ANSWERS

2.1 Many computer vision techniques can be used to detect third partially visible face:

1. Haar Cascades: This is a machine learning-based approach that uses Haar-like features to detect faces. Haar Cascades are trained using positive and negative samples of faces to learn the features that distinguish a face from non-face regions in an image. You can train a Haar Cascade classifier using a dataset of partially visible faces to detect the third half face in Fig. 1.
2. Deep Learning-based approaches: Deep learning-based approaches, such as Convolutional Neural Networks (CNNs), have shown excellent performance in face detection. You can train a CNN-based model using a dataset of partially visible faces to detect the third half face in Fig. 1.
3. Template Matching: Template matching is a simple technique that involves comparing a template image of a face with the input image. In this case, you can create a template image of a partially visible face similar to the third face in Fig. 1 and use it to detect the face in the input image.
4. Feature-based approaches: Feature-based approaches involve detecting specific facial features, such as eyes, nose, and mouth, and using them to detect faces. In the case of partially visible faces, you can use the visible features, such as the eyes or mouth, to detect the face.

The parameters that are important for face detection algorithm are:

5. Scale: The scale parameter determines the size of the image regions that are analyzed for the presence of a face. It is important to choose an appropriate scale to ensure that the algorithm can detect faces of different sizes.
6. Threshold: The threshold parameter determines the minimum score or confidence level required for a region to be classified as a face. A higher threshold can reduce false positives, but it can also increase false negatives.
7. Classifier: The classifier parameter determines the algorithm used to classify image regions as faces or non-faces. This can be a machine learning-based approach such as Haar Cascades or a deep learning-based approach such as CNNs.
8. Training data: The training data parameter refers to the dataset used to train the face detection algorithm. The training data should be diverse and representative of the types of faces that the algorithm is expected to detect.
9. Image pre-processing: Pre-processing techniques such as image normalization, contrast enhancement, or noise reduction can improve the accuracy of face detection algorithms.

2.2 Intersection-over-Union (IoU) is a common metric for evaluating object detection algorithms because it provides a simple and intuitive measure of the accuracy of the algorithm's detections. IoU is easy to understand and calculate, and it can be used to compare the performance of different algorithms on the same dataset.

IoU measures the overlap between the predicted bounding box and the ground truth bounding box for an object. The IoU score ranges from 0 to 1, where a score of 1 indicates a perfect overlap between the predicted and ground truth bounding boxes, and a score of 0 indicates no overlap. By comparing the IoU scores of different algorithms, we can determine which algorithm is better at localizing objects in an image.

Another reason why IoU is a common metric is that it is widely used in benchmark datasets and competitions for object detection, such as the Pascal VOC and COCO datasets. This allows researchers to compare their algorithms against other state-of-the-art methods on a standardized benchmark, making it easier to assess the effectiveness of new algorithms.

Overall, IoU is a widely used metric for evaluating object detection algorithms due to its simplicity, intuitiveness, and standardization across benchmark datasets.

2.3 We have captured the accuracy of Viola Jones and HoG over 11 different images in our datasets.The accuaracy and the number of face detected in each image is given below:

1.) Faces Detected :- 3/3

Intersection Over Union for Viola-Jones :- 0.71

Faces Detected:- 3/3

Intersection Over Union for HoG :- 0.55

2.) Faces Detected:- 4/3

Intersection Over Union for Viola-Jones :- 0.62

Faces Detected:- 3/3

Intersection Over Union for HoG :- 0.69

3.) Faces Detected:- 1/2

Intersection Over Union for Viola-Jones :- 0.4

Faces Detected:- 2/2

Intersection Over Union for HoG :- 0.76

4.) Faces Detected:- 3/2

Intersection Over Union for Viola-Jones :- 0.35

Faces Detected:- 2/2

Intersection Over Union for HoG :- 0.59

5.) Faces Detected:- 1/1

Intersection Over Union for Viola-Jones :- 0.74

Faces Detected:- 1/1

Intersection Over Union for HoG :- 0.83

6.) Faces Detected:- 2/2

Intersection Over Union for Viola-Jones :- 0.53

Faces Detected:- 2/2

Intersection Over Union for HoG :- 0.53

7.) Faces Detected:- 2/2

Intersection Over Union for Viola-Jones :- 0.53

Faces Detected:- 2/2

Intersection Over Union for HoG :- 0.41

8.) Faces Detected:- 2/1

Intersection Over Union for Viola-Jones :- 0.53

Faces Detected:- 1/1

Intersection Over Union for HoG :- 0.53

9.) Faces Detected:- 1/1

Intersection Over Union for Viola-Jones :- 0.65

Faces Detected:- 1/1

Intersection Over Union for HoG :- 0.71

10) Faces Detected:- 2/2

Intersection Over Union for Viola-Jones :- 0.6

Faces Detected:- 2/2

Intersection Over Union for HoG :- 0.56

11.) Faces Detected:- 2/2

Intersection Over Union for Viola-Jones :- 0.57

Faces Detected:- 2/2

Intersection Over Union for HoG :- 0.63

Algorithm	Pros	Cons
VJ	Fast detection speed, low computational requirements	Struggles with detecting complex shapes/textures, may produce false positives in cluttered backgrounds
	Works well with small training datasets	Not as accurate as some other algorithms
	Works well with lower resolution images	
HoG	Able to detect objects with complex shapes/textures, high accuracy	Slower detection speed, may require more computational resources
	Can be effective with small objects or objects with small variations in texture	Can be sensitive to changes in lighting conditions
	Effective at handling occlusion (partial obstruction of object)	Can have difficulty detecting partially visible objects
	Can be used with a variety of classifiers	Requires careful parameter tuning to achieve optimal performance

3.1 The face recognition system uses pose estimation and normalization techniques to deal with faces in different poses. In Python, the dlib library provides a function hierarchy that can be used to implement these techniques. The `get_frontal_face_detector()` function in dlib uses a pre-trained model to detect faces in an image and estimate their pose. This function returns a list of rectangles that indicate the position and size of the detected faces.

To estimate the pose of each face, the `shape_predictor()` function can be used to identify key facial landmarks such as the eyes, nose, and mouth. This function returns a set of (x,y) coordinates for each landmark, which can be used to estimate the orientation and angle of the face. The pose estimation information can then be used to normalize the face images, so that they are all aligned in the same orientation and pose.

The normalization process is important because it allows the recognition algorithm to focus on the features of the face that are invariant to changes in pose, such as the shape of the eyes, nose, and mouth. By aligning the faces in a consistent orientation, the system can more accurately compare the features of the query image to those in the known dataset.

In summary, the face recognition system uses a hierarchy of functions in dlib to detect faces, estimate their pose, and normalize the face images. By doing so, the system can handle faces in different poses and improve the accuracy of the recognition results. The specific functions and techniques used within the hierarchy include **`get_frontal_face_detector()`** and **`shape_predictor()`**, which detect facial landmarks and estimate pose, and normalization techniques that align the face images in a consistent orientation. The overall impact of pose estimation and normalization on the accuracy of the face recognition system is significant, as it allows the system to compare features that are invariant to changes in pose and improve the accuracy of the recognition results.

3.2 The face recognition with masked image was able to detect one image out of three image.