Face Detection using HOG

Alejandro Trujillo
Universidad de los Andes
Bogotá, Colombia
af.trujilloa@uniandes.edu.co

Santiago Martínez
Universidad de los Andes
Bogotá, Colombia
s.martinez1@uniandes.edu.co

Abstract

The task of face detection has been explored for decades and its results have been astonishing. In this paper we will train and discuss an SVM-HOG model for face detection, quantitatively and qualitatively. The average precision obtained in our best method was of 0.528, which isn't a good result for today's standards but shows the reader the advantages and limitations of these kind of approaches to detection problems.

1. Introduction

The problem of Detection is a very important task in the field of computer vision. It uses the same concepts of classification, however it's highly unbalanced, as in one image there could be thousand negatives and only a few positive, for which reason the ACA doesn't work well as an evaluation method. The Histogram of oriented gradients (HOG) is an useful tool that helps describe the shape of an object as an histogram, which is useful when using a classifier. As faces mostly present a similar shape, perhaps it's a good idea to use HOG as a descriptor for face detection.

2. Methodology

2.1. Dataset Description

The images use during test and train are from multiple origins.

- Train set, Caltech crop Faces: A collection od cropped human faces with size 36x36 (6713 images)

- Train set, Train-Non Face Scenes: A collection of random images representing landscapes, animals, buildings and others. [2] (275 images in the simplified version and 2891 in the full version)

- Test, 120 images containing different amount of faces and fully annotated.

- Extra test scenes: Non-annotated samples from class

Figure 1. Example random train set images -Negatives

Figure 2. Example random train set images -Positives

In figures 1 and 2 There is random examples of the training images used to train the SVM model.

2.2. HOG

The HOG multi-scale algorithm uses Oriented gradients to describe an image shape. the gradient is calculated densely (in cells evenly distributed on the image) in order to softly include spatial information in the descriptor.The Vl_Hog function from the vl_feat library for Matlab was used in order to compute the HOG transform. This function receives as parameters the size in pixels for each cell, the amount of orientations used when calculating the gradient.

2.3. Algorithm

The algorithm used is based on one developed by James Hays from Brown University which uses hog and a sliding window in order to detect faces in images. Firstly, an SVM was trained using the cropped images of faces and calculating the HOG for each one of them. The same was done for the non-face examples resizing them to 36x36 in order to obtain a descriptor of the same size. After training the SVM, A sliding window algorithm was developed, which uses different lengths based on the image size. These windows are in square and rectangular (oriented in both sides) and include multiple scales. Slices where taken from the images with the size of these windows and resized to 36x36, moving in steps of a determined length, Afterwards, the HOG was calculated from the slice and classified according to the previously trained SVM. then a threshold was established in order to classify a slice as face or not face. The threshold and step were decided through experiments.

2.4. Evaluation Method

As the detection task is a highly unbalanced problem (significantly more negative samples than positive ones in an image), the ACA isn't a proper metric, as it only takes into account the recall of the algorithm. For this reason, a Precision-Recall curve is calculated and the AP (Average Precision) serves as a proper metric.

3. Results

3.1. Experiments

We decided to not change the default parameters for the HOG algorithm and deal with the sliding window instead. Another significant change done was that the non face examples were changed from 275 images to 2000, as the full dataset was downloaded. This considerably changed the performance of the algorithm under the same valued parameters.

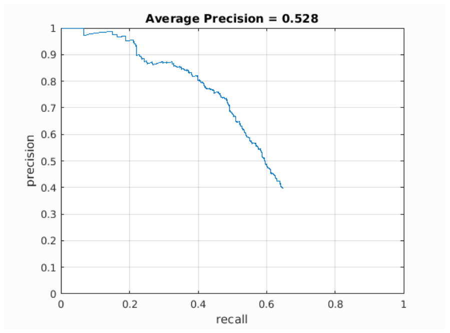

Figure 3. results using 2000 Negative Training images

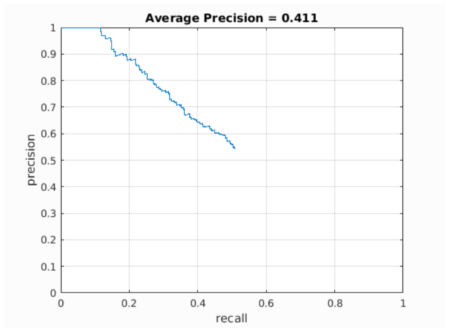

Figure 4. results using only 274 Negative Training images

In figures 3 and 4, one can see the difference in performance when using the same hyper-parameters and a different amount of training images. The results improve considerably given the amount of negative examples given to the machine, for which reason we decided to keep it that way.

The threshold used was augmented a little amount and the average precision went considerably down, for which reason we decided to keep the original hyper-parameters.

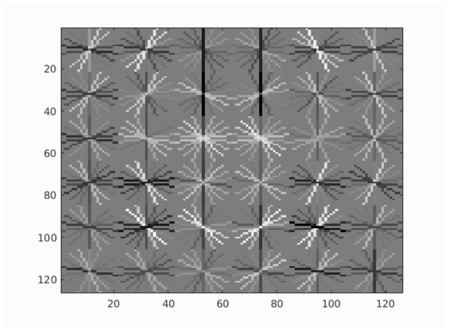

Figure 5. SVM HOG Model

In the figure 5 we find the HOG SVM model obtained after training the SVM. One can clearly see a face-like shape to it, as it was trained to detect this kind of shapes.

3.2. False Positives Analysis

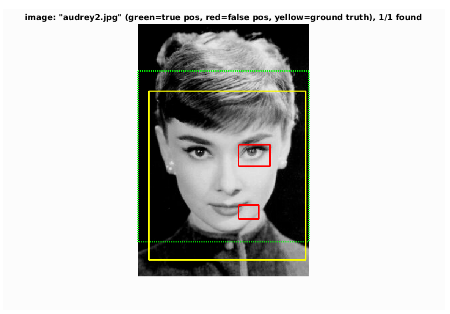
image: "audrey2.jpg" (green=true pos, red=false pos, yellow=ground truth), 1/1 found

Figure 6. Example 1 qualitative results

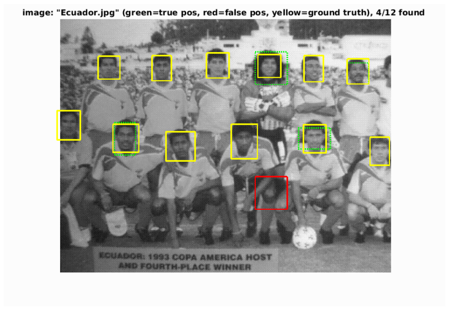
image: "Ecuador.jpg" (green=true pos, red=false pos, yellow=ground truth), 4/12 found
ECUADOR: 1993 COPA AMERICA HOST AND FOURTH-PLACE WINNER

Figure 7. Example 2 qualitative results

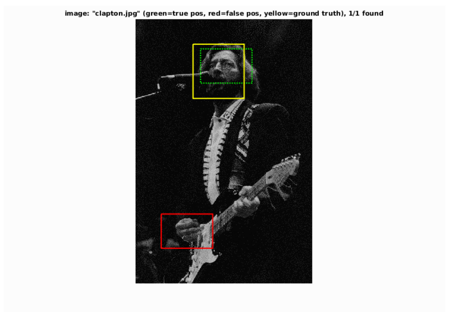
image: "clapton.jpg" (green=true pos, red=false pos, yellow=ground truth), 1/1 found

Figure 8. Example 3 qualitative results

in figures 6, 7 and 8 one can see some of the recurrent patterns in the false positives of the algorithm. One can see that the algorithm confuses faces with eyes, knees and hands. This is due to the similar shape they present as the eye and knee have a circular shape similar to that of a face. A way to correct these kinds of errors would be to perform a hard-negatives mining, adding these sample of false-positives to the non-face category and re-train the SVM.

3.3. False Negatives Analysis

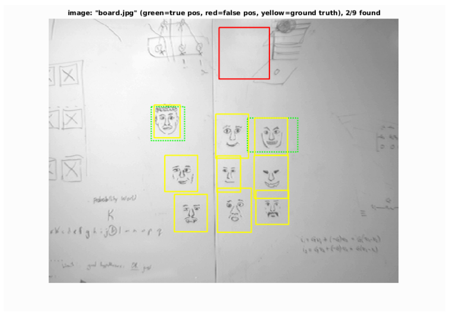
image: "board.jpg" (green=true pos, red=false pos, yellow=ground truth), 2/9 found

Figure 9. Example 4 qualitative results

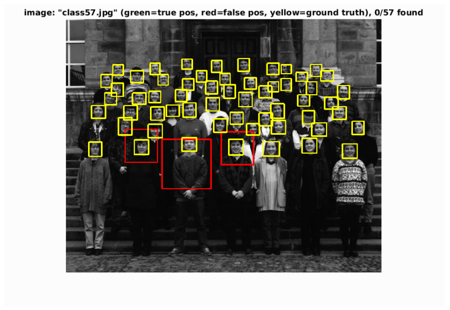
image: "class57.jpg" (green=true pos, red=false pos, yellow=ground truth), 0/57 found

Figure 10. Example 5 qualitative results

in figure 9 one can see the most recurring false negative found, which is the lack of a contour defining the face. This is understandable as we are using shape and drawings that doesn't have a strong contour won't be classified as face by the algorithm. Another significant problem is the size of the face, as can be seen in figure 10, where even with the detected faces, the scale is different enough for the evaluation algorithm to consider them false positives. Other less frequent examples include lack of illumination and perspective change.

3.3.1 Viola-Jones Algorithm

To prove the use viola-jones algorithm, we used a libraries corresponding from matlab [[3]], the name and command used was vision.CascadeObjectDetector in this case we dont train the database, because the program is optimized to detect peoples faces. The work from algorithm implement by Matlab it good as in the images (11,13,12,14) located the faces better that HOG, HOG found the faces but also give us false positives. Other variant in this comparison was watch more detections in some images before of to move the threshold trying to better itself

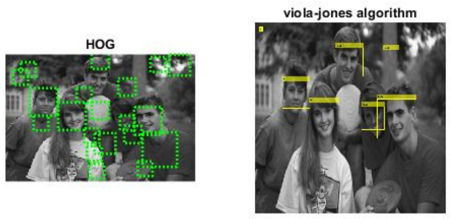

Figure 11. In this probe look that the false positives, it's on many places of image

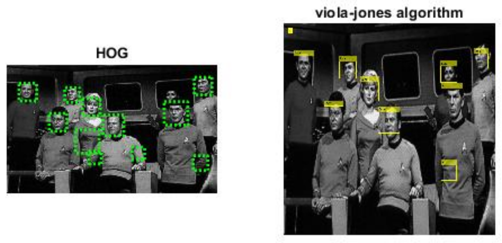

Figure 12. In this image we found all tribulation of star trek

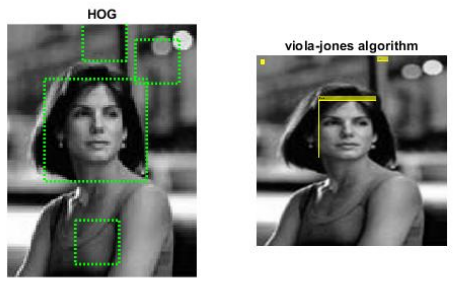

Figure 13. In the image the false positive from our algorithm were evident

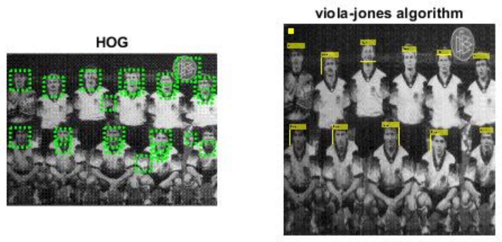

Figure 14. the soccer team were found with sort of false positives

4. Conclusions

The HOG approach for this problem resulted partially successful, one way to improve the algorithm would be to perform a "Hard negatives mining" over the test set, as this would force the algorithm to reject similar shapes. However "Hard negatives mining" would only solve the problem where the algorithm confuses faces with other structures, as for faces the algorithm doesn't detect there is not much to do, as its a limitation of the HOG-SVM classification.

References

[1] *"Man and Woman Shaking Hands Free Stock Photo", Pexels.com, 2019. [Online]. Available: https://www.pexels.com/photo/man-and-woman-shaking-hands-1249158/. [Accessed: 01- Apr- 2019].*

[2] *"Index of /Images/users/antonio/static*$_s un_d atabase/b$", $Labelme.csail.mit.edu$, 2019.$[Online].Available$: $http : //labelme.csail.mit.edu/Images/users/ antonio/static_s un_d atabase/b/.[Accessed$: $01 - Apr - 2019]$.

[3] *Detect objects using the Viola-Jones algorithm - MATLAB - MathWorks Amrica Latina. (2019). La.mathworks.com. Retrieved 1 April 2019, from https://la.mathworks.com/help/vision/ref/vision. cascadeobjectdetector-system-object.html*