**INTRODUCTION:**

A facial recognition system is a technology capable of identifying or verifying a person from a digital image or a video frame from a video source. There are multiple methods in which facial recognition systems work, but in general, they work by comparing selected facial features from a given image with faces within a database. It is also described as a Biometric Artificial Intelligence based application that can uniquely identify a person by analysing patterns based on the person's facial textures and shape. Here we use the face\_recognition library that recognizes and manipulates faces from Python or from the command line with the world’s simplest face recognition library. This library is built using dlisb‘s state-of-the-art face recognition with deep learning.– a modern C++ toolkit that contains several machine learning algorithms that help in writing sophisticated C++ based applications. The model has an accuracy of 99.38% on the labelled  [Faces in the Wild](http://vis-www.cs.umass.edu/lfw/) benchmark.

**Procedure:**

Here we use modules like OpenCV ,face\_recognition, and NumPy.

Classification of the face is done by followed steps:

The model used comes with the partial implementation of the FaceNet model - a unified embedding for face recognition and clustering. first we take the frame of the given video source and perform basic image processing, frame alignment. we can also compute the facial landmarks which enables us to pre-process and align the face present in the frame:

phase 1- -face alignment: this is the process of detecting the geometric structure of the face. added to the we do perform the attempt to obtain the canonical alignment of the face based on the translation, rotation and scaling of the input feed. using face alignment has demonstrated its worth in increase face recognition accuracy in some pipelines,

phase 2- Adaptive video enhancement and video upscaling: Basic upscaling is the easiest method of stretching a lower resolution image onto a bigger display. Pixels from the lower resolution image are copied and repeated to fill out all the pixels of the upper resolution display. Filtering is applied to smooth the image and round out unwanted jagged edges which will come into sight because of the stretching. The result is a picture that matches on a 4K display, but can often appear muted or blurry.

AI -based upscaling Traditional upscaling starts with a low-resolution image and tries to improve its visual quality at higher resolutions. AI upscaling takes a different approach: Given a low-resolution image, a deep learning model predicts a high-resolution image that would downscale to look like the original, low-resolution image. To predict the upscaled images with high accuracy, a neural network model must be trained on countless images. The deployed AI model can then take low-resolution video and produce incredible sharpness and enhanced details no traditional scaler can recreate. Edges look sharper, hair looks scruffier and landscapes pop with striking clarity.

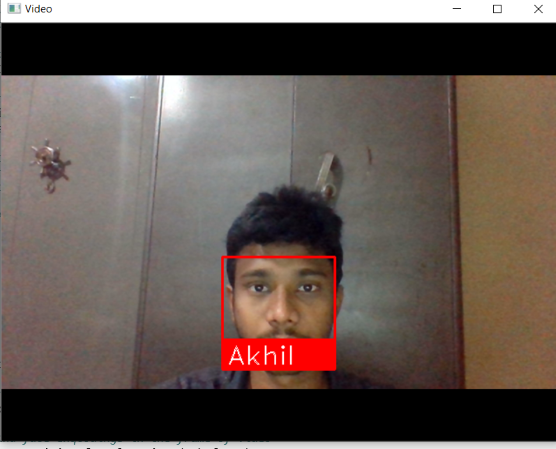
Phase 3- deep neural network architecture - the faceNet deep learning model that computes 128-d embedding that quantifies the face. The main key aspects of this model are: the input taken by the model and the triplet loss function.

Triplet loss function-Triplet loss is a loss function for artificial neural networks where a baseline (anchor) input is compared to a positive (truthy) input and a negative (falsie) input. The distance from the baseline(anchor) input to the positive (truthy) input is minimized, and the distance from the baseline (anchor) input to the negative (falsie) input is maximized.

the deep learning model computes the 128-d embedding vectors of each face and then the weights of the network are updated via triplet loss function. The 128-d embeddings of the anchor and positive image lie closer together and pushing the embeddings for the negative image father and the network is able to learn to quantify faces and return highly robust and discriminating embeddings suitable for face recognition.

The model is trained by above 3 phases by provided image dataset. During the time of testing or recognition the image is taken as frame of the given video source. Now, once we have encoded each image into a feature vector, the problem becomes much simpler. If the distance between feature vector (128-d embeddings) of image provided in database and feature vector (128-d embeddings) of test image is quite small then the image is classified as known person (It displays a tag as known person name bounded by face) else it classified as unknown person.

Result:



The model is trained by image provided as input during initial stage. In testing phase, we provide the frame of the given video source to the model and it classifies as known person i.e. Akhil.

When we provide as an image which is not in image dataset. Hence the model classifies as an unknown person.



Conclusion: