FACE DETECTION

~ using VIOLA JONES face detection Algorithm

CSE 573: Computer Vision and Image Processing
Programming Assignment 3

Participant 1: Aditi Ghatge
UBID: 50338244
Participant 2: Shreya Pampattiwar
UBID: 50336817

==========

Table of Contents

Abstract

Face detection, also called as facial detection is an artificial intelligence (AI) based computer technology used to find and identify human faces in digital images.

Face detection technology can be applied to various fields -- including security, biometrics, law enforcement, entertainment and personal safety -- to provide surveillance and tracking of people in real time.
Only recently have our smartphones been able to use a human face as a password to unlock the device. Just like fingerprints, faces are unique with millions of tiny features that differentiate one from the other. It may not always be obvious to us humans, but machines synthesize and evaluate every small piece of data, leading to more objective accuracy.

Embedded in our devices, facial detection can be used in many ways.

Goal

To build and implement the Viola-Jones Algorithm for face detection in the wild.

The Viola-Jones Algorithm:

Viola-Jones is a frontal face detection algorithm which has an ability to detect faces extremely rapidly. Unlike the other algorithms which use auxiliary information such as image differences in video sequences or pixel color in color images to achieve high frame rates, Viola-Jones only uses information present in a single gray-scale image.

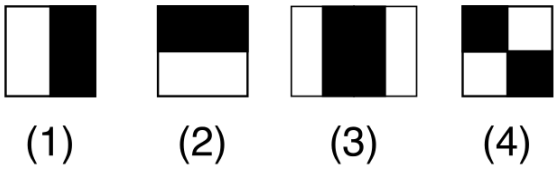
The algorithm has four stages:

1. Haar Feature Selection
2. Creating an Integral Image
3. Adaboost Training
4. Cascading Classifiers

Let us look at them one by one and also understand the implementation of these in our project.

Haar Like Features

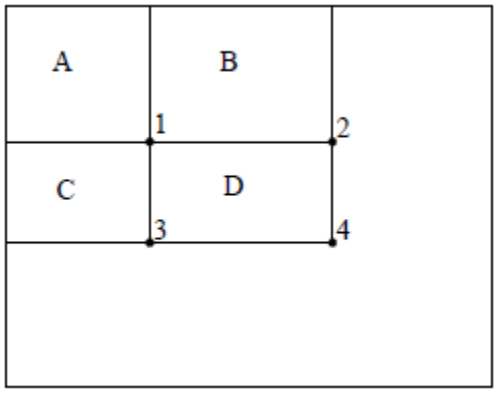
All human faces share some similar properties. The regularities like the eye region being darker than the upper-cheeks and the nose bridge region being brighter than the eyes can be matched using Haar Like Features. The motivation for using features rather than pixels is that features can act to encode ad-hoc domain knowledge that is difficult to learn using a finite quantity of training data. Also feature based system operates much faster than the pixel-based system.



Example of Haar Features: Square shaped kernels. Source

Integral Image

Rectangle features can be computed very rapidly using an intermediate representation for the image which we call the integral image. The integral image at location x,y contains the sum of the pixels above and to the left of x,y inclusive:



$$s(x,y) = s(x,y-1) + i(x,y) \quad (1)$$
$$ii(x,y) = ii(x-1,y) + s(x,y) \quad (2)$$

Adaboost Training

The training process uses AdaBoost to select a subset of features and construct the classifier. A large set of images, with size corresponding to the size of the detection window, is prepared. Each image has index *I*, *I* = 1…*L*. For each image, a corresponding value y_I, is established. y_I=1 for faces and y_I=0 for nonfaces.

Initialize weights $w_{1,I} = \frac{1}{2P_-}, \frac{1}{2P_+}$, for y_I=0,1 respectively where P_- and P_+ are the number of nonfaces and faces in the image set.

The algorithm is executed for an arbitrary number of rounds, *I*.

For *i* = 1…*I*

1. Normalize the weights as follows so that $w_{i,I}$ is a probability distribution:

$$\frac{w_{i,I}}{\sum_{j=1}^{n} w_{i,j}} \rightarrow w_{i,I}$$

2. For each feature *j,* train a classifier h_j which is restricted to using a single feature. The classifier's error rate is evaluated with respect to $w_{i,I}$:

$$\varepsilon_j = \sum_{I=0}^{L-1} w_{i,I} \left| h_j(x_I) - y_I \right|$$

3. Choose the classifier, h_i, with lowest error ε_i.

Update the weights:

$$w_{i+1,I} = w_{i,I} \beta_i^{1-\varepsilon_i}$$

$$\beta_i = \frac{\varepsilon_i}{1-\varepsilon_i}$$ The final classifier is:

$$h(x) = \begin{cases} 1, & \sum_{i=0}^{I-1} \alpha_i h_i(x_I) \geq \frac{1}{2} \sum_{i=0}^{I-1} \alpha_i \\ 0, & \text{otherwise} \end{cases}$$

$$\alpha_i = \log \frac{1}{\beta_i}$$

Cascading Classifiers

The complete face detection cascade has 38 stages with over 6000 features. Nevertheless the cascade structure results in fast average detection times. A cascade of gradually more complex classifiers achieves even better detection rates.

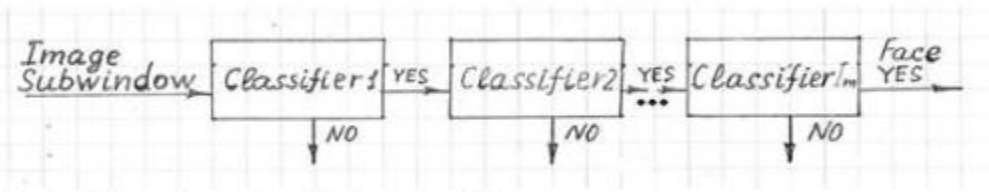


Fig.3 Object detection Viola-Jones filter

Technical Implementation

We have implemented each of the modules listed above in the Viola-Jones algorithm.

integralOf()

This method calculates the sum of pixels under a rectangle using the integral images method discussed above. The time complexity of this method is O(1), which means that it will give you the results you need in a single traversal over an image which makes it the ideal method for detecting our Haar like features.

haarFeaturesOf()

This method is implemented to detect haar features in the image which will tell us if there is a human face in the wild.

Here, we have implemented this by taking care of 4 cases as shown in the diagram above:

i left:light, right:dark
ii top:dark, bottom:light
iii center:dark, left:light, right:light
iv topLeft:dark, topRight:light, bottomLeft:light, bottomRight:dark

The light and dark rectangles traverse over the image. In each of the cases, we then calculate the sum of pixels in the black area minus the sum of pixels in the white area. To calculate the sum of pixels, we will use the integralOf method.

Now, for a uniform area or surface, this sum will be very close to zero. We discard such areas. When there is feature of a face is detected, the sum will be a very large number indicating that the areas in white and black rectangles are very different. Thus, we use such areas as features.

applyFeaturesToData()

This method applies the features extracted by the haarFeaturesOf() method to the data image. The result is that we will get a detected face from the image.

Results:

We tested the classifiers with the training data as provided and the results were as expected.

References:

[1] https://searchenterpriseai.techtarget.com/definition/face-detection
[2] https://towardsdatascience.com/the-intuition-behind-facial-detection-the-viola-jones-algorithm-29d9106b6999
[3] https://en.wikipedia.org/wiki/Viola%E2%80%93Jones_object_detection_framework
[4] https://medium.com/datadriveninvestor/understanding-and-implementing-the-viola-jones-image-classification-algorithm-85621f7fe20b
[5] https://www.vocal.com/video/adaboost-training-algorithm-for-viola-jones-object-detection/