Project 3 - Implementing Viola Jones' Face Detection Algorithm

Name **Varsha Ganesh** UB ID **50288433** Email **vganesh2@buffalo.edu**

1.Objective

The main objectives of the Project include:

- To read and understand the given research paper written by Paul Viola and Michael Jones.
- To implement the Viola Jones' face detection Algorithm to distinguish the faces from Non-faces in the given test data set.
- To generate a **result.json** to be compared with the ground-truth.json to compute the accuracy of the implemented model of Viola Jones Face Detection algorithm.

2. Problem Definition

Viola Jones' Face detection algorithm is essentially an object detection algorithm that has competitive object detection rates in real-time. It was proposed by Paul Viola and Michael Jones in the year 2001. It is primarily used to detect faces although it can be trained to identify a lot of other objects as well.

Viola Jones Algorithm has three Main Features, which include:

- Computing the Integral image
- Constructing a classifier by selecting a small number of important features using AdaBoost
- combining successively more complex classifiers in a cascade structure which dramatically increases the speed of the detector by focusing attention on promising regions of the image.

Haar basis functions are used in object detection using a sum of pixels in a single rectangular area. Since Viola Jones algorithms use more than one rectangular area to detect objects, they are more complex and accurate. Also, since Viola Jones makes use of integral images to make these computations, it is much faster.

3. Implementation

3.1 Integral Images

The integral image value at a point (x, y) is the sum of all pixels above and left of the current pixel. An integral image is an image representation that evaluates the rectangular features in constant time. Because each features rectangular area is always adjacent to at least one other rectangle, it follows that any two-rectangle feature can be computed in constant array references. The integral image, s(x,y) can be computed using the following formula, provided the indexes are valid,
i.e, x-1 >= 0 and y-1 >=0.

$$s(x,y) = i(x,y) + s(x-1,y) + s(x,y-1) - s(x-1,y-1)$$

3.2 Constructing a classifier using AdaBoost

The speed with which features may be evaluated does not adequately compensate for their number and it would be prohibitively expensive to evaluate them all when testing an image. Hence Adaboost is used to select the best features and to train classifiers that use them. The

$$h(\mathbf{x}) = \operatorname{sgn}\left(\sum_{j=1}^{M} \alpha_j h_j(\mathbf{x})\right)$$

algorithm constructs a “strong” classifier as a linear combination of weighted “weak” classifiers. Each weak classifier is a feature based threshold function.

3.3 Combining successively more complex classifiers in a cascade structure

The cascade architecture is used to achieve better detection rates. In cascading, each stage consists of a layer of classifiers. The job of each layer is to determine whether a given sub-window is definitely not a face or may be a face. A given sub-window is immediately discarded as not a face if it fails in any of the stages. Due to lack of time, I’ve not implemented cascade architecture in this project. I’ve only used **50 adaboost classifiers in this implementation.**

3.4 Non Maximal Suppression

As we scale the features to match the Image and compare with different sizes, Multiple regions of a face may be detected by the classifier. Inorder to make sure the same face is not produced at several instances in the result, we have to suppress the result using Non Maximal Suppression Algorithm,as below.

4. Training

Dataset for Positive samples: FDDB + Random Data
Dataset for Positive samples: Random Data
No of Positive Samples = 800
No of Negative Samples = 800
No of Classifiers chosen = 15
Training Time : 8 hrs

5. Testing

The model performed exceptionally well on Full frontal faces. However, it was not so good at predicting images with multiple faces, especially when the faces are turned at an angle. Like the above image.

6. Inferences

Due to lack of time, I was not able to implement the Cascade classifier. Also, trained the model only on 800 positive + 800 negative images. I'm confident this would perform much better if I had implemented cascade classifier and trained the model on larger data set.

7. References

- https://www.cs.cmu.edu/~efros/courses/LBMV07/Papers/viola-cvpr-01.pdf
- https://medium.com/datadriveninvestor/understanding-and-implementing-the-viola-jones-image-classification-algorithm-85621f7fe20
- https://www.pyimagesearch.com/2014/11/17/non-maximum-suppression-object-detection-python/
- https://github.com/btuan/ViolaJones/blob/master/violajones.py