

Extracting and Identifying individuals from media files using supervised and unsupervised learning methods

OVERVIEW:

THE GOAL OF THE PROJECT ANALYSING VIDEOS OF SURVEILLANCE TO EXTRACT AND IDENTIFY HUMAN FACES PRESENT OR PARTICIPATED IN THE VIDEO TO SAVE TIME FOR HUMANS INSPECTING THE VIDEO AND DISPLAY THE IDENTITIES OF THE KNOWN INDIVIDUALS AND PRESENT THE UNKNOWN INDIVIDUALS AS WELL.

The current development takes video files and employs the **Intel Open-Source Computer Vision Library - CV2** Library methods to extract the image frames from the video. Using different face recognition algorithms like **HAAR Cascades** and **Convolutional Neural Networks** we identify faces present in individual frames and deploy the optimal solution as per efficiency.

The output of the current model will contain the faces present in the video frames. This output works as raw data for our identification model. Here we will employ the **DLib- ClustImage** to cluster similar faces based on the feature extraction using different **unsupervised learning** methods like **Principal Component Analysis (PCA)**, **Histogram Oriented Gradient (HOG)**, etc. The similar faces are clustered together giving us the unique individuals present in our media file.

This will be the Test data for our identification model which employs **supervised classification** to identify and name the individuals and mention those who are not present in our data.

JOURNEY:

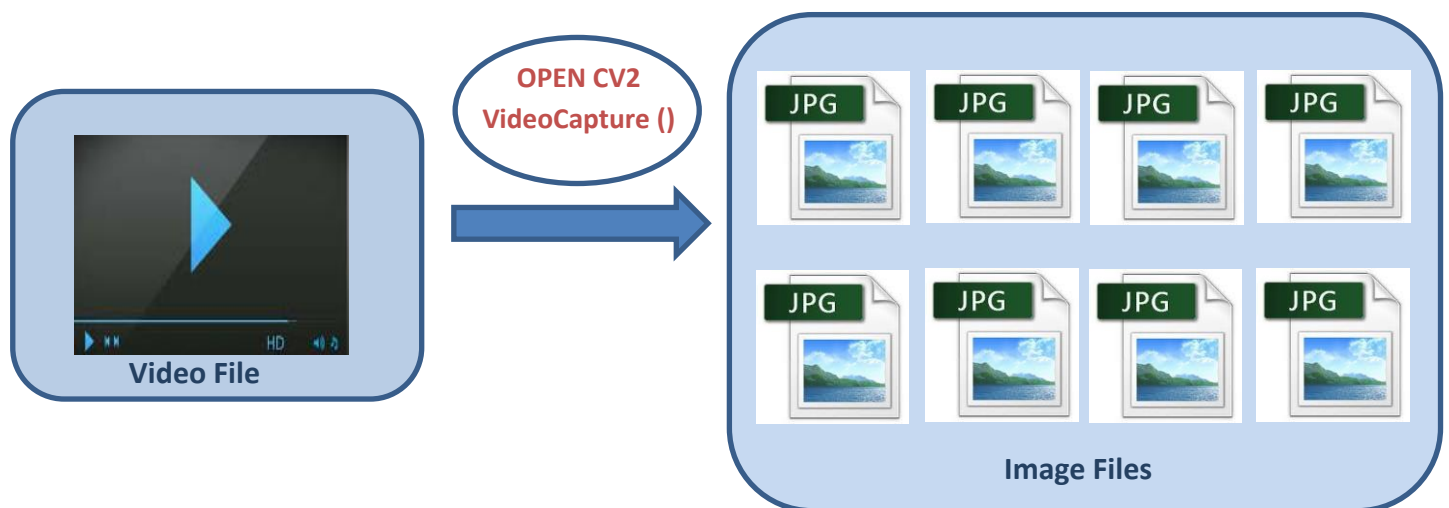
The project goes under several phases of Data Analysis or steps as mentioned below for cleaning, transforming, processing, visualizing, and modelling data to get results:

1. Data Collection
2. Data Exploration
3. Data Pre-processing
4. Feature Extraction
5. Training Model
6. Testing Model
7. Embracing Failure
8. Evaluation

1. Data Collection

The project uses the TensorFlow OpenCV i.e., CV2 library to extract frames from the video. The frames data collected acts as the raw data for the project and input for the next method. The raw frames contain frames that may or may not contain human images hence it includes the number of redundancies and null data which requires to be pre-processed before feeding as training data for our report.

The Intel Open-Source Computer Vision Library popularly known as **OPEN CV2 or CV2** is used for machine perception exposes the method- **VideoCapture** which returns frames of images from the video, hence acting as the data collection phase of the project providing us the raw image files i.e., raw data.



2. Data Exploration

The project uses the OpenCV i.e., CV2 library to extract frames from the video. The frames data collected acts as the raw data for the project and input for the next method. The raw frames contain frames that may or may not contain human images hence it includes the number of redundancies and null data which requires to be pre-processed before feeding as training data for our report.

3. Data Pre-processing

Data pre-processing includes preparing the input for the Unsupervised clustering models. The current data comprises images that contain frames from video files. These frames may or may not contain face data hence it is required to extract faces from the frames which contain faces. This process can be co-related to the elimination of null values i.e., extra frames which do not contain faces and the other portions of images that do not contain facial data. Hence extraction of images works as data cleaning by eliminating the null or non-facial images/portions.

We explored multiple methods to recognize faces in images and picked the optimal method in terms of output and time complexity. Based on the algorithm there are different methodologies to identify facial data in images. Following methods were employed to identify facial data:

1. *HAAR Cascades:*
2. *Convolutional Neural Networks (CNN)*

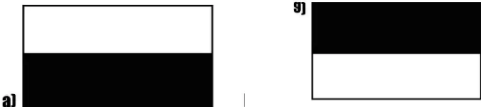

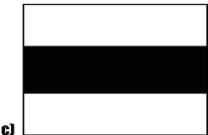


HAAR Cascades:

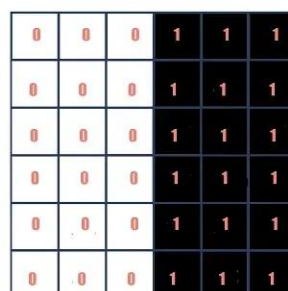
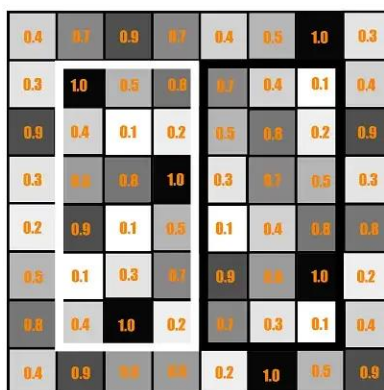
HAAR cascades originally known as Viola-Jones Face Detection Technique is an early methodology to extract objects from images. It was developed before the Deep Learning methods became popular to identify objects in images.

The technique makes advantage of Viola and Jones's edge or line-detecting features (*referred: research paper "Rapid Object Detection using a Boosted Cascade of Simple Features" published in 2001*).

It has a wide range of utilities based on the trained files like models for face detection, eye detection, upper body and lower body detection, license plate detection etc. A feature selection window scans image in block of pixels instead of individual pixels resulting into a weak identifier and hence a faster and efficient object recognition model.

The algorithm uses different feature windows to identify different shapes and borders of images.

	<p>These set of 2 rectangles help into identifying the horizontal borders across images with change of intensities in pixels.</p>
	<p>These set of 2 rectangles help into identifying the vertical borders across images with change of intensities in pixels.</p>
	<p>These set of 3 rectangles help into identifying the darker regions between lighter regions across images with change of intensities in pixels.</p>
	<p>These set of 3 rectangles help into identifying the lighter regions between darker regions across images with change of intensities in pixels.</p>
	<p>These set of 4 rectangles help into identifying the diagonals across images with changes in intensities of pixels.</p>



SUM OF THE DARK PIXELS/NUMBER OF DARK PIXELS -
SUM OF THE LIGHT PIXELS/NUMBER OF THE LIGHT PIXELS

$$\frac{(0.7 + 0.4 + 0.1 + 0.5 + 0.8 + 0.2 + 0.3 + 0.7 + 0.5 + 0.1 + 0.4 + 0.8 + 0.9 + 0.6 + 1.0 + 0.7 + 0.3 + 0.1)/18}{(1.0 + 0.5 + 0.8 + 0.4 + 0.1 + 0.2 + 0.6 + 0.8 + 1.0 + 0.9 + 0.1 + 0.5 + 0.1 + 0.3 + 0.7 + 0.4 + 1.0 + 0.2)/18}$$

$$0.51 - 0.53 = -0.02$$

This rectangle is a sample representation of an image with pixel values 0.0 to 1.0.

The rectangle at the center is a haar kernel which has all the light pixels on the left and all the dark pixels on the right

haar calculation is done by finding out the difference of the average of the pixel values at the darker region and the average of the pixel values at the lighter region. If the difference is close to 1, then there is an edge detected by the haar feature.

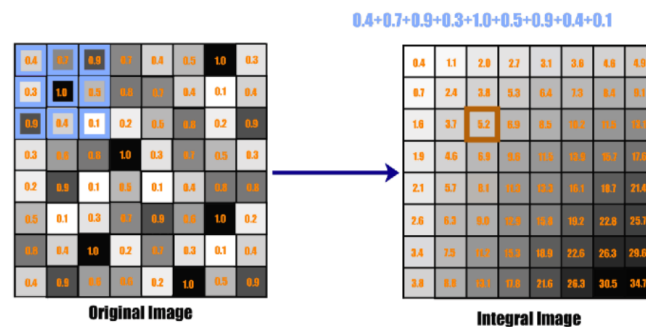
How does HAAR cascade detect edges?

Calculate the total of all the picture pixels located in the haar feature's darker and lighter areas, respectively.

And then find out their difference.

If edge separates dark pixels on right and light pixels on left or vice versa; i.e., HAAR value ~ 1

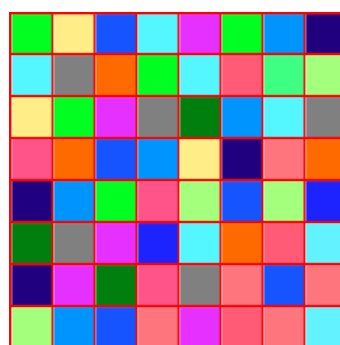
we say that there is an edge detected



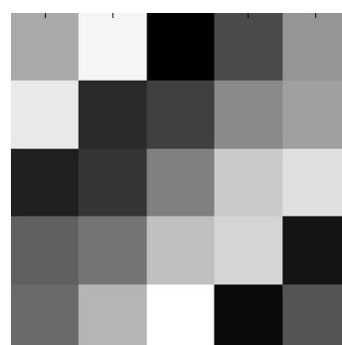
Why grey scaling is important?

Every image can be represented as a matrix or a 2D array that have a row and column number. The cells are the pixels of the image. A 10x10 image will contain 100 pixels.

A 8 bit colour image, a pixel contains a combination of 3 data -Red, Green, Blue components (RGB), and each component ranges from 0 to 255 based on the intensity.



Color Image



Greyscaled Image

For a grayscale image, each pixel contains only one component and ranges from 0 to 255

The number of data that can be represented or need to be processed in each case varies drastically. **Thus, making a colour image to grey scale reduces the data processing overhead by 3 times. Therefore, increasing processing speed.**

Why scaling factor is important?

Scale factor: The size by which the shape is enlarged or reduced is called as its scale factor. It is used when we need to increase the size of a 2d shape, such as circle, triangle, square, rectangle, etc.

$y = Kx$ is an equation, then K is the scale factor for x .

Hence $y \propto x$, i.e., y is proportional to x where k is the proportionality constant.

(Dimensions of Original Shape) \times (scale Factor) = (Dimension of new shape)

e.g.: If we must find the enlarged triangle similar to the smaller triangle, we need to multiply the side-lengths of the smaller triangle by the scale factor.

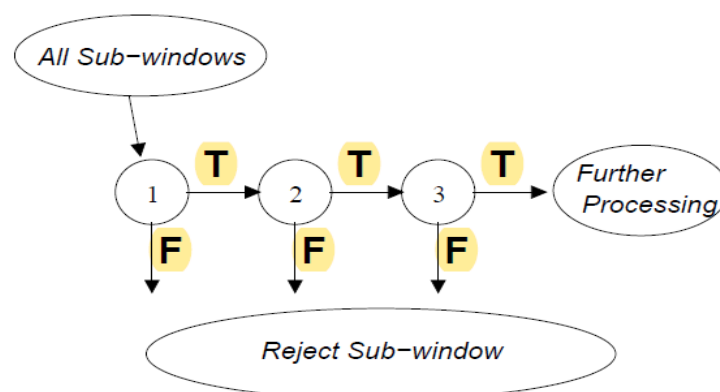
Similarly, if we must draw a smaller triangle similar to bigger one, we need to divide the side-lengths of the original triangle by scale factor.

Scale factoring is reduced image size for each iteration to this would generate a greater number of layers since the image is downsized each time by only 5%. With this large number of layers, it can detect objects in images, irrespective of their scale in image and location.

When our image has faces in different sizes it becomes important to analyse image with different scales.

Cascading + Facial Recognition:

This algorithm for constructing a cascade of classifiers which achieves increased detection performance while radically reducing computation time. The important component is that smaller, and therefore more efficient, boosted classifiers can be constructed which reject many of the negative sub-windows while detecting almost all positive instances (i.e., the threshold of a boosted classifier can be adjusted so that the false negative rate is close to zero). Simpler classifiers are used to reject most sub windows before more complex classifiers are called upon.



Features will again run on the training images to detect if there's a facial feature present or not. Another method added is attentional cascade which discards a window of pixels if it does not pass the feature detection hence saving the overhead of further calculations with inclusion of the window.

HAAR algo:

It first identifies the first set of features and filters primary windows and discards the windows that fail to pass feature. In the second recursion to identify features a more complex window will be analysed from the initially passed windows. Hence it saves time and processing by eliminating the failed windows on feature test (a crucial point to consider as data cleaning under data pre-processing).

Convolutional Neural Networks (CNN):

Convolutional Neural Networks (CNN) is one of the most powerful algorithms in Deep Neural Networks. CNN is a type of Artificial Neural Network (ANN) used in image recognition and processing which is specially designed for processing data(pixels).

Here we have used **DLIB** to implement CNN importing **cnn_face_detection_model_v1** and using a trained model to identify faces- **mmod_human_face_detector.dat**.

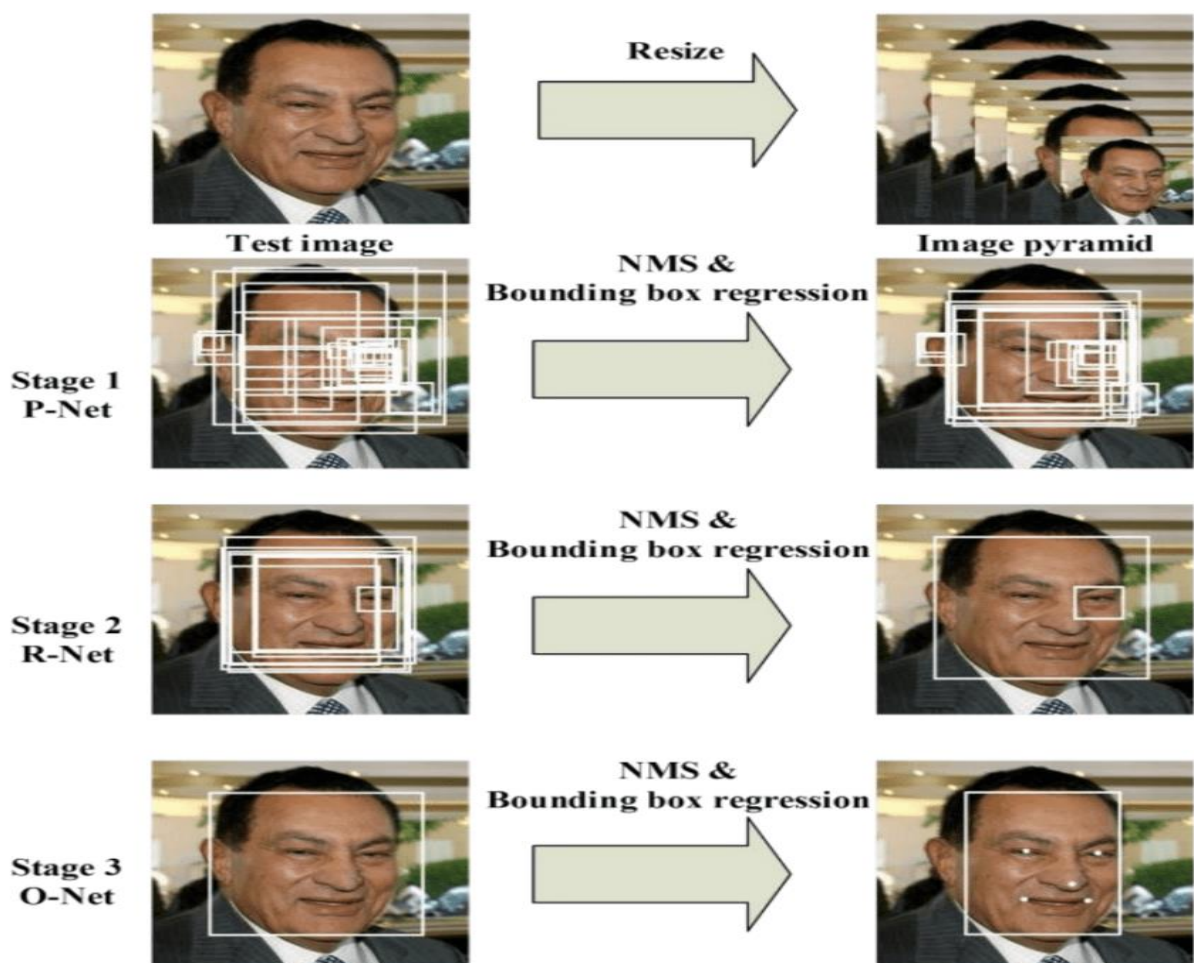
The goal is to get a deep neural network to produce numbers that describe a face (known as face encodings or features).

Popular approaches called *the “Multi-Task Cascaded Convolutional Neural Network,”* or *MTCNN* for short, described by Kaipeng Zhang, et al. in the 2016 paper titled *“Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks.”*

MTCNN is capable of also recognizing other facial features such as eyes and mouth, called landmark detection.

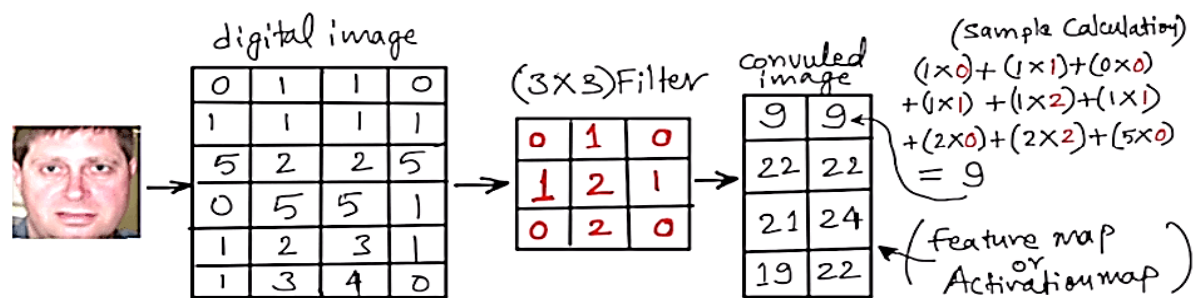
The network uses a cascade structure with three network- first the image is rescaled to a range of different sizes (called an image pyramid)

- then the first model (Proposal Network or P-Net) proposes candidate facial regions
- the second model (Refine Network or R-Net) filters the bounding boxes
- the third model (Output Network or O-Net) proposes facial landmarks



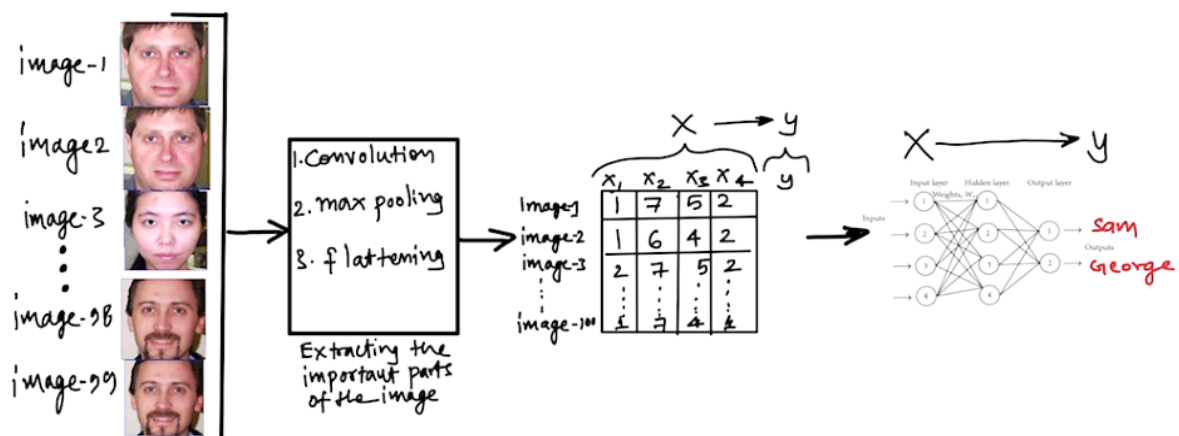
Pipeline for the Multi-Task Cascaded Convolutional Neural Network

Using the machine learning library DLIB for implementing CNN. Use pre-trained models for facial recognition



The Convolution step in CNN

CNN mimics the way humans see images, by focussing on one portion of the image at a time and scanning the whole image. CNN boils down every image as a vector of numbers.



The algorithm takes arrays of pixels of image for input layer of the neural network.

The first thing you should do is feed the pixels of the image in the form of arrays to the input layer of the neural network (MLP networks used to classify such things).

The hidden layers carry Feature Extraction by performing various calculations and operations. The architecture of a convolutional network typically consists of four types of layers: convolution, pooling, activation, and fully connected. So finally, there is a fully connected layer that you can see which identifies the exact object in the image.

Increasing or decreasing the convolution, max pooling, and hidden ANN layers and the number of neurons in it. With more layers or neurons added, the model becomes slower.

Four Layers of CNN

There are three types of layers in Convolutional Neural Networks:

- 1) **Convolutional Layer:** In a typical neural network each input neuron is connected to the next hidden layer. In CNN, only a small region of the input layer neurons connects to the neuron hidden layer.
- 2) **Pooling Layer:** The pooling layer is used to reduce the dimensionality of the feature map. There will be multiple activation & pooling layers inside the hidden layer of the CNN.
- 3) **Fully Connected Layer:** Fully Connected Layers form the last few layers in the network. The input to the fully connected layer is the output from the final Pooling or Convolutional Layer, which is flattened and then fed into the fully connected layer.
- 4) **ReLU Activation Layer:** The convolution maps are passed through a nonlinear activation layer, such as Rectified Linear Unit (ReLU), which replaces negative numbers of the filtered images with zeros.

There are redundant images in the output folder of the faces detected from the video frames. If these data is fed to the classification-identification algorithm it creates a lot of redundant and repetitive calculation resulting into costing processing and time. Hence it becomes very important to cluster all the similar faces together and then classify each cluster to identify identities.

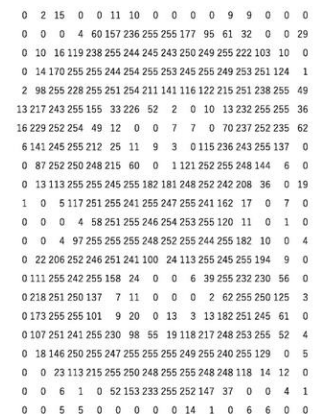
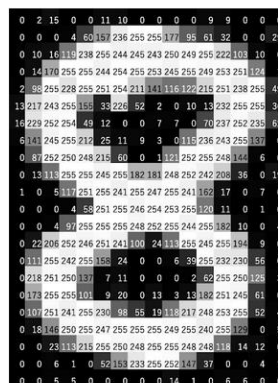
For clustering images there is a need for strong feature extraction technique because images co-relate based on the features of the image.

Machines store images in the form of a matrix of numbers. The size of this matrix depends on the number of pixels we have in any given image.

Let's say the dimensions of an image are 180 x 200 or $n \times m$. These dimensions are basically the number of pixels in the image (height x width).

These numbers, or the pixel values, denote the intensity or brightness of the pixel. Smaller numbers (closer to zero) represent black, and larger numbers (closer to 255) denote white. You'll understand whatever we have learned so far by analysing the below image.

The dimensions of the below image are 22 x 16, which you can verify by counting the number of pixels:



A coloured image is typically composed of multiple colours and almost all colours can be generated from three primary colours – red, green and blue.



Colour Image

141	142	143	144	145
151	152	153	154	155
161	162	163	164	165

35	36	37	38	39	173	174	175
45	46	47	48	49	183	184	185
55	56	57	58	59	193	194	195
65	66	67	68	69			

31	32	33	34	35	76	77	78	79
41	42	43	44	45	86	87	88	89
51	52	53	54	55				
61	62	63	64	65				
71	72	73	74	75				
81	82	83	84	85				

G

B

R

G

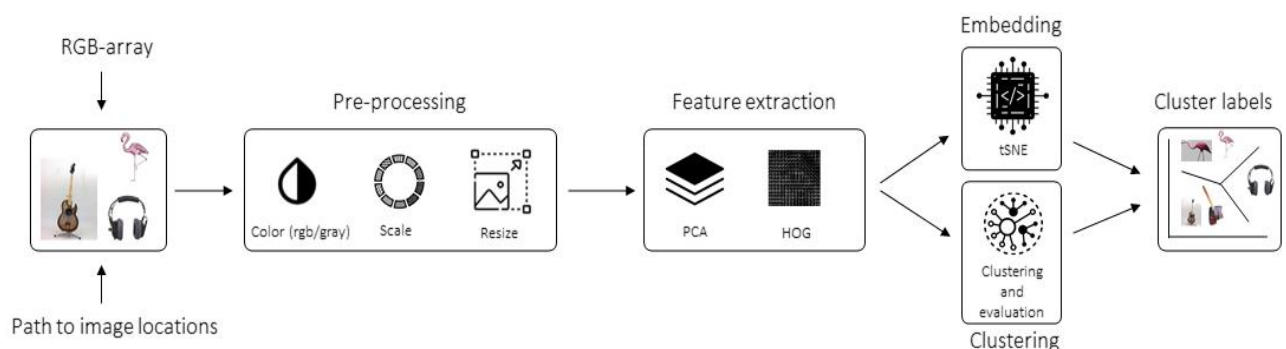
B


```

0 2 15 0 0 11 10 0 0 0 0 9 9 0 0
0 0 0 4 60 157 236 255 255 177 95 61 32 0 0 29
0 10 16 119 238 255 244 245 243 250 249 255 222 103 10 0
0 14 170 255 255 244 254 255 253 245 255 249 253 251 124 1
2 98 255 228 255 251 254 211 141 116 122 215 251 238 255 49
13 217 243 255 155 33 226 52 2 0 10 13 232 255 255 36
16 229 252 254 49 12 0 0 7 7 0 70 237 252 235 62
6 141 245 255 212 25 11 9 3 0 115 236 243 255 137 0
0 87 252 250 248 215 60 0 1 121 252 255 248 144 6 0
0 13 113 255 255 245 255 182 181 248 252 242 208 36 0 19
1 0 5 117 251 255 241 255 247 255 241 162 17 0 7 0
0 0 0 4 58 251 255 246 254 253 255 120 11 0 1 0
0 0 4 97 255 255 255 248 252 255 244 255 182 10 0 4
0 22 206 252 246 251 241 100 24 113 255 245 255 194 9 0
0 111 255 242 255 158 24 0 0 6 39 255 232 230 56 0
0 218 251 250 137 7 11 0 0 0 2 62 255 250 125 3
0 173 255 255 101 9 20 0 13 3 13 182 251 245 61 0
0 107 251 241 255 230 98 55 19 118 217 248 253 255 52 4
0 18 146 250 255 247 255 255 255 249 255 240 255 129 0 5
0 0 23 113 215 255 250 248 255 255 248 248 118 14 12 0
0 0 6 1 0 52 153 233 255 252 147 37 0 0 4 1
0 0 5 5 0 0 0 0 0 0 14 1 0 6 6 0 0

```

It works using a multi-step process of carefully pre-processing the images, extracting the features, and evaluating the optimal number of clusters across the feature space. The optimal number of clusters can be determined using well known methods such as *silhouette*, *dbindex*, and *derivatives* in combination with clustering methods, such as *agglomerative*, *kmeans*, *dbscan* and *hdbscan*.



We will be using **Principal Component Analysis (PCA)** for clustering in our project.

It is a statistical approach to reduce variables in image to concentrate features. Every image in the training set is represented as a linear combination of weighted eigenvectors called eigenfaces.

Through PCA, the data from any subject can be represented using a linear model, where $\bar{\mathbf{w}}$ is the average velocity field for all the subjects:

$$\mathbf{w} = \bar{\mathbf{w}} + \Phi \mathbf{b},$$

$$\bar{\mathbf{w}} = \frac{1}{N} \sum_{j=1}^N \mathbf{w}_j$$

Principal Components- are the columns of the matrix, matrix that represent the modes of variation of the velocity fields. They are computed from the $2M \times 2M$ covariance matrix S :

$$\mathbf{S} = \frac{1}{N} \sum_{j=1}^N (\mathbf{w}_j - \bar{\mathbf{w}})(\mathbf{w}_j - \bar{\mathbf{w}})^T$$

Cluster analysis:

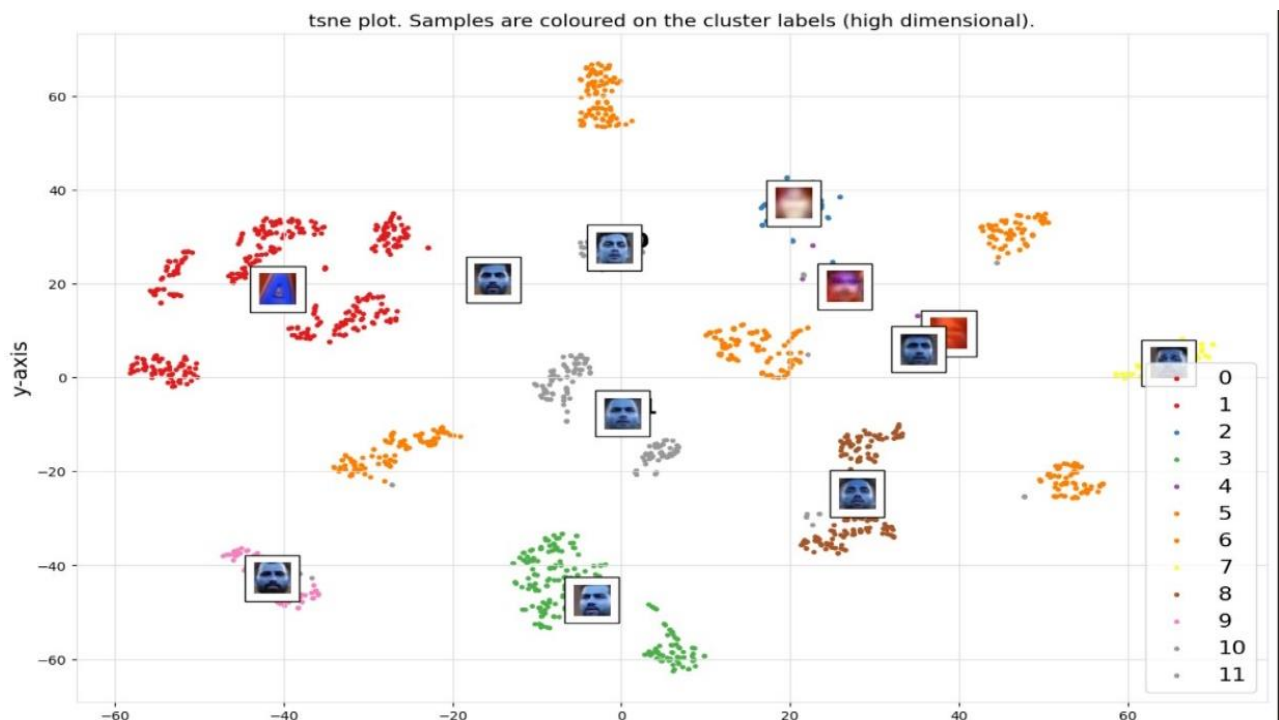
Velocity field of the j th subject as a vector $\mathbf{w}_j = [u(j)1, \dots, u(j)M, v(j)1, \dots, v(j)M]^T$, and let the Pearson correlation between the j th, i th subject velocity vectors be denoted as $\rho_{ji} = \text{cov}(\mathbf{w}_j, \mathbf{w}_i) / \text{std}(\mathbf{w}_j) \text{std}(\mathbf{w}_i)$, where cov stands for covariance. The similarity metric between two subjects is defined as :

$$d(\mathbf{w}_j, \mathbf{w}_i) = 1 - \rho_{ji}$$

Two clusters D and D^* which contain n and n^* subjects, respectively. Then the average linkage (distance) between two clusters is measured as:

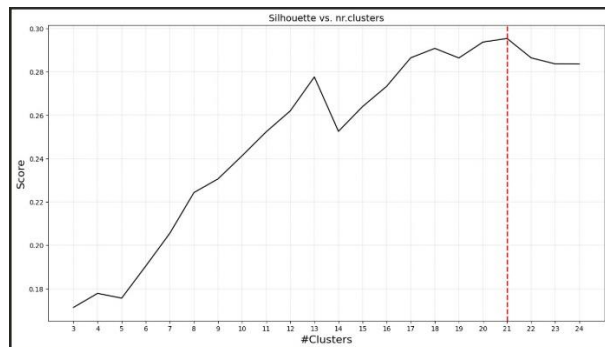
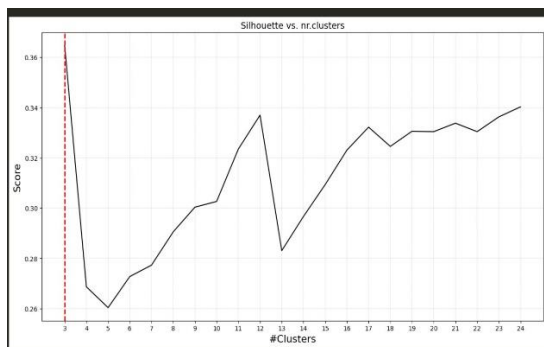
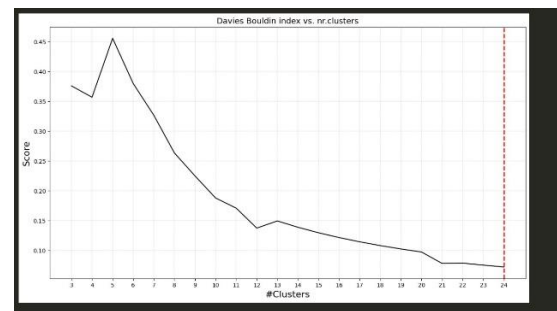
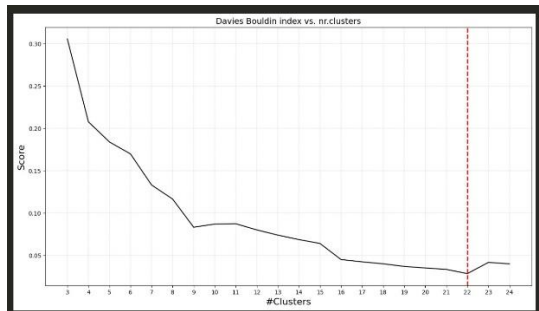
$$d_{avg}(D, D^*) = \frac{1}{n \cdot n^*} \sum_{V_j \in D} \sum_{V_i \in D^*} d(\mathbf{w}_j, \mathbf{w}_i)$$

It is understood that the larger the calculated distance value, the greater the difference between subjects (clusters). The cluster analysis begins with each subject as its own cluster and at each stage chooses the “best” merge of two subjects or of two clusters of subjects if their distance is minimized until, in the end, all subjects are merged into a single cluster.



Analysing different evaluation methods for clustering

Clusters vs Score plots



5. Training Model

Training Models are supervised model which learn from a predefined dataset about the predictions.

The model is given a training set with several inputs along with the outputs for the corresponding inputs. Hence it can identify, recognise the individuals which will be given as test data.

The Current recognition model we are using is the SKLearn Support Vector Machine Classifier. It classifies the test images into one of the pre-existing labels which matches the image features.

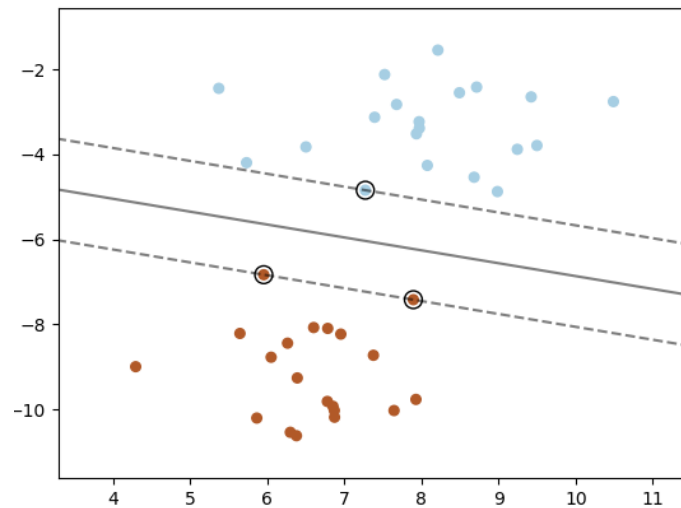
How does SVM read Image data?

The DLIB library exposes **face_recognition.face_encodings** function to turn image data into scaler encoded data. These scaler encodings of images are then provided to our Support Vector Classifier to train.

In this SVM algorithm, we plot each data image encoding as a point in n-dimensional space (where n is the number of features) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiates the two classes very well.

A support vector machine constructs a hyper-plane or set of hyper-planes in a high or infinite dimensional space. This hyperplane will be used for classification. Intuitively, a good separation between encodings of different training sets is achieved by the hyper-plane that has the largest distance to the nearest training data points of any class of encodings, **The larger the margin between hyperplane and encoding the lower the generalization error of the classifier.**

The figure below shows the decision function for a linearly separable problem, with three samples on the margin boundaries, called "support vectors".

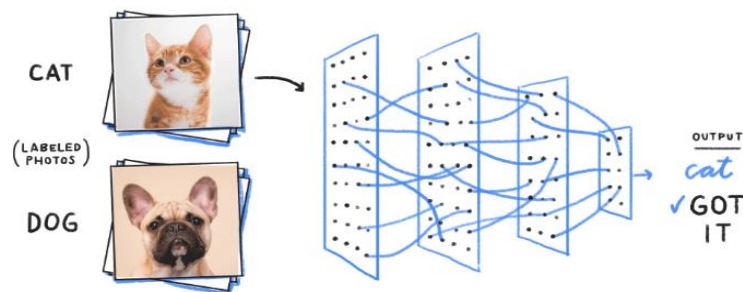


6. Testing Model

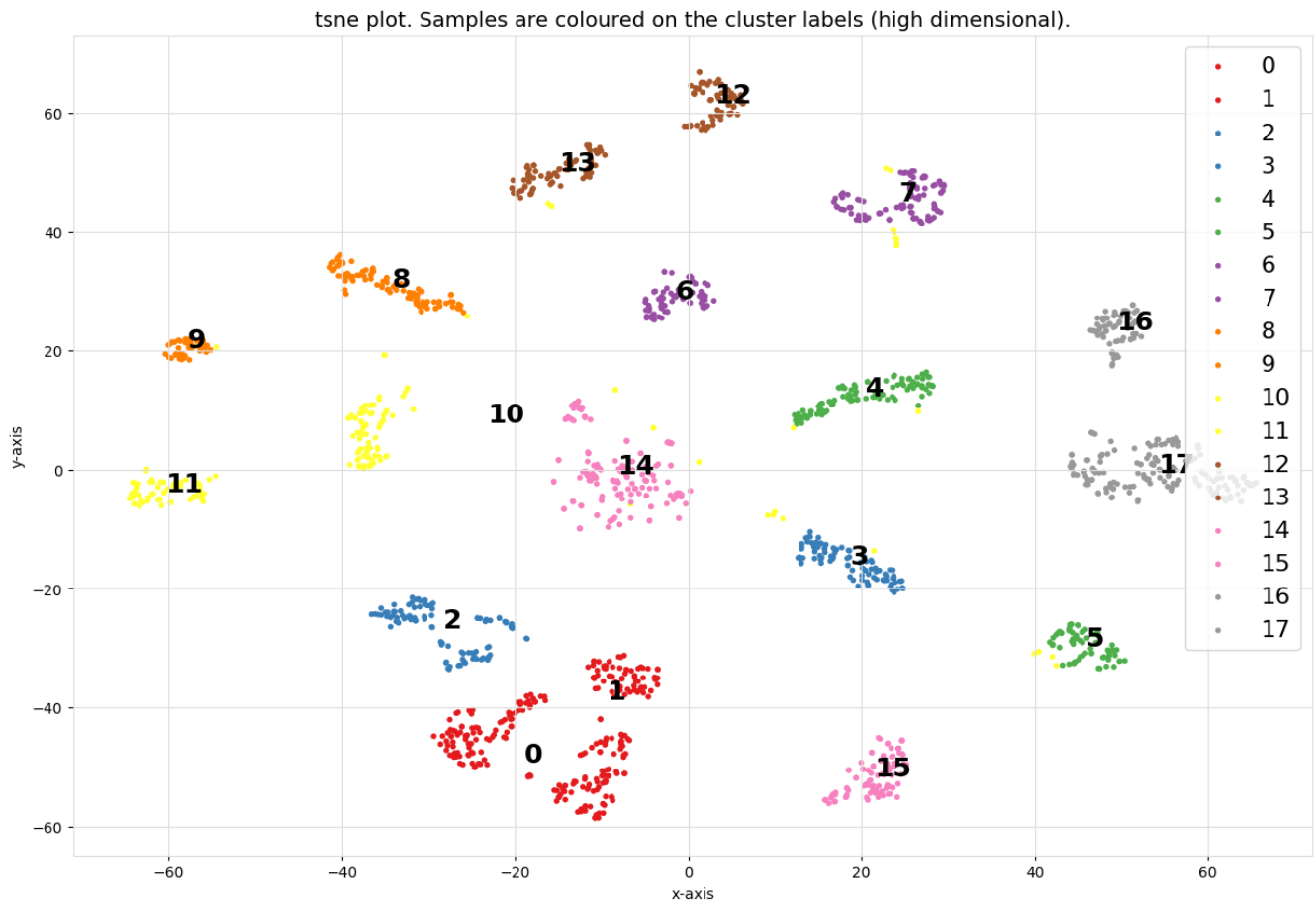
The test data for Model is a subset of data which will be provided as input to our model and the output will be a label of the cluster our test image belongs to. Here the test data to our Support Vector Machine Classifier is the facial images.

The Unsupervised clustering clustered our facial data into different clusters. Those clusters contain all the similar images from the pool of facial images extracted from the frames of video. These clusters will provide one image from each cluster. This set of unique images will work as the test data for our classification model.

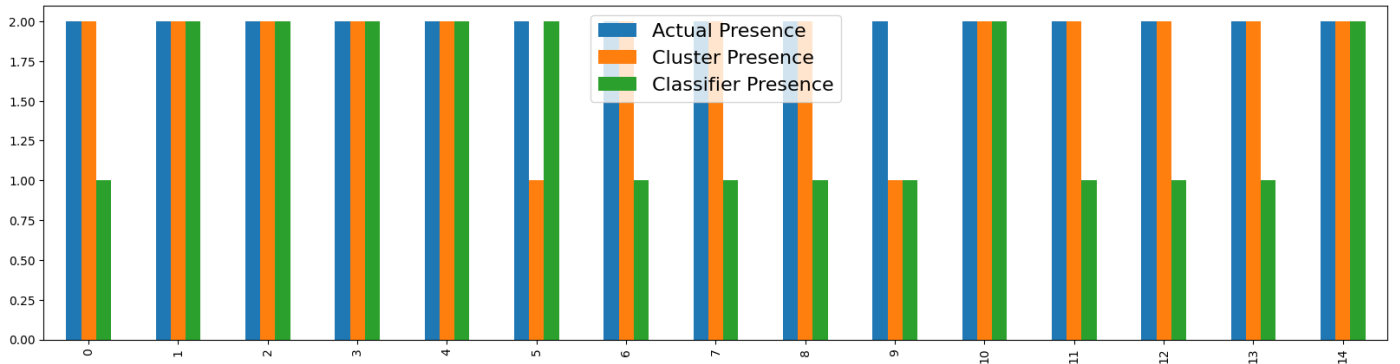
The classification model will compare the test image against the classifier which was trained on our training data. It will label our test image with one of the labels provided to our training images or will give us output that it did not match the training data.



	ActualNames	Cluster_Names	ClassifierNames
0	Ashwin	Ashwin	NaN
1	Bhuvneshwar Kumar	Bhuvneshwar Kumar	Bhuvneshwar Kumar
2	Hardik Pandya	Hardik Pandya	Hardik Pandya
3	Ishan Kishan	Ishan Kishan	Ishan Kishan
4	Jasprit Bumrah	Jasprit Bumrah	Jasprit Bumrah
5	K L Rahul	NaN	K L Rahul
6	Mohammad Shami	Mohammad Shami	NaN
7	Rahul Chahar	Rahul Chahar	NaN
8	Ravindra Jadeja	Ravindra Jadeja	NaN
9	Rishabh Pant	NaN	NaN
10	Rohit Sharma	Rohit Sharma	Rohit Sharma
11	Shardul Thakur	Shardul Thakur	NaN
12	Suryakumar Yadav(Sky)	Suryakumar Yadav(Sky)	NaN
13	Varun Chakravorty	Varun Chakravorty	NaN
14	Virat Kohli	Virat Kohli	Virat Kohli



Results



The presented plot represents the presence of the individuals in Actual Video File, Unsupervised Clusters and the classifier recognition. The values with 2 are describe as they are present while the value 1 represents that they are absent in the following set. Here the numbers of individuals present in the original video file were 14. While the individuals correctly identified and are present in both clusters and classifier recognition are 6. Hence the prediction accuracy for the current dataset is :

$$(6 / 14) * 100 = 42.85 \% \text{ i.e. our prediction accuracy}$$

Also there are false positives which are present in the classification recognition but not in clusters:

$$(1 / 14) * 100 = 7.14 \% \text{ i.e. our false positive percentage}$$

7. Evaluation

Variance in Facial recognitions using HAAR Cascading with respect to the min-neighbours value:

minNeighbours	face count
0	818
1	19
2	17
3	15
4	15
5	15
6	15
7	15
8	15
9	15
10	15
11	14
12	14
13	14
14	14
15	14
16	14
17	14
18	14
19	14
20	14
21	14
22	14
23	13
24	13
25	13
26	13
27	13
28	13
29	13

The parameter – number of min neighbours drastically impacts the identification of faces in HAAR cascade face recognition. With change in value from 0 to 1 the face recognition dropped from 818 to 19 which shows it dropped a large number of non facial images in the facial image collection.

Time taken By HAAR Cascade Vs Convolutional Neural networks to determine faces from a single image:

	Algorithm Runtime	Faces Count
cnn	3min \pm 4.32 s per loop (mean \pm std. dev. of 3 ...	15
haarcascade	204 ms \pm 18.6 ms per loop (mean \pm std. dev. of...	15

We employed CNN and HAAR cascade into face recognition from image.

Both the algorithms successfully identified 15 faces which is equal.

The time is calculated based on an average of 3 iterations of the algorithms and the results are drastically different.

CNN took 3 mins on average of 3 iterations to identify faces while HAAR took 204 milliseconds.

Hence we decided to employ the HAAR Cascading to recognise faces instead of CNN, However both the methods are strongly capable and based on different algorithms and techniques.

Future Scope

This project can be extended to include live face detection and recognition. The scale of the project plays major role in further enhancements of the projects and can be made dynamic implementations. The technology may be utilized to provide greater security and monitoring in the defence ministry, airports, and any other significant locations.

Implementing and understanding various evaluating methods for Machine Learning and Data Mining Applications. Though there is a lot of advancement in facial recognition systems, there are still some concerns that's needs to be addressed. The issues such as Privacy, Accuracy, Infrastructure and Reliability/ Efficiency pertaining challenges needs to be acknowledged.

Facial recognition technology has a promising future. According to forecasters, this technology is anticipated to expand at an impressive rate and produce significant income in the years to come. The two most important areas that will be significantly impacted are security and surveillance. Private businesses, public spaces, and educational institutions are some more places that are now embracing it. In order to prevent fraud in debit/credit card transactions and payments, particularly those made online, it is anticipated that shops and financial institutions will also embrace it in the upcoming years. This technique would close the gaps in the widely used yet ineffective password scheme. Robots that use face recognition technology may eventually make an appearance. They may be useful in finishing tasks that are impractical or challenging for people to complete.