Face Detection With Viola-Jones Object Detection Framework

Davide Vettore 868855

1. Objective:

The objective of this activity is to train a Viola-Jones Object Detection Framework which is able to detect faces. The performances of the model are evaluated on a set of test images.

2. Introduction:

The Viola-Jones Object Detection Framework swiftly and precisely discerns objects within images by integrating multiple methodologies:

1. Some types of **Features Detection**: These features are essentially patterns or characteristics of an object that are computationally efficient to compute. Our framework employs Local Binary Patterns as it has been shown to outperform existing feature sets in the context of face detection[1] while maintaining computational efficiency. LBP, along with Haar-like features, excels in capturing fine-scale textures. Alternatively, Histogram of Oriented Gradient (HOG) is commonly employed for object detection tasks.
2. **Integral image**: This technique allows for the rapid calculation of the sum of pixel values within any rectangular area of an image. This integral image representation significantly reduces the computational cost of evaluating features.
3. **AdaBoost algorithm**: Strating from the assumption that a small number of features are enough to build an effective classifier, AdaBoost algorithm iteratively trains a series of weak classifiers, each focusing on different features, and combines them into a strong classifier that can efficiently classify objects.
4. **Cascade classifier**: It consists of a series of stages, each containing a set of weak classifiers trained by the AdaBoost algorithm and the corresponding strong classifier. The cascade works by quickly rejecting negative regions in an image at early stages based on simple features, while dedicating more computational resources to regions that are more likely to contain objects.

The dataset used for training our detector includes both positive and negative classes. Positive instances indicate images containing faces, with a count of 6713 occurrences (three samples can be seen in Figure 1), while negative instances, representing images without faces, are only 274 occurrences. In order to address this substantial class imbalance, we will employ multiple negative class augmentation techniques.

[1] Ojala, T., M. Pietikainen, and T. Maenpaa. 'Multiresolution Gray-scale and Rotation Invariant Texture Classification With Local Binary Patterns.'

Figure 1: Samples from the positive class used for training.

3. Procedure:

As previously mentioned, prior to initiating the training process, it is important to conduct negative class augmentation. This procedure serves the purpose of equilibrating the dataset by creating supplementary synthetic negative instances. To enhance dataset diversity, the following techniques are employed:

- **Vertical and Horizzontal flipping**: Each image undergoes both vertical and horizontal flipping.
- **Rotation**: Ten random rotations (ranging from 0 to 360 degrees) are applied to each image.
- **Cropping**: Six random cropping operations are performed.
- **Scaling**: Six random zoom operations, ranging from 0.75 to 1.25, are executed.
- **Brightness adjustment**: Six random adjustments to brightness levels are made.
- **Gaussian noise**: Six random Gaussian noise factors, ranging from 5 to 20, are incorporated.

After applying negative class augmentation, we now have a total of 10,138 training samples in the negative class.

Figure 2: Negative class augmentation: original image, vertical flipping, random rotation, random zooming, random brightness adjustment and gaussian noise addition.

We can now proceed with the training phase. The trainCascadeObjectDetector method we are utilizing requires several inputs, including the positive and negative training classes, along with the following parameters:

- *NegativeSamplesFactor*: This parameter specifies the ratio of negative samples to employ during training, which has been set to 2.
- *FalseAlarmRate*: It denotes the desired upper threshold for the false alarm rate per negative sample, set at 0.01.
- *TruePositiveRate*: This parameter signifies the desired lower threshold for the true positive rate per positive sample, established at 0.999.
- *FeatureType*: It dictates the type of features utilized during training; in this case, it has been designated as LBP, as previously stated.

To assess the performances of our detector, the detected bounding boxes are compared with the ground truth annotations, the average precision (ap), recall, and precision values are computed. A threshold parameter, set to 0.2, determines the minimum overlap required for a detection to be considered correct.

4. Results:

The precision-recall curve is evaluated using the computed recall and precision values (Figure 3). The title of the plot displays the average precision obtained from the evaluation.

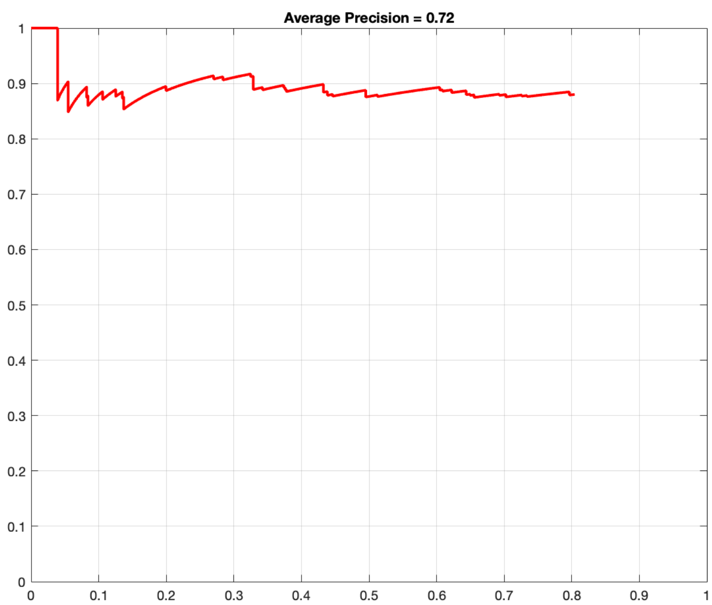

Figure 3: Precision-recall curve.

We can now observe how the detector performs on the testing data. The yellow boxes represents the faces detected by our model, while the red boxes mark the actual faces whose coordinates can be found in the ground truth (*GT.mat*) matlab matrix. The performances of our model were evaluated on those coordinates.

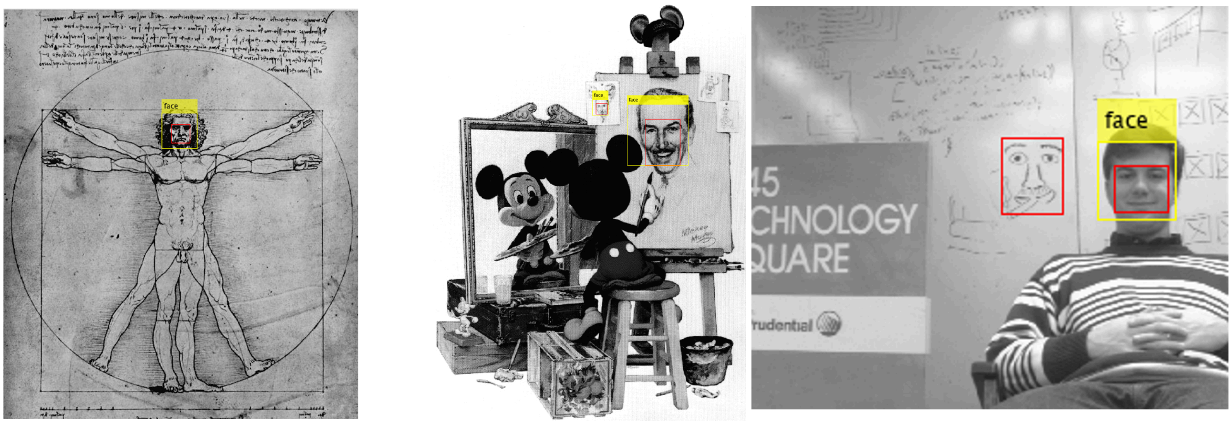

Figure 4: Detector dealing with drawings

The test data contains a variety of examples, including detailed drawings, where the detector accurately identifies facial features. However, it encounters challenges with poorly rendered drawings and stylized faces.

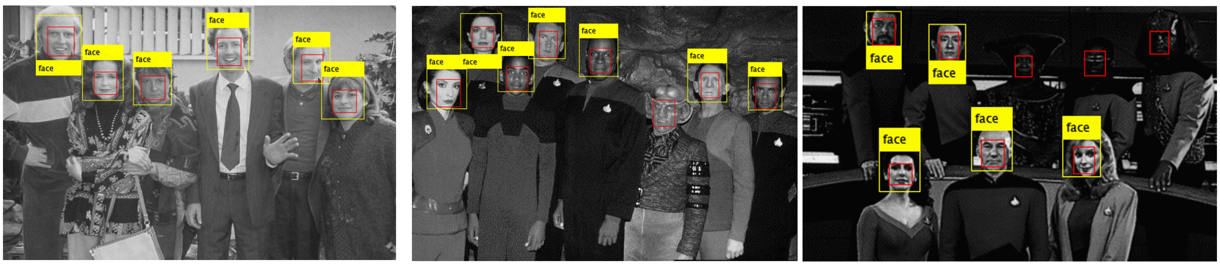

Figure 5: Detector dealing with group of people

The detector demonstrates proficiency in detecting groups of people but encounters difficulties in scenarios with insufficient lighting and distorted facial features. Notably, it exhibits strong performance in detecting both profile and slightly angled faces, a notable improvement over previous models that struggled with the latter scenario.

Figure 6: Other notable examples

As a final observation, the model demonstrates the ability to detect cartooned faces, as well as a picture depicting a girl with closed eyes and a slightly angled face. Notably, when examining the game card picture, it is essential to observe how the detector correctly marks the orientation of the face but fails to detect the same face when mirrored along the horizontal axis.

5. Conclusions:

The Viola-Jones Object Detection Framework, incorporating Local Binary Patterns for feature detection, showcases promising capabilities in accurately detecting faces across various scenarios. Our approach to negative class augmentation effectively addresses the substantial class imbalance present in the training dataset, contributing to improved model performance.

The detector exhibits impressive performance in detecting various facial orientations, including profile and slightly angled faces, representing a significant advancement over previously trained models.

However, the detector exhibits limitations in accurately detecting mirrored faces, as observed in specific scenarios, suggesting that performing an augmentation also in the positive training class may address the problem and increase the detecting power.