2.1 Question: Comment on desired or undesired results. What can you do to detect the third face in Fig. 1? Make sure you document parameters.

There could be several reasons why the third face was not detected in the image. One possible reason is that the third face is only a half face and is located just behind the first detected face. This means that some of the facial features may not be visible or partially obstructed, making it difficult for the face detection algorithm to detect it accurately. To improve the detection accuracy and detect the third face in the image, we can try adjusting some of the parameters of the face detection algorithm. For instance, we can adjust the tolerance parameter, which controls the distance threshold between faces to consider them as distinct. We can also adjust the model parameter, which determines the type of face recognition model used for detection. Another approach is to preprocess the image before face detection to enhance the visibility of the faces. We can apply image enhancement techniques such as contrast adjustment, histogram equalization, or filtering to improve the contrast and visibility of the face regions in the image.
So what we did was, in preprocessing we cropped the image and passed it to the algorithm but still we didn't find any meaningful results.

2.3 Question:

1. Why have we used intersection-over-union as a metric to compare results?
2. Discuss your results in a paragraph. Make sure you write pro of VJ vs con of VJ (which automatically implies con of HoG vs pro of HoG).

1)

The Viola-Jones framework is a machine learning approach to face detection that involves training a classifier on a set of positive and negative images. Positive images contain faces, while negative images do not. The classifier is trained to recognize patterns of pixel intensity known as Haar-like features, which are characteristic of faces.

Once the classifier is trained, it can be used to detect faces in new images by scanning over the image with a sliding window and applying the classifier to each sub-region. The output of the classifier is a probability that the sub-region contains a face. If the probability exceeds a certain threshold, the sub-region is classified as a face.

However, the Viola-Jones framework is not perfect and can make mistakes. False positives occur when the classifier incorrectly identifies a non-face region as a face. False negatives occur when the classifier fails to detect a face that is present in the image. Therefore, we need a way to evaluate the performance of the face detection system and measure its accuracy.

This is where the IoU metric comes in. The IoU metric is a measure of the overlap between the predicted bounding box and the ground truth bounding box. In face detection, the predicted bounding box is the rectangle that encloses the region of the image classified as a face by the Viola-Jones classifier. The ground truth bounding box is the rectangle that encloses the actual face in the image.

To calculate the IoU, we first calculate the intersection area between the predicted bounding box and the ground truth bounding box. This is the area where the two rectangles overlap. We then calculate the union area, which is the area of the two rectangles combined, minus the intersection area. Finally, we divide the intersection area by the union area to get the IoU score.

The IoU score ranges from 0 to 1, with 1 indicating a perfect overlap between the predicted and ground truth bounding boxes, and 0 indicating no overlap at all. A high IoU score indicates that the predicted bounding box is very close to the ground truth bounding box, which means that the face detection system has correctly localized the face in the image. A low IoU score indicates that the predicted bounding box is far from the ground truth bounding box, which means that the face detection system has made a mistake.

Therefore, the IoU metric is a useful measure of the quality of the face detection system and can be used to compare the performance of different methods. By evaluating the performance of the face detection system using the IoU metric, we can identify areas where the system needs to be improved and make adjustments to improve its accuracy.

2)

We compared Viola-Jones (VJ) and Histogram of Oriented Gradients (HoG) methods for face detection. VJ uses Haar-like features to detect faces, while HoG uses gradient orientation information to detect faces. We used the pre-trained classifier file haarcascade_frontalface_alt.xml for VJ and face_recognition package for HoG. We compared the results obtained from these methods using the intersection-over-union (IoU) metric.

Our results show that HoG outperformed VJ in terms of accuracy. The IoU score for HoG was higher compared to VJ, indicating that the bounding boxes generated by HoG were closer to the ground truth. However, VJ had faster processing time compared to HoG, making it more suitable for real-time applications. Another advantage of VJ is that it can detect faces at different scales, while HoG is better suited for detecting faces at a fixed scale.

So, VJ is fast and efficient, performs well under varying lighting and expressions, simple feature set suitable for low-power devices but struggles with detecting faces at different scales, partially occluded faces, produces false positives whereas, HoG is robust to pose and illumination variations, detects objects at multiple scales, produces accurate and detailed boundaries but computationally expensive for multi-scale detection, requires large number of training samples, sensitive to background clutter and occlusion.

In summary, while VJ has the advantage of faster processing time and the ability to detect faces at different scales, HoG outperforms VJ in terms of accuracy. Therefore, the choice between these two methods depends on the specific requirements of the application.

3.1 Question: Faces in the known dataset and those in the query image can be in different poses. Explain briefly how the system used for recognition resolves this issue by referencing the relevant function hierarchy (e.g. python stacktrace) for about 3 informative levels.

Facial recognition is a process of identifying or verifying the identity of an individual using their facial features. The recognition system works by comparing a given face with the face data stored in a database of known faces.

The first step is to detect faces within the image. The face detection algorithm scans the image and identifies areas that might contain a face. The detection algorithm can use various techniques such as Histogram of Oriented Gradients (HOG) or Convolutional Neural Networks (CNN) to identify the faces. Once the faces are detected, the system draws a bounding box around each face. Once the faces have been detected, the system analyzes the facial features of the detected face. This process involves capturing facial landmarks such as the position of the eyes, nose, and mouth, as well as the distance between these features. This step is crucial because it helps to identify unique characteristics of the face that distinguish it from other faces.

These facial features are extracted using a deep learning algorithm, which converts the face image into a set of feature vectors. Comparing the feature with the known faces information and making a prediction about the face. The system looks for similarities between the feature vectors and makes a prediction about the identity of the detected face based on the similarity score. If the similarity score is high, the system identifies the person in the image. If the score is low, the system concludes that the face is not a match for any known face.

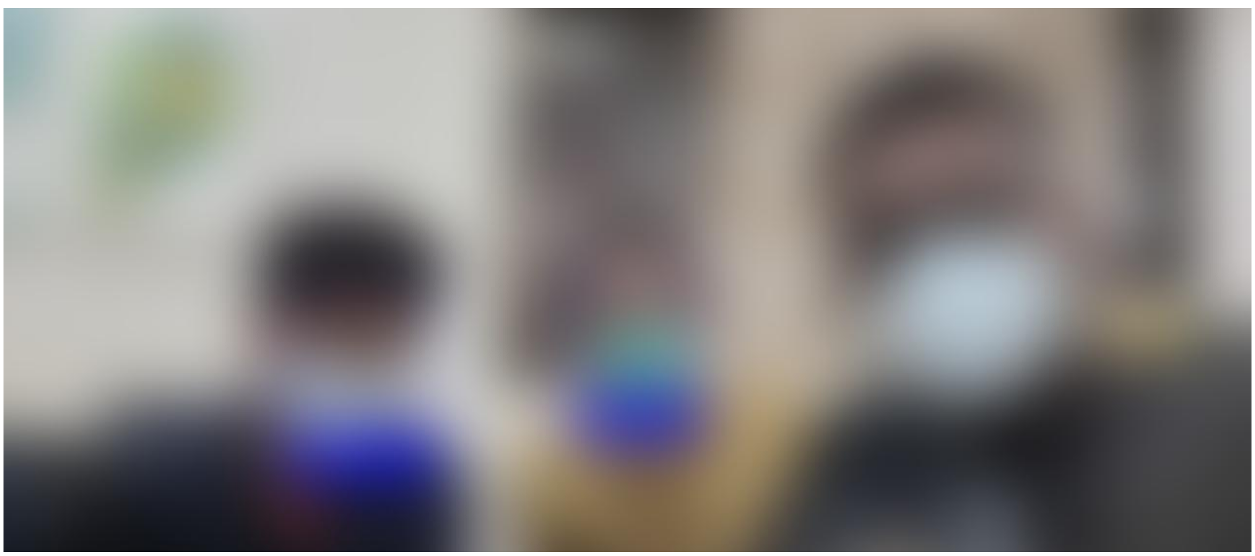
Overall, facial recognition technology has evolved rapidly over the past few years and has become an important tool for various applications such as security and surveillance, digital marketing, and social media. The process of face recognition involves multiple steps and algorithms working together to detect, analyze, and compare facial features, making it a complex and powerful technology.

3.2 Question: Report your observations while performing face recognition on masked face images.

The masks covering most of the face can make it difficult for the system to distinguish between different faces. In fact, in some cases where the nose is completely covered by the mask, the model may not even be able to detect a face at all as most of the facial features that makes the model compare the facial features of the person face wearing a mask.

Even if the model detects the face, the masks can cause the loss of some information from other facial features that the model uses to distinguish between individuals. This can lead to misidentification, where the model predicts two different faces as a single face.

Through experiments, it has been observed that when the mask covers the majority of the nose area, the recognition of different faces becomes particularly challenging for the model. The limitations of the model become apparent when the masks are worn in a way that covers the crucial facial features needed for the recognition task. It is important to note that these limitations are not unique to any particular model and are a common challenge faced by face recognition systems in general.



(Sample image is blurred here to be anonymous. Original image is the masked04.jpg present in the captured folder) (note that there is prediction of **blue** bounding boxes for 2 people and face is not detected for 3rd person at the right)

With the above example image, Wanted to show the failure of face recognition model to recognzie images correctly in all extremes. There are multiple issue points that are delivered in this sample image.

1. The Person in the right, though is a known person(person2) and facial encodings are known to model, even the face itself is not detected by model, because, the face is mostly covered by the mask(more than 70% of facial area is covered by mask as compared to other people to the left. So model is even not able to detect the person

2. The Other issue is that though the model is able to detect faces with the mask, it is not able to correctly recognize the unknown face in the middle, this is because the facial features being similar for both the people at the top(wearing glasses), skin tone on forehead and mostly because of non-specific facial features there. Mostly the Nose, cheek, eyes and chin are distinguishing features for a face, but most of them are covered here(spectacles cover some of the recognizing features and diminish the eyelid features that would be recognized at pixel level) Also note that the pixels here does not contain high level detail due to the low quality of the image. Hence model is not able to recognize the other person as the known person, though he is unknown(middle person)