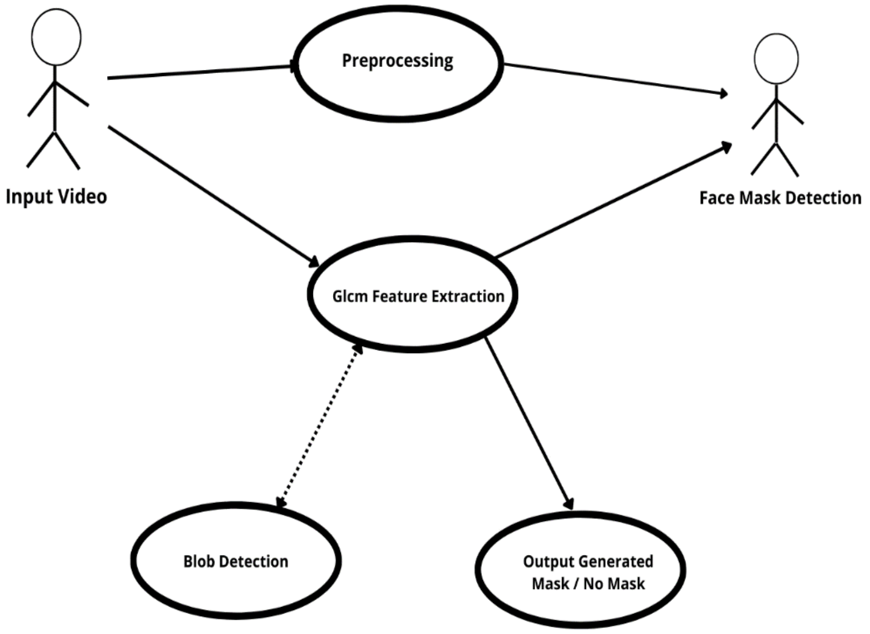
A data type object (an instance of numpy.dtype class) describes how the bytes in the fixed-size block of memory corresponding to an array item should be interpreted. It describes the following aspects of the data:

1. Type of the data (integer, float, Python object, etc.)

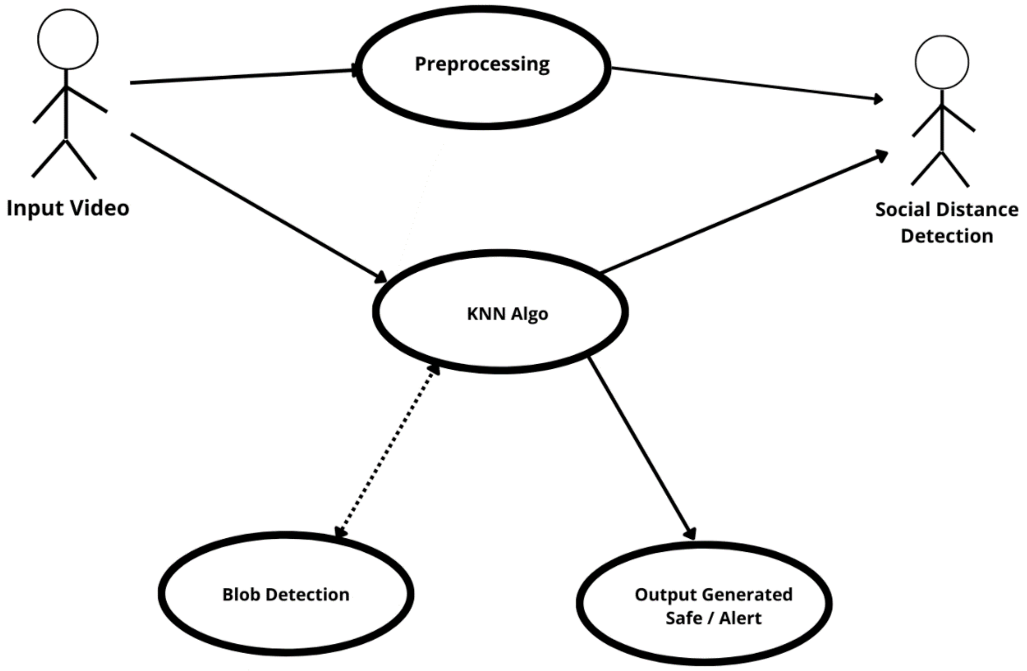
2. Size of the data (how many bytes is in e.g. the integer) 3. Byte order of the data (little-endian or big-endian) 4. If the data type is structured, an aggregate of other data types, (e.g., describing an array item consisting of an integer and a float), (a) what are the names of the “fields” of the structure, by which they can be accessed, (b) what is the data-type of each field, and (c) which part of the memory block each field takes. 5. If the data type is a sub-array, what is its shape and data type. To describe the type of scalar data, there are several built-in scalar types in Numpy for various precision of integers, floating-point numbers, etc. An item extracted from an array, e.g., by indexing, will be a Python object whose type is the scalar type associated with the data type of the array. Note that the scalar types are not dtype objects, even though they can be used in place of one whenever a data type specification is needed in Numpy. Structured data types are formed by creating a data type whose fields contain other data types. Each field has a name by which it can be accessed. The parent data type should be of sufficient size to contain all its fields; the parent is nearly always based on the void type which allows an arbitrary item size. Structured data types may also contain nested structured sub-array data types in their fields. Finally, a data type can describe items that are themselves arrays of items of another data type. These sub-arrays must, however, be of a fixed size. If an array is created using a data-type describing a sub-array, the dimensions of the sub-array are appended to the shape of the array when the array is created. Sub-arrays in a field of a structured type behave differently, see Field Access. Sub-arrays always have a C-contiguous memory layout.

3.2 Use Case Diagram

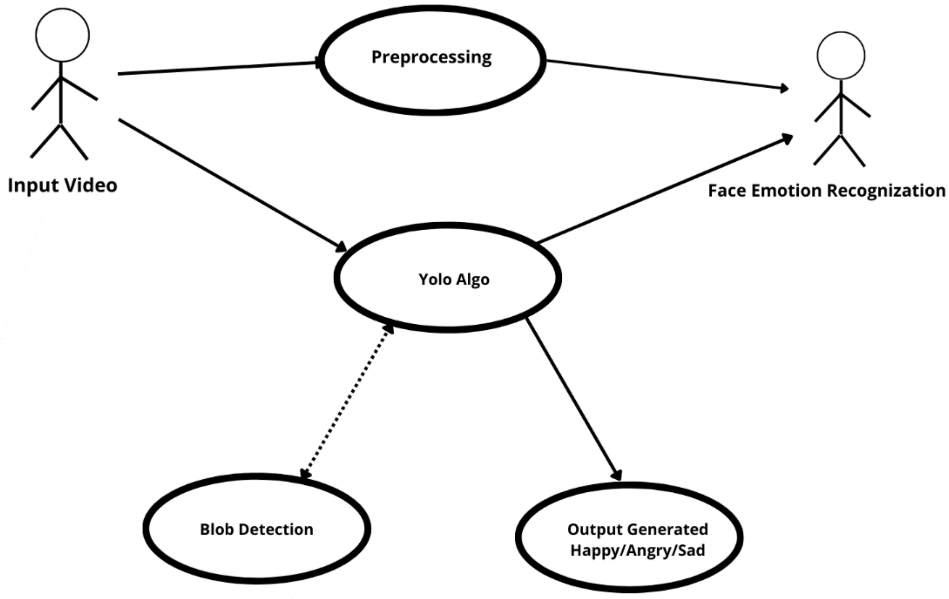
Face Mask Detection



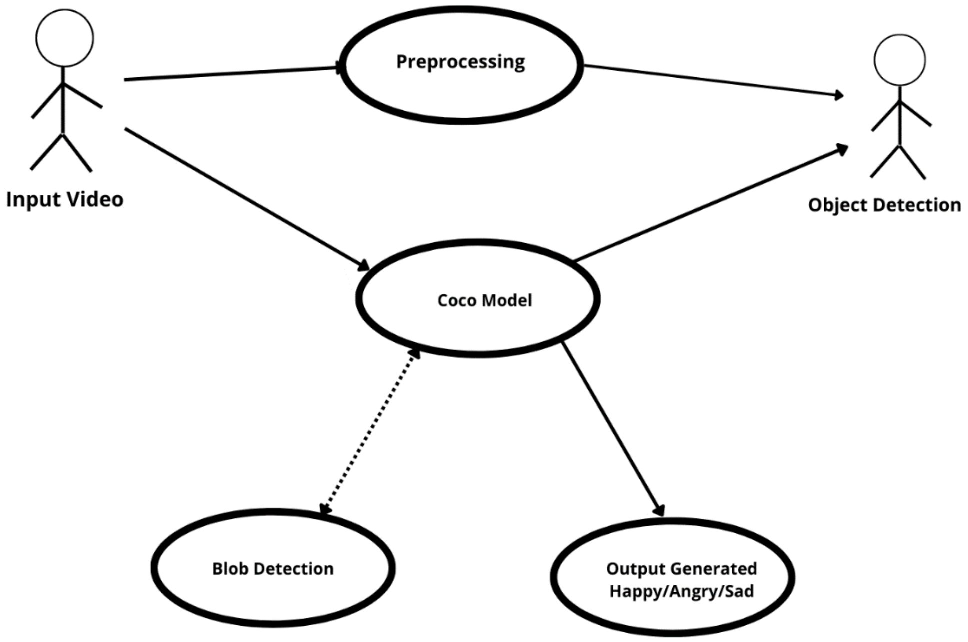
Social Distance Detection:



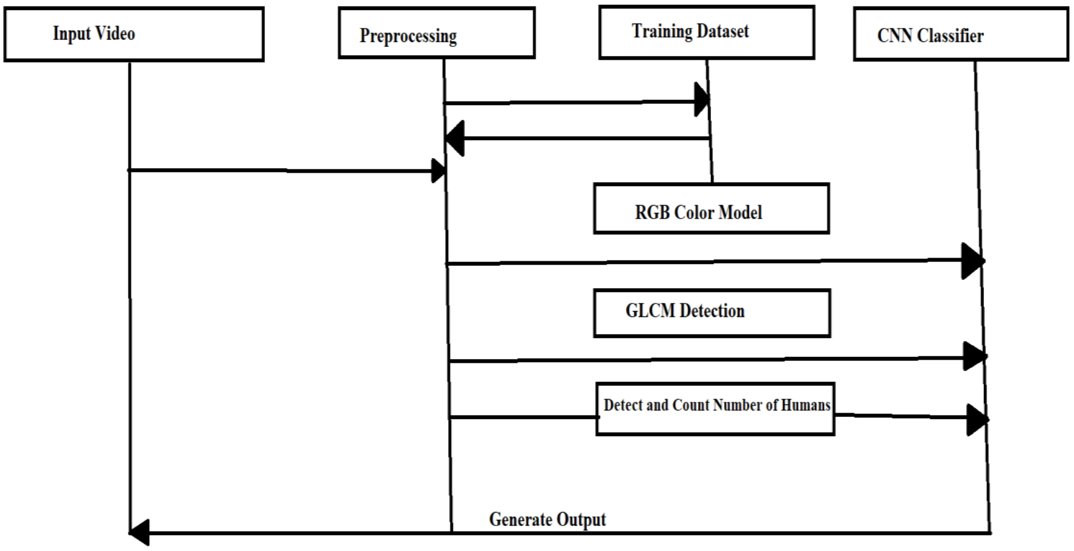
Face Emotion Recognition



Object Detection:



3.3 Sequence Diagram:

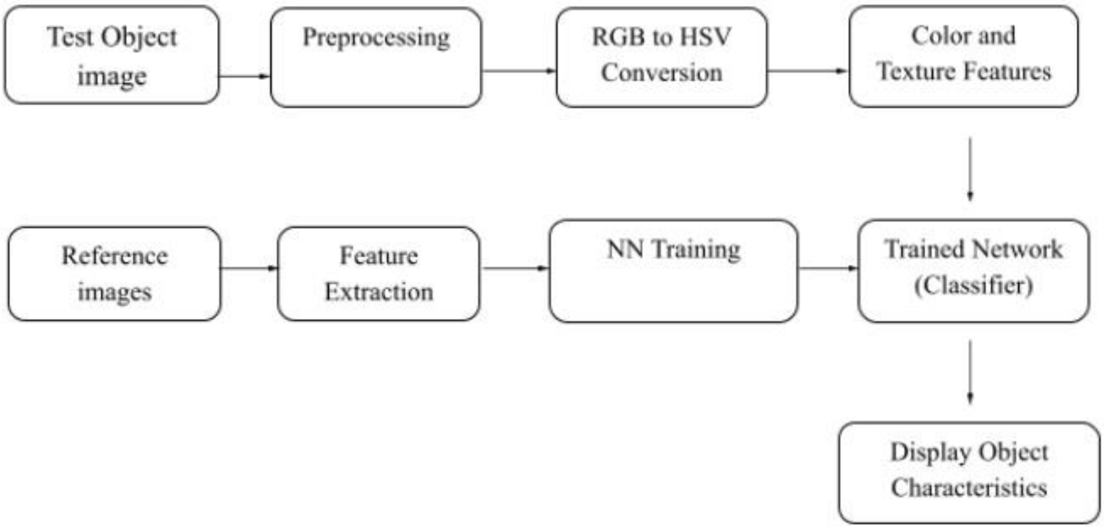


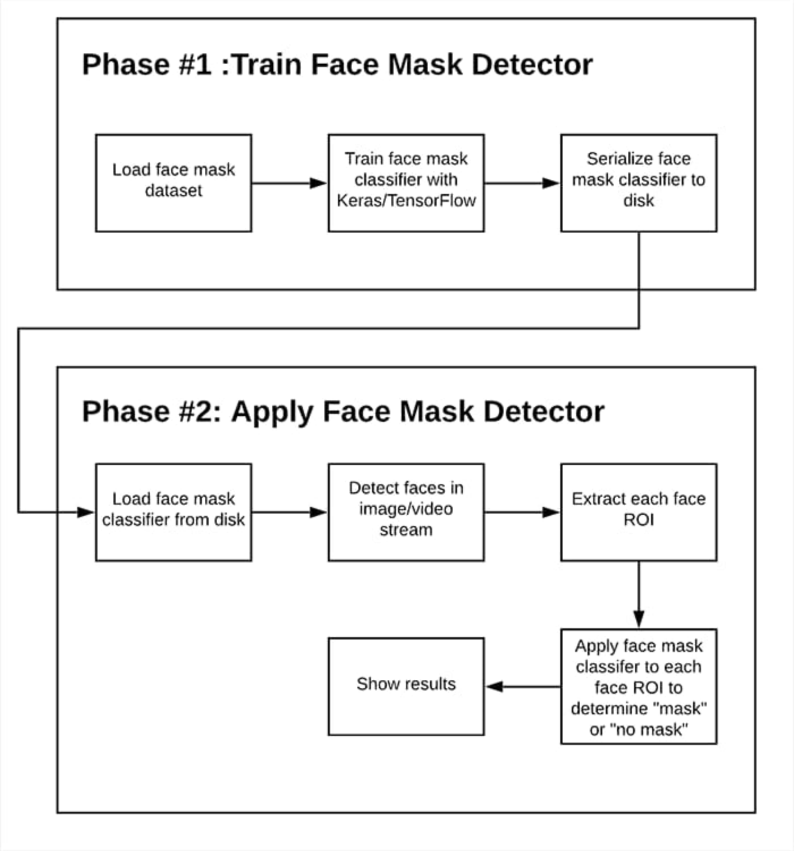
Sequence Diagram:

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

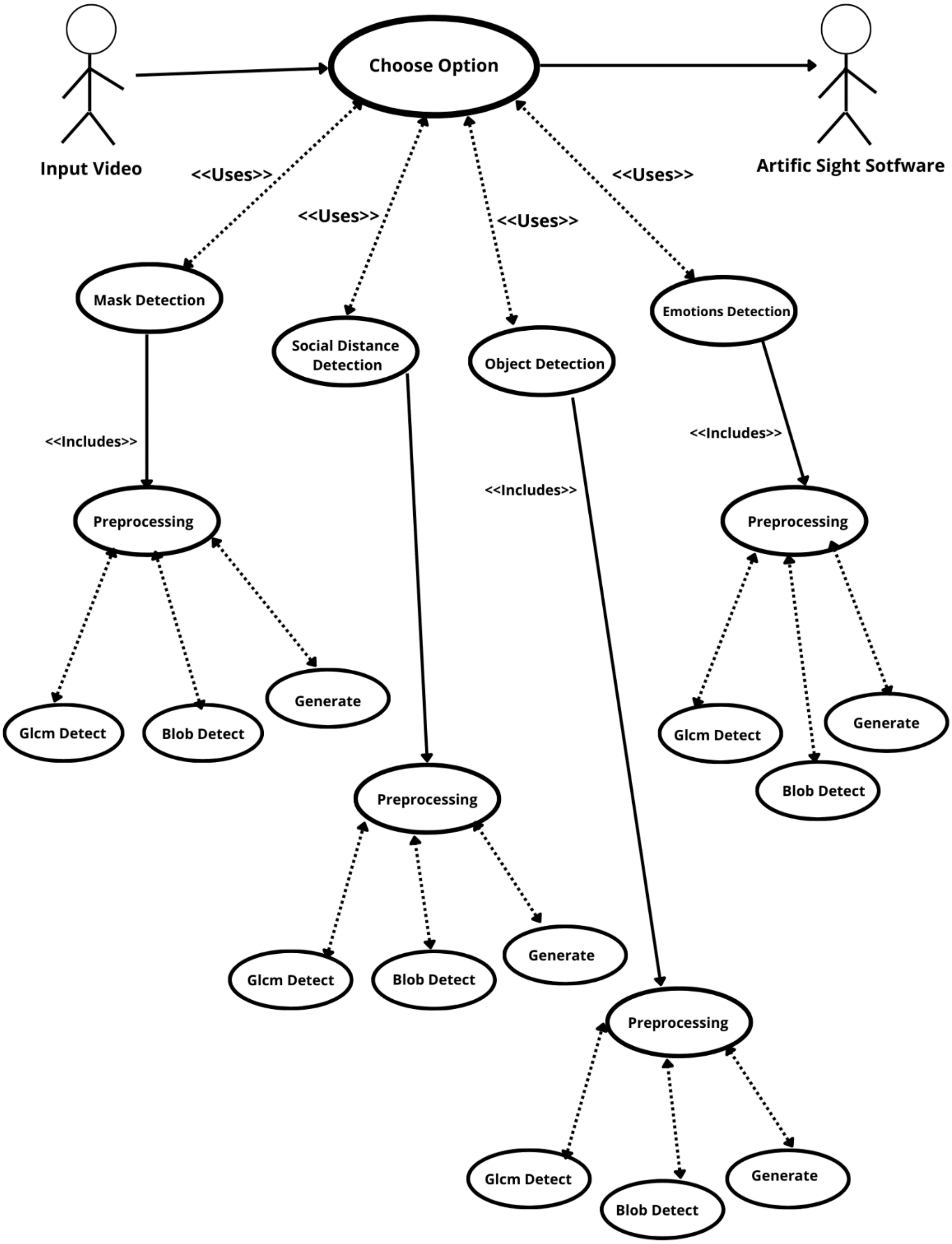
4.DESIGN

4.1 Class Diagram




Phase #1 :Train Face Mask Detector
Load face mask dataset
Train face mask classifier with Keras/TensorFlow
Serialize face mask classifier to disk
Phase #2: Apply Face Mask Detector
Load face mask classifier from disk
Detect faces in image/video stream
Extract each face ROI
Apply face mask classifer to each face ROI to determine "mask" or "no mask"
Show results

4.2 Block Diagram / Architecture:


Input Video
Choose Option
Artific Sight Sotfware
<<Uses>>
<<Uses>>
<<Uses>>
<<Uses>>
Mask Detection
Social Distance Detection
Object Detection
Emotions Detection
<<Includes>>
<<Includes>>
<<Includes>>
Preprocessing
Glcm Detect
Blob Detect
Generate
Preprocessing
Glcm Detect
Blob Detect
Generate
Preprocessing
Glcm Detect
Generate
Blob Detect
Preprocessing
Glcm Detect
Blob Detect
Generate

5.IMPLEMENTATION

PREPROCESSING

Digital Image Processing. Digital image processing deals with **manipulation of digital images through a digital computer**. It is a subfield of signals and systems but focus particularly on images. DIP focuses on developing a computer system that is able to perform **processing** on an **image**. The input of that system is a digital **image** and the system process that **image** using efficient algorithm

It allows a much wider range of algorithms to be applied to the input data and can avoid problems such as the build-up of noise and distortionduring processing.

1. Importing the image via image acquisition tools;
2. Analysing and manipulating the image;
3. Output in which result can be altered image

Image Pre-processing is a common name for operations with images at the lowest level of abstraction. Its input and output are intensity images. The aim of pre-processing is an improvement of the image data that suppresses unwanted distortions or enhances some image features important for further processing.

DIGITAL IMAGE PROCESSING

CLASSIFICATION OF IMAGES

There are 3 kinds of pix applied in Digital Image Processing. They are

1. Binary Image
2. Gray Scale Image
3. Colour Image

Binary picture

A binary photograph is a virtual photo that has most effective viable values for every pixel. Typically the two colorings used for a binary image are black and white despite the truth that any colors may be used. The color used for the item(s) in the photograph is the foreground color while the rest of the picture is the historic past coloration.

Binary pix are also called bi-stage or -stage. This approach that every pixel is stored as a single bit (zero or 1).This name black and white, monochrome or monochromatic are regularly used for this idea, however may also additionally designate any pics that have best one pattern in keeping with pixel, together with grayscale photos

Binary pics often upward push up in virtual image processing as masks or because the result of positive operations collectively with segmentation, thresholding, and dithering. Some input/output gadgets, which include laser printers, fax machines, and bi-level laptop shows, can most effective manage bi-degree pix.

Gray scale images

A grayscale Image is virtual photo is an picture in which the price of every pixel is a single sample, this is, it consists of satisfactory depth information. Images of this type, also referred to as black-and-white, are composed solely of sun shades of gray(0-255), various from black(0) on the weakest depth to white(255) at the maximum effective.

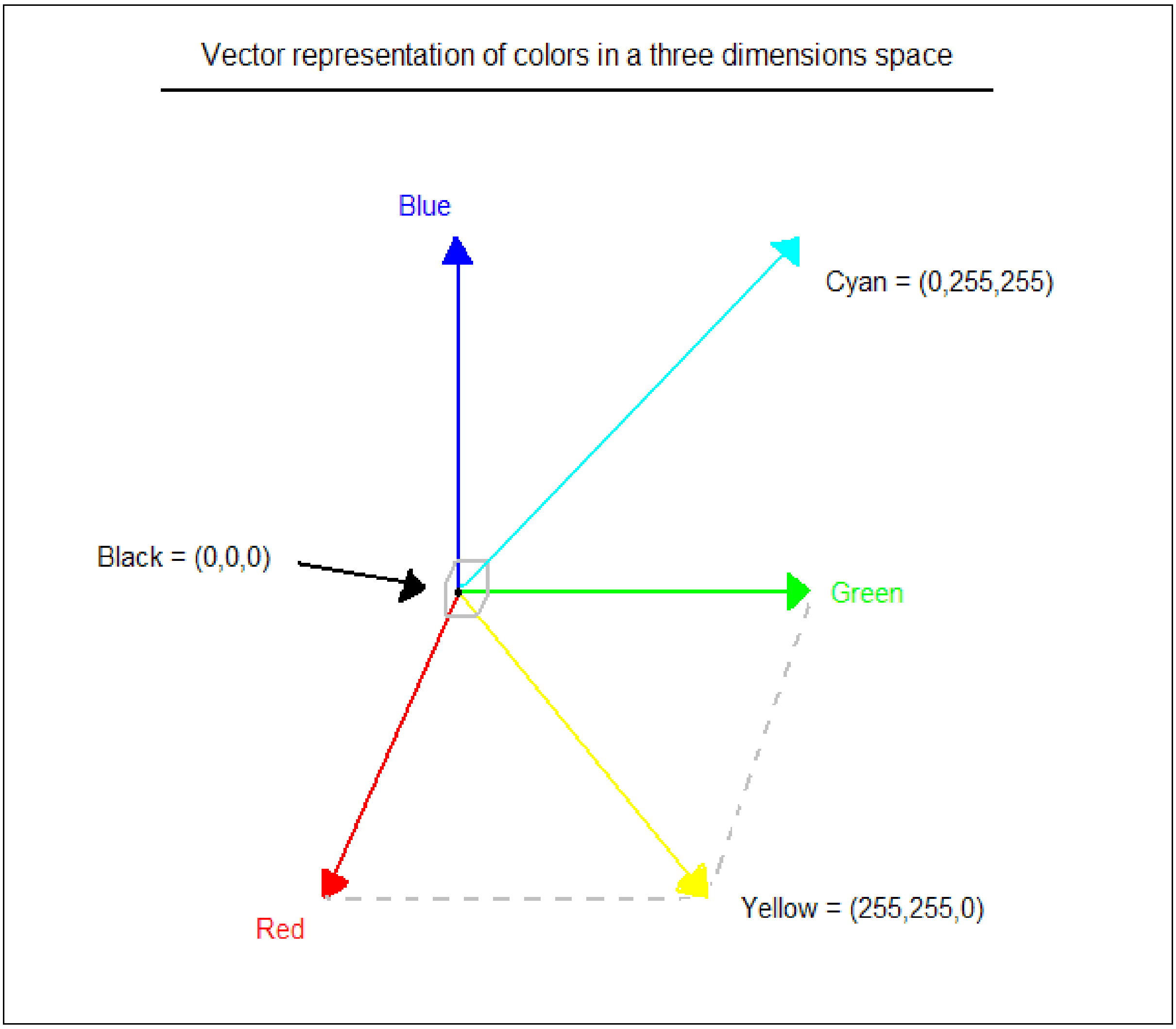
Grayscale snap shots are extremely good from one-bit black-and-white pix, which in the context of laptop imaging are pix with only the 2 sun sunglasses, black, and white (additionally known as bi-diploma or binary photos). Grayscale snap shots have many sun sun shades of grey in among. Grayscale photos are also referred to as monochromatic, denoting the absence of any chromatic version.

Grayscale pix are frequently the quit end result of measuring the intensity of moderate at every pixel in a single band of the electromagnetic spectrum (e.G. Infrared, seen mild, ultraviolet, and lots of others.), and in such instances they will be monochromatic proper at the same time as simplest a given frequency is captured. But moreover they may be synthesized from a whole color picture; see the phase approximately changing to grayscale.

Colour picture

A (digital) colour image is a virtual photograph that includes shade records for every pixel. Each pixel has a specific rate which determines its appearing shade. This fee is certified by means of manner of 3 numbers giving the decomposition of the color inside the 3 number one sun shades Red, Green and Blue. Any shade seen to human eye may be represented this manner. The decomposition of a colour inside the 3 number one hues is quantified through some of amongst 0 and 255. For instance, white may be coded as R = 255, G = 255, B = 255; black can be referred to as (R,G,B) = (0,0,0); and say, amazing pink may be : (255,0,255).

In great phrases, an photo is an intensive -dimensional array of colour values, pixels, each of them coded on three bytes, representing the 3 primary shades. This allows the image to encompass a complete of 256x256x256 = 16.Eight million particular shades. This approach is likewise referred to as RGB encoding, and is particularly tailor-made to human imaginative and prescient



Hue Saturation Process of RGB SCALE Image

From the above determine, sun shades are coded on 3 bytes representing their decomposition on the 3 primary colorings. It sounds obvious to a mathematician to right away interpret colors as vectors in a three size area in which every axis stands for one of the number one hues. Therefore we are able to gain of maximum of the geometric mathematical requirements to cope with our colors, which consist of norms, scalar product, projection, rotation or distance.

The identification of devices in an picture and this method may additionally possibly begin with photograph processing techniques collectively with noise removal, observed via the usage of (low-diploma) feature extraction to find out lines, areas and probable regions with sure textures.

The smart bit is to interpret collections of those shapes as single devices, e.G. Vehicles on a avenue, boxes on a conveyor belt or cancerous cells on a microscope slide. One motive this is an AI problem is that an item can seem very one-of-a-type while viewed from unique angles or underneath amazing lighting. Another problem is finding out what skills belong to what item and which can be information or shadows and so on. The human visible device plays these responsibilities specifically unconsciously however a laptop calls for skilful programming and masses of processing strength to method human commonplace overall performance. Manipulation of records inside the shape of an photograph through severa viable strategies. An image is generally interpreted as a -dimensional array of brightness values, and is maximum familiarly represented via way of such patterns as the ones of a photographic print, slide, tv display, or movie show display. An image may be processed optically or digitally with a laptop.

IMAGE ACQUISATION

Image Acquisition is to collect a digital photograph. To collect this requires an picture sensor and the functionality to digitize the sign produced thru the sensor. The sensor might be monochrome or coloration TV camera that produces an entire photo of the trouble area each 1/30 sec. The photograph sensor may also be line test virtual digicam that produces a single photo line at a time. In this situation, the gadgets movement beyond the road.

IMAGE ENHANCEMENT

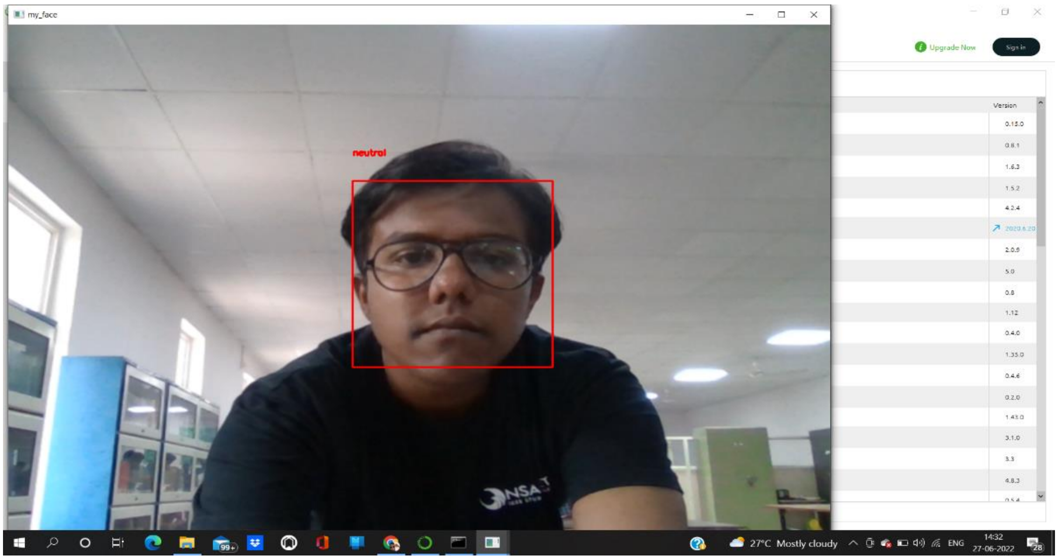
Image enhancement is most of the best and maximum appealing regions of digital image processing. Basically, the concept in the back of enhancement strategies is to perform detail that is obscured, or clearly to spotlight certain features of exciting an picture. A familiar instance of enhancement is at the same time as we growth the assessment of an photograph due to the fact "it appears higher." It is essential to remember the fact that enhancement is a completely subjective place of photo processing.

IMAGE RESTORATION

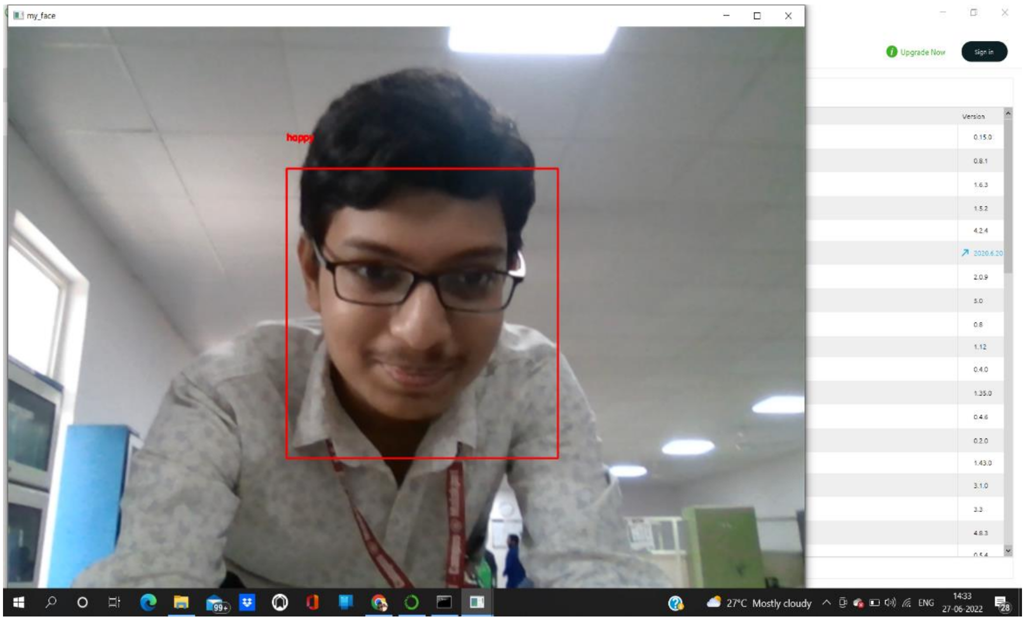
Image recuperation is a place that also deals with enhancing the appearance of an image. However, not like enhancement, that is subjective, photograph recuperation is goal, in the experience that restoration techniques will be predisposed to be based on mathematical or probabilistic fashions of picture degradation.

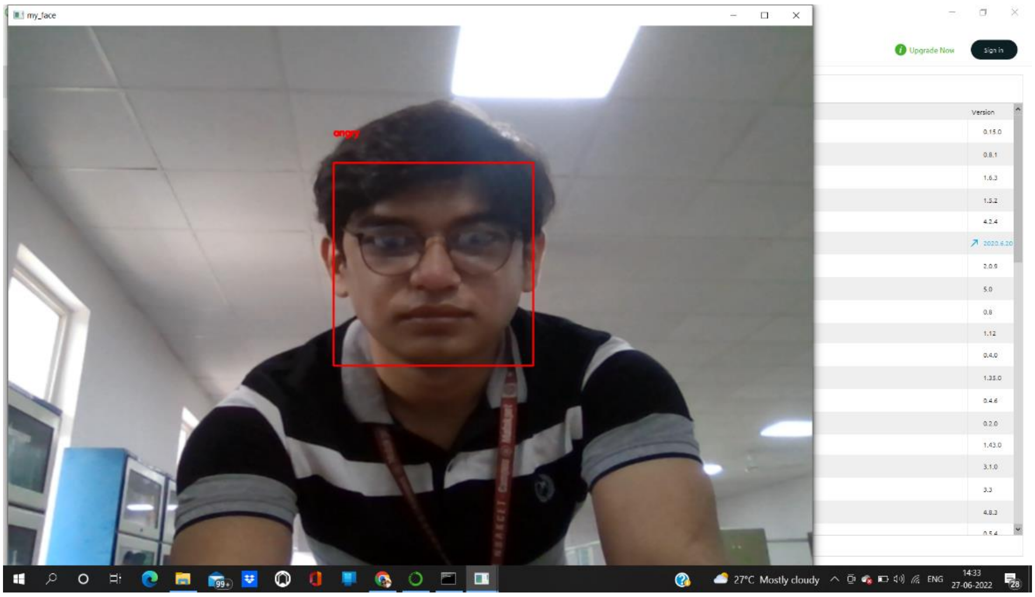
5.1 Screenshots

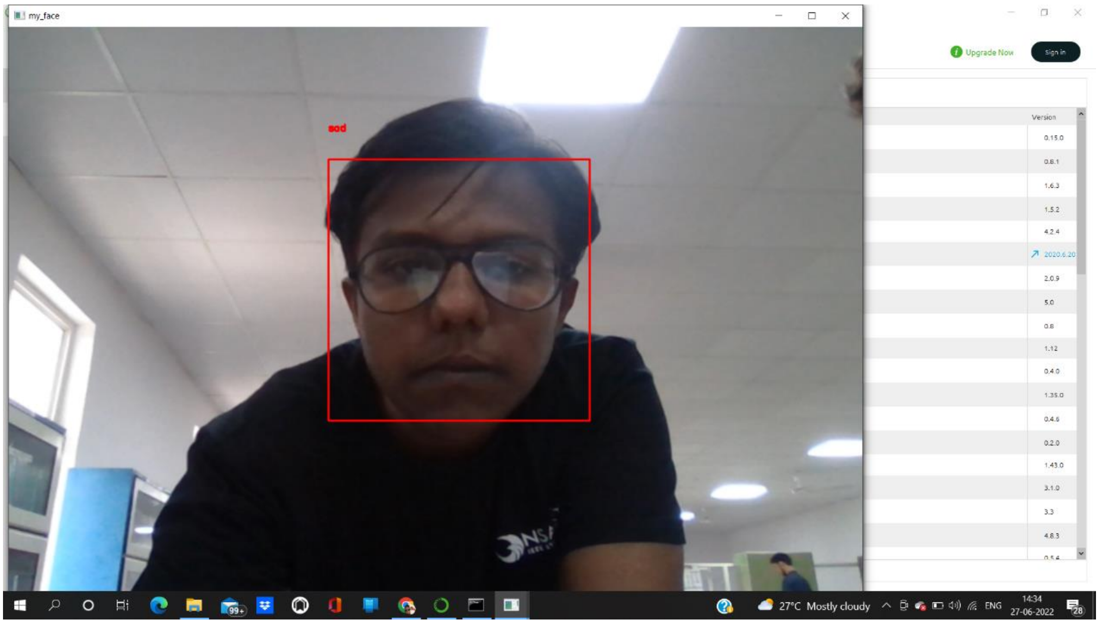
1.Emotion Recognization

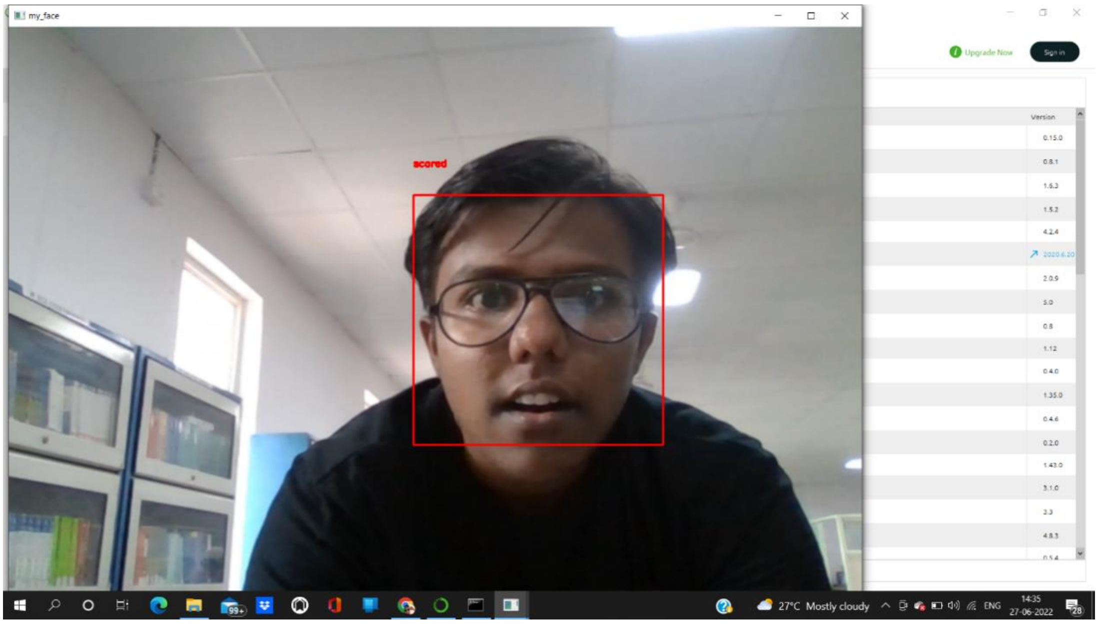


Version
0.15.0
0.8.1
1.6.3
1.5.2
4.2.4
2020.6.20
2.0.9
5.0
0.8
1.12
0.4.0
1.35.0
0.4.6
0.2.0
1.43.0
3.1.0
3.3
4.8.3
0.5.4

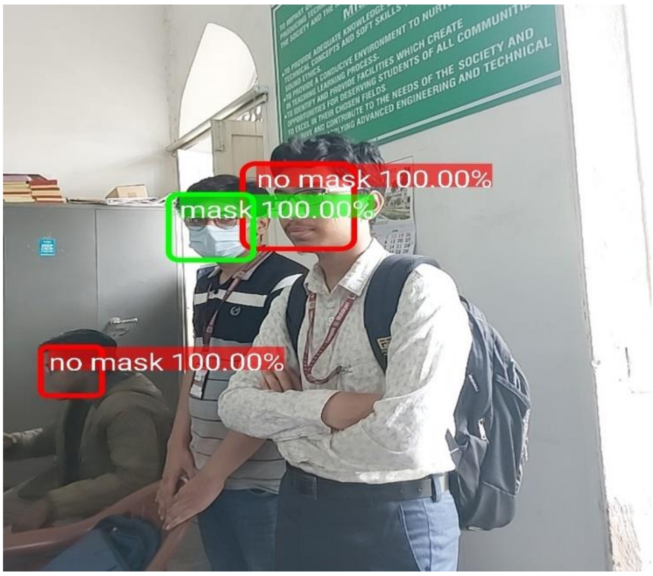

my_face
happy
Upgrade Now
Sign in
Version
27°C Mostly cloudy
ENG
14:33
27-06-2022

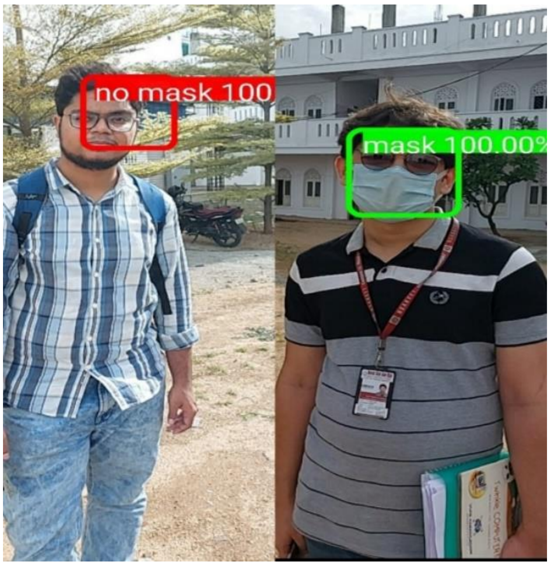

my_face
angry
Upgrade Now
Sign in
Version
27°C Mostly cloudy
ENG
14:33
27-06-2022

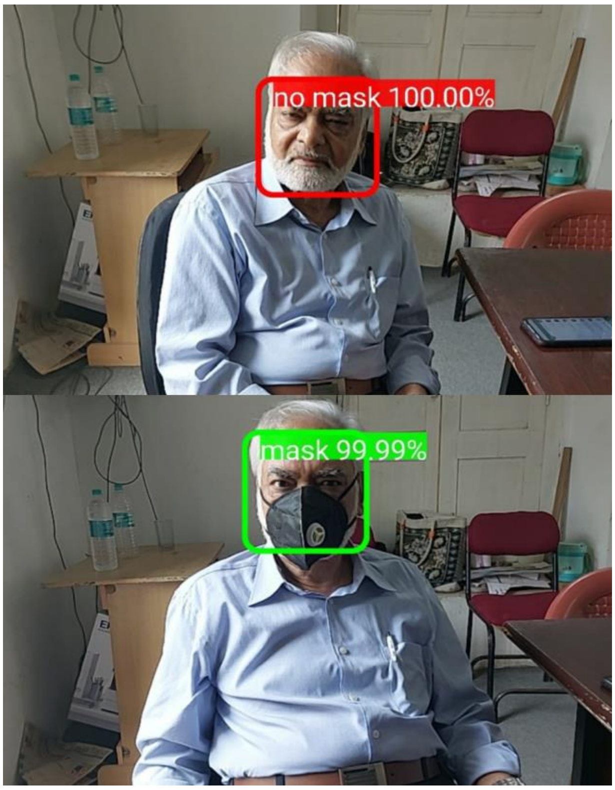

my_face
sad
Upgrade Now
Sign in
Version
0.15.0
0.8.1
1.6.3
1.5.2
4.2.4
2020.6.20
2.0.9
5.0
0.8
1.12
0.4.0
1.35.0
0.4.6
0.2.0
1.43.0
3.1.0
3.3
4.8.3
27°C Mostly cloudy
ENG
14:34
27-06-2022


my_face
scared
Upgrade Now
Sign in
Version
0.15.0
0.8.1
1.6.3
1.5.2
4.2.4
2020.6.20
2.0.9
5.0
0.8
1.12
0.4.0
1.35.0
0.4.6
0.2.0
1.43.0
3.1.0
3.3
4.8.3
27°C Mostly cloudy
ENG
14:35
27-06-2022

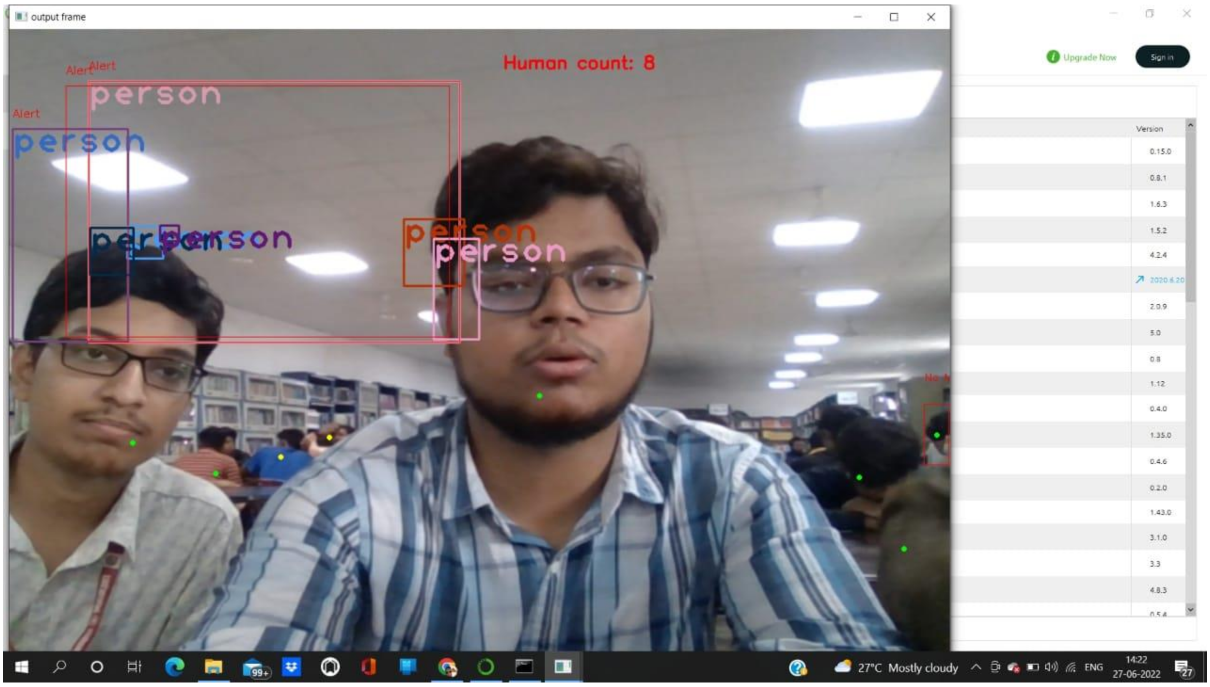
2.Face Mask Detection

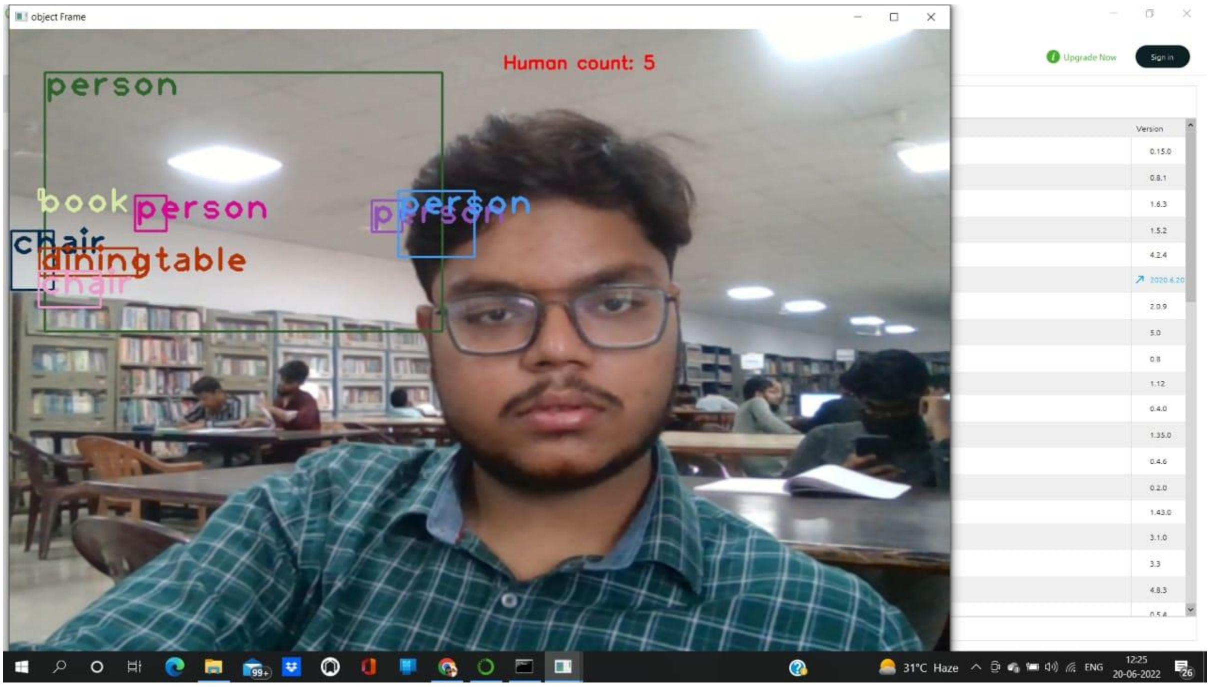


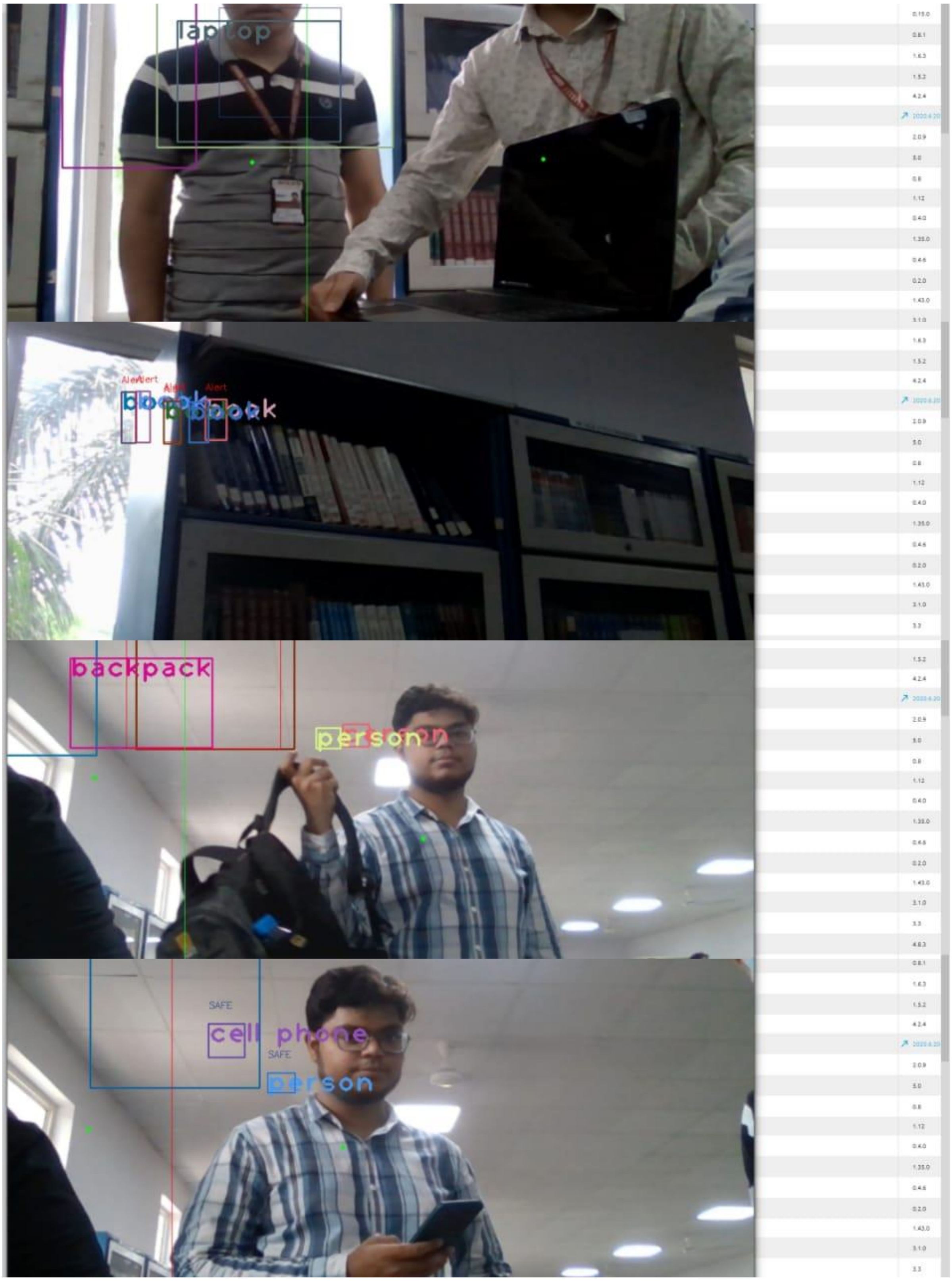


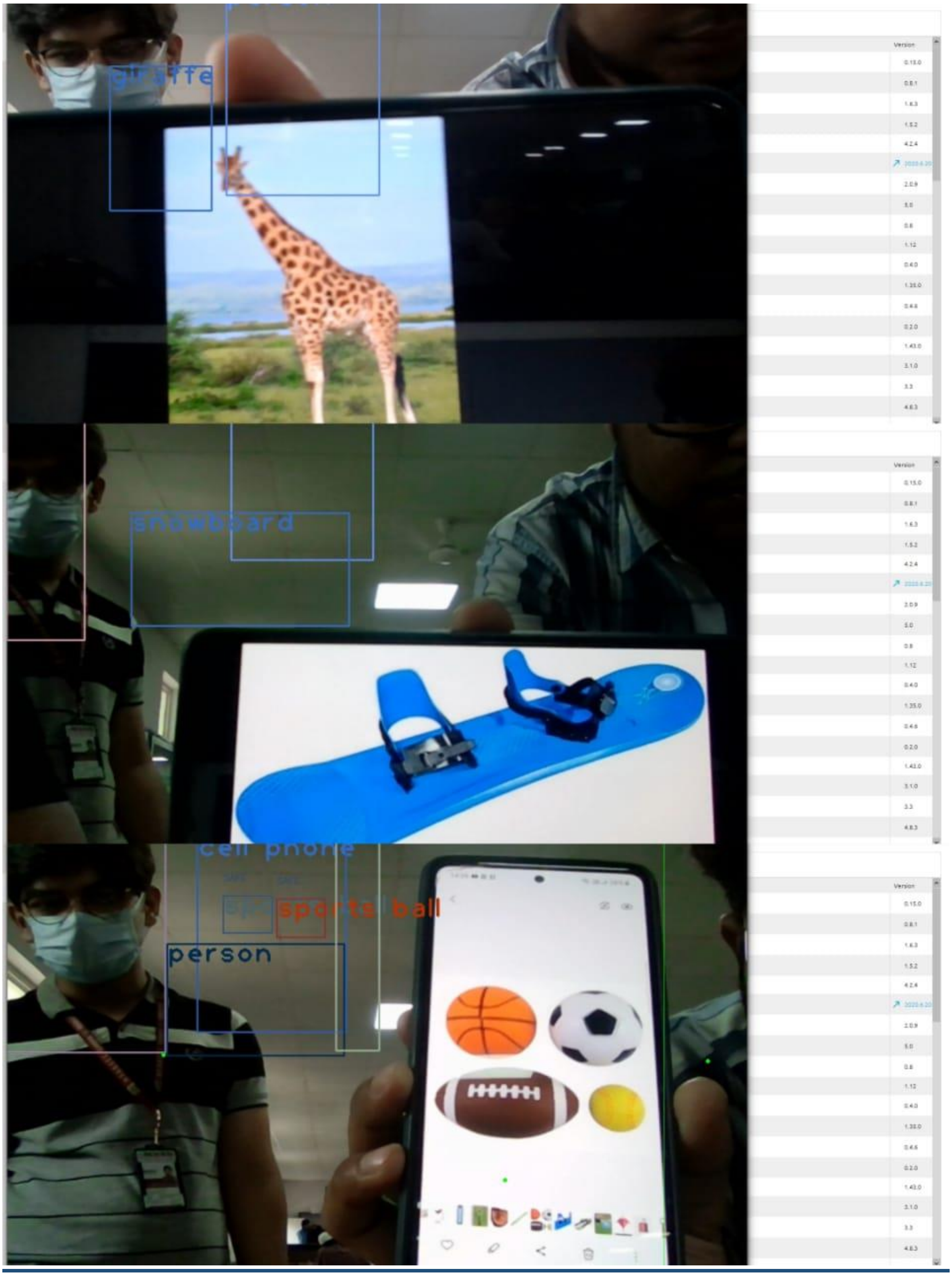


3.Object Detection:

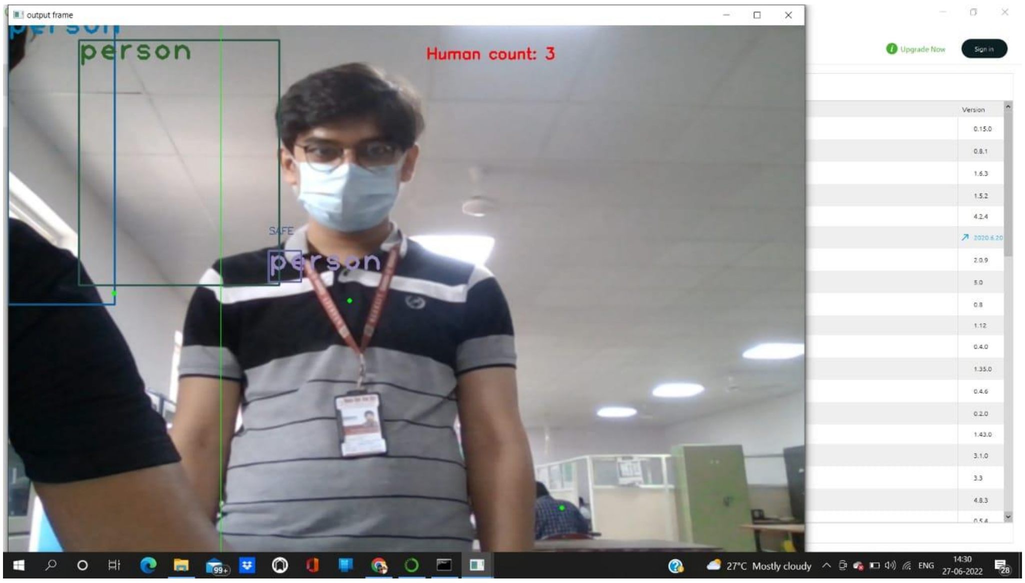





laptop
Alert
book
book
backpack
person
SAFE
cell phone
SAFE
person


giraffe
snowboard
cell phone
sports ball
person
Version
0.15.0
0.8.1
1.6.3
1.5.2
4.2.4
2020.6.20
2.0.9
5.0
0.8
1.12
0.4.0
1.35.0
0.4.6
0.2.0
1.43.0
3.1.0
3.3
4.6.3

4.Social Distance Detection:



6.Testing

6.1 Test Cases:

S.no	Input	Output	Result
1.	All four modules training data under single terminal window	Creates complexity while training	Fail
2.	One or both module at a time	Frame distortion	Fail
3.	All four module together under 3 terminal windows	Provides accurate result with few frame alignment distortion	Pass

7.CONCLUSION

In this we have proposed a real time object detection and track number of object using machine learning technique. Due to its powerful learning ability and advantages in dealing with occlusion, scale transformation and background switches, deep learning based object detection has been a research hotspot in recent years. This paper provides a detailed review on deep learning based object detection frameworks which handle different sub-problems, such as occlusion, clutter and low resolution, with different degrees of modifications on ANN.in this we used GLCM feature extraction.

7.1 Applications

- Object detection in Retail
- Animal detection in Agriculture
- People detection in Security
- Human behaviour understanding
- Synthetic human expressions
- Vehicle detection with AI in Transportation
- Medical feature detection in Health care
- The system can detect when the person is not using a mask
- Social distancing aims to decrease or interrupt transmission of COVID-19 in a population by minimising contact between potentially infected individuals and healthy individuals

7.2 Future Scope

The future of object detection technology is in the process of proving itself, and much like the original Industrial Revolution, it has the potential, at the very least, to free people from tedious jobs that will be done more efficiently and effectively by machines. The future scope of Emotion recognition is to increase the efficiency of emotion recognition systems in terms of accuracy, work has to be performed using RGB datasets formed under uncontrolled conditions, deep neural networks as emotion classifiers, compound emotions, micro expressions and multi modal behavioural systems such as body movements, facial expressions, voice etc. to form robust automatic recognition systems. The face mask detection can be integrated into any high-resolution video surveillance devices and not limited to mask detection only. The model can be extended to detect facial landmarks with a facemask for biometric purposes. Social distancing detector can include AR (Augmented Reality) libraries from openCV and use them.

Appendix A: Glossary

SRS : Software Requirements Specification

SVM : Support Vector Machine

Tkinter : This framework provides Python users with a simple way to create GUI elements using the widgets found in the Tk toolkit.

Tk : Tk widgets can be used to construct buttons, menus, data fields, etc. in a Python application.

ML: Machine Learning

References

[1] X. Wu, D. Hong, J. Chanussot, Y. Xu, R. Tao, and Y. Wang, ‘‘Fourier-based rotation-invariant feature boosting: An efficient framework for geospatial object detection,’’ IEEE Geosci. Remote Sens. Lett., vol. 17, no. 2, pp. 302–306, Feb. 2020.

[2] X. Wu, D. Hong, J. Tian, J. Chanussot, W. Li, and R. Tao, ‘‘ORSIm detector: A novel object detection framework in optical remote sensing imagery using spatial-frequency channel features,’’ IEEE Trans. Geosci. Remote Sens., vol. 57, no. 7, pp. 5146–5158, Jul. 2019.

[3] S. Ren, K. He, R. Girshick, and J. Sun, ‘‘Faster R-CNN: Towards realtime object detection with region proposal networks,’’ IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[4] Y. Li, S. Li, C. Chen, A. Hao, and H. Qin, ‘‘Accurate and robust video saliency detection via self-paced diffusion,’’ IEEE Trans. Multimedia, vol. 22, no. 5, pp. 1153–1167, May 2020.