**Methodology**

* Generally, the structures of face recognition system consist of three major steps:-

1. Face Detection
2. Facial Feature Extraction
3. Emotion recognition

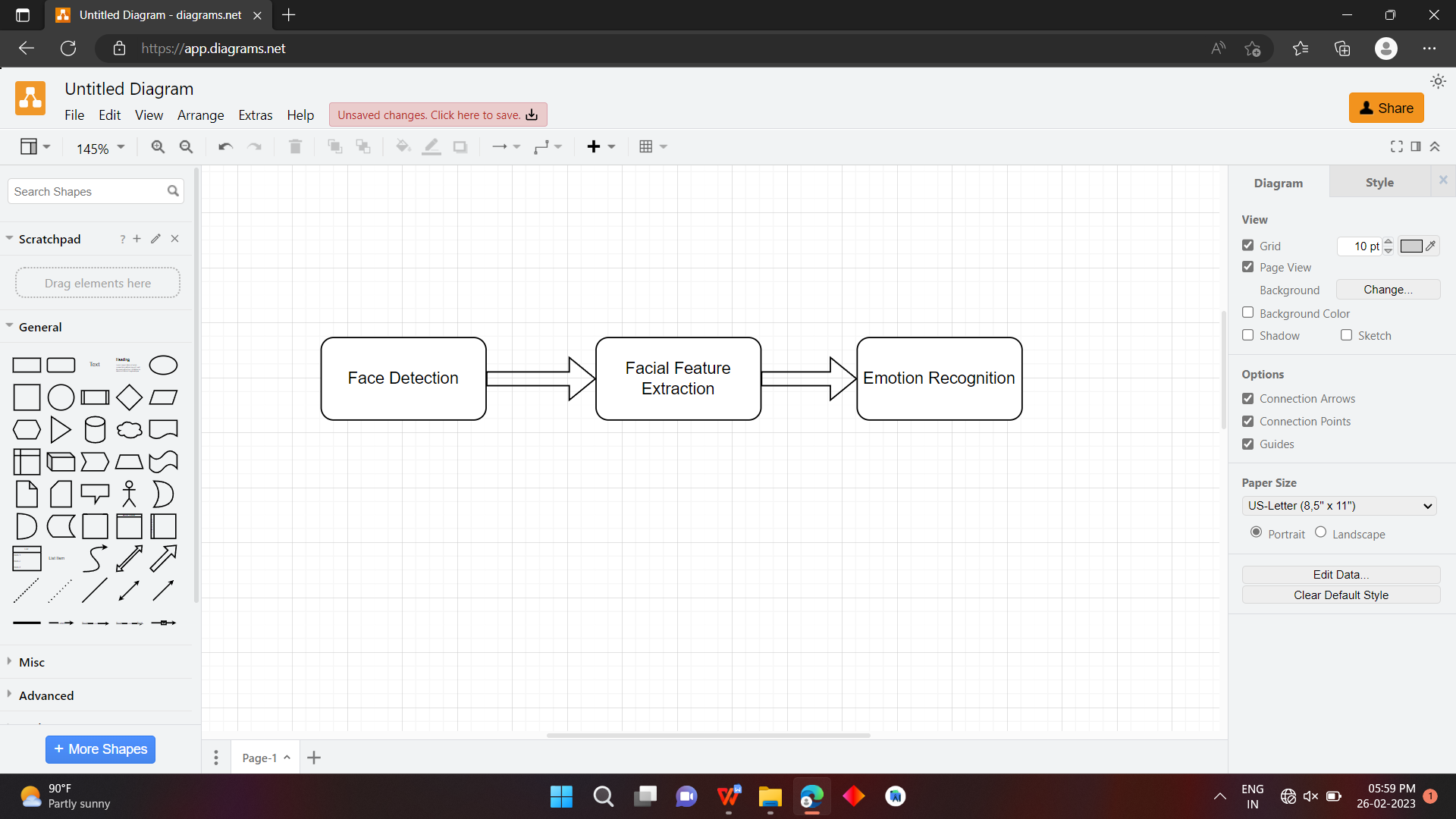


Fig 1: Project Outline

1. **Face Detection:-**

Face detection is the main module for any face recognition system which outlines and mark boundaries of face regions from messy images(Fig 1). Face detection is a type of application classified under “Computer vision” technology. It is the process in which algorithms are developed and trained to properly locate faces or objects , in images. These can be in real time from a video camera or from photographs. An example where this technology is used are in airport security systems. In order to recognize a face, the camera software must fist detect it and identify the features before making an identification. Likewise, when Facebook makes tagging suggestions to identify people in photos it must first locate the face. On social media apps like Snapchat, face detection is required to augment reality which allows users to virtually wear dog masks using fancy filters. another use of face detection is in smartphone face ID security.

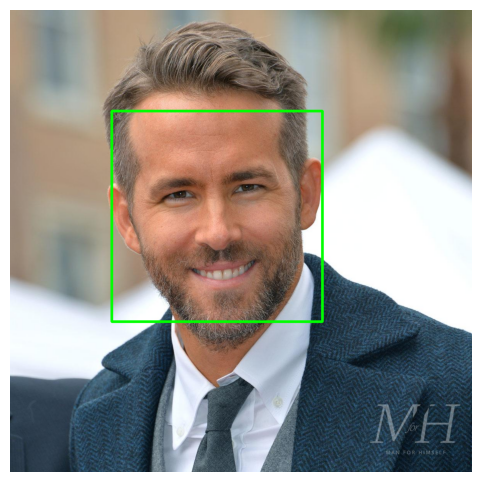
 

Fig 2: Face Detection

Object Detection using Haar feature-based cascade classifiers is an effective method proposed by Paul Viola and Michael Jones in the 2001 paper, “Rapid Object Detection using a Boosted Cascade of simple Features”. It is a machine learning based approach in which a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images.

Here we are work with face detection, Initially, the algorithm needs a lot of positive images and negative images of faces to train the classifier. Then we need to extract features from it. The algorithm works by breaking down an image into small rectangular regions and then applying a series of classifiers to each regions to determine whether it contains the object of interest or not. Once the Haar-like features have been computes for each region of the image, the Ada-boost algorithm is used to train a series of classifiers that can be used to distinguish between positive and negative examples of the object of interest.

In this project, we Implemented a system for locating face in digital images.Face detection uses classifiers, which are algorithms that detects what is either a face(1) or not a face(0) in image.

The Haar cascade algorithm uses a set of haar-like features to identify regions in an image that are likely to contain an object. These features are simple rectangular patterns that can be used to distinguish between the object and its background. On observation, the eye area appears darker than the check area while the nose area looks brighter than the eye area. You can visualize these feature in the below figure 3.

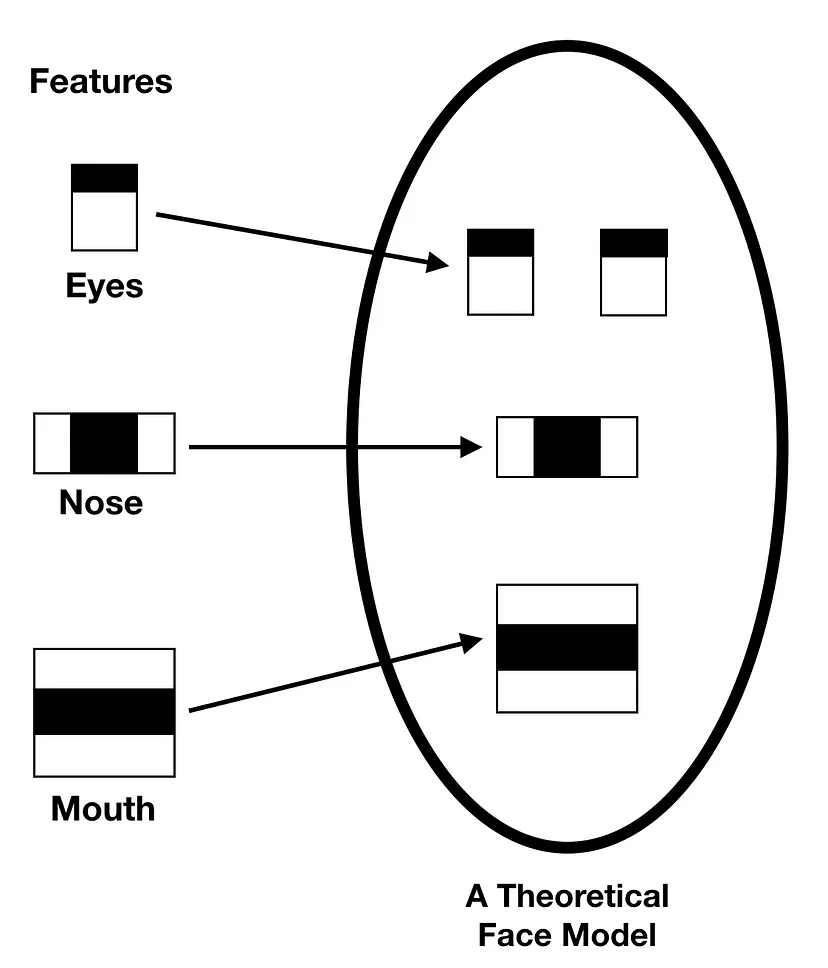


Fig 3 Face Model for Haar cascade

Using these features and the computations of pixels, the algorithm identifies more than 100000 data points.we can see in the Fig 3 that Haar Feature like Line feature, edge feature, etc are used to detect face region. Eyes, nose and Mouth are present in the same square than that square will be our face location.The training data used in this project is an XML File called: haarcascade\_frontalface\_default.xml.

Overall, the Haar cascade algorithm is a powerful and efficient technique for object detection that has been widely used in computer vision applications.

We are going to use the detectMultiscale module from OpenCV.what this does is create a rectangle with coordinates(x,y,w,h) around the face detected in image. This contains code parameters that are the most important to consider.

Workflow of face detection:-

1. Load the Haar Cascade Frontal Face Algorithm.
2. Initialize the camera
3. Read frames from the camera.
4. Convert color images into grayscale.
5. Get the face coordinates.
6. Draw a rectangle and put the appropriate message.
7. Display the output.
8. **Facial Feature Extraction:-**

Identifying faces in photos or live videos is very cool, but this isn’t enough information to create powerful applications, we need more information about the person’s face, like position, whether the mouth is opened or closed, whether eyes are opened, closed, looking up and etc. So for this purpose we need to facial feature to get the more information about the face.Facial feature extraction is the process of detecting and extracting various features of a human face, such as eyes, nose, mouth, and eyebrows, among others.

The dlib library is a powerful tool for performing facial feature extraction in python. It’s a landmark’s facial detector with per-trained models, the dlib is used to estimate the location of 68 coordinates (x,y) that map the facial point on a person’s face like image(Fig 4) below.These points are identified from the pre-trained model where IBUG300-W dataset was used.

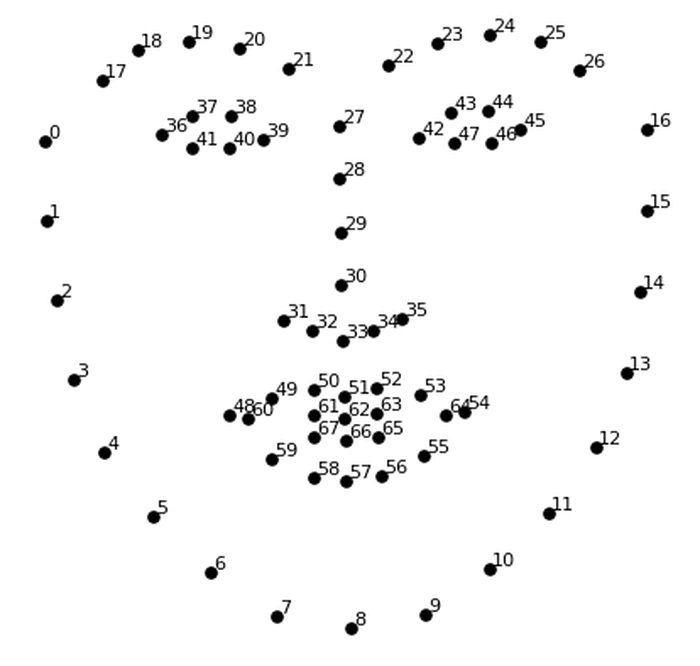


Fig 4: Facial Feature points

* The Locations of the Facial parts are as follows:

1. The Jawline is accessed with points [0,16]
2. The Right eyebrow is accessed with points [17,21]
3. The left eyebrow is accessed with points [22,26]
4. The nose is accessed with points [27,35]
5. The Right eye is accessed with point [36,41]
6. The left eye is accessed with points [42,47]
7. The Mouth is accessed with points [48, 67]

Once a face is detected , the dlib library uses a per-trained facial landmark predictor to extract the facial features. The landmark predictor is a machine learning model that has been trained on a large dataset of labeled facial images.it takes as input an image patch around a detected face and outputs the coordinates of various facial landmarks, such as the corners of the eyes, nose, mouth, and eyebrows, among others as shown in figure 5.

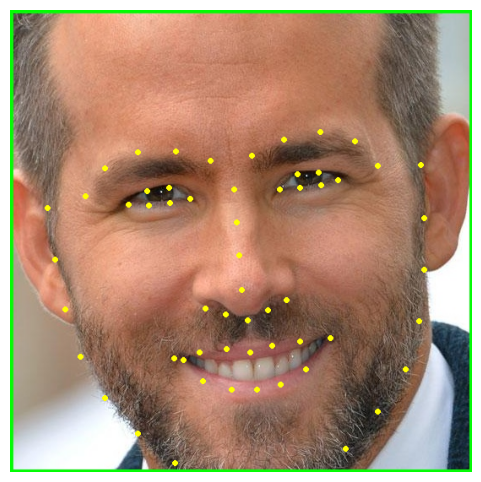
 

Fig 5 Landmark Feature Points

The landmark predictor uses a technique called shape prediction, which involves fitting a set of predefined shapes to the input image patch.the predefined shapes are generated by clustering a large number of facial landmark coordinates from the training dataset. The landmark predictor then uses a regression model to adjust the position of the predefined shape to fit the input image patch. The resulting shape is used to extract the facial features.

The dlib library provides a convenient interface for performing facial feature extraction in python. The library can be used to detect and extract their facial features in real-time video streams or static images. The extracted features can be used for various tasks, such as face recognition, emotion detection and facial expression analysis.

**Literature Review**

1. **Facial Emotion Recognition Using Computer Vision - Jonathan, Andreas Pangestu Lim, Gede Putra Kusuma and Amalia Zahra**
2. **Facial Emotion Detection And Recognition - Amit Pandey, Aman Gupta, Radhey Shyam**

**Reference:-**

1. Jonathan, Andreas Pangestu Lim, Gede Putra Kusuma and Amalia Zahra, “Facial Emotion Recognition Using Computer Vision. The 1st 2018 INAPR International Conference, 7 Sep 2018, Jakarta, Indonesia.
2. Amit Pandey, Aman Gupta and Radhey Shyam, “FACIAL EMOTION DETECTION AND RECOGNITON”, International Journal of Engineering Applied Sciences and Technology, Vol2, Issue 1, ISSN No. 2455-2143, Pages 176-179, 2022.