HOG Face Detection

Mauricio Neira
Universidad de los Andes
Cra 1 Nº 18A - 12, Bogotá - Colombia
m.neira10@uniandes.edu.co

Daniel Rodriguez
Universidad de los Andes
Cra 1 Nº 18A - 12, Bogotá - Colombia
da.rodriguez1253@uniandes.edu.co

Abstract

In this study we have implemented a multiscale HOG face detection algorithm. We find the multiscale approach of utmost necessity when handling real life data because of the variance in face sizes. We use a simple linear classificator SVM that is able to correctly separate most of our input data with ample margin. We study how increasing large negative training samples enhances the HOG template that learns facial human features. Finally, we characterize the quality of face identification with the Precision-Recall metric when varying the threshold value of our linear classificator.

1. Introduction

Detection of human faces and bodies is a challenging task because of the multitude of poses, illumination, sizes and blackground cluttering. For these specific applications algorithms like DT and SIFT have been developed. In this paper, we implement a multiscale HOG face detection algorithm. The main parameter in the HOG implementation is the size of the cell because it determines the detail with which features are extracted. In the multiscale strategy, the range of scales used is essential and should be decided based upon the size conversion between positive training images and test images.

Even though the algorithm can only distinguish images of faces from images that are not faces, it can be turned into a face detector by creating a sliding window that predicts at each location in the image whether the current window is a face or not.

The algorithm will be evaluated as any detection algorithm whose annotations are bounding boxes. Each prediction bunding box will be compared to the annotation bounding box. If the Jaccard index of the 2 boxes exceeds a particular threshold, it will be considered a true positive. Any prediction bounding box that does not match any annotations will be a false positive and all annotation bounding boxes that do not have a corresponding prediction will be considered false negatives. With these numbers, it is possible to calculate the precision, recall and f-measure. Slight tweaks in the algorithm can then be made to build a precision-recall curve to evaluate the performance of the face-detector at several regimes.

2. Images used

We use the Caltech 10,000 Web Faces Dataset. These faces were randomly selected from the internet by tipping common names in Google. The dataset has 10,524 human faces of various resolutions and in different settings, e.g. portrait images, groups of people, etc.

Figure 1. Example face image from the traning set.

Figure 2. Example images from the Imagenet database. Courtesy of [1].

3. Approach

Overall description of your strategy including any modifications/enhancements you applied to it.

Our strategy is based on two different datasets that serve as positive and negative training. First, we extract HOG features of every image in the positive training examples. Then, we set a specific number of negative features (100000

for experiments shown here) and crop them from the negative training examples. The size of the image cropped is equal to the size of training images to get a HOG descriptor of equal size. Instead of incorporating a larger set of negative training images we increment the number of samples extracted. Finally, we use an Support Vector Machine classifier to linearly separate both sets. We modified the HOG cell size to get a more detailed level of description at the cost of computational time. However, given the simple linear classification the computational cost is actually very small.

3.1. Implementation Details

Our positive training samples are all of size 36×36. This represents some challenges when we try to identify faces of different sizes as discussed below. We use a HOG cell size of 4 to increase the number of HOG descriptors to 2511. We find values below this number to be of less accuracy. We employ a negative training set of 100000 samples, to ease the linear classification. To extract them, we randomly crop patches of the images that have size 36×36, bypassing the need to enhance the negative training set with more images.

4. Experiments and discussion

4.1. Linear separation with SVM

The following figure displays the linear separation of the two positive (green) and negative (red) training datasets, and the the null line. Even though, on the training set we obtained an accuracy of , this set of parameters was not enough to find faces on the test (10%). Consequently, we incremented the size of the negative examples to 100010 and decreased the cell size to 4 to obtain higher precision and detail.

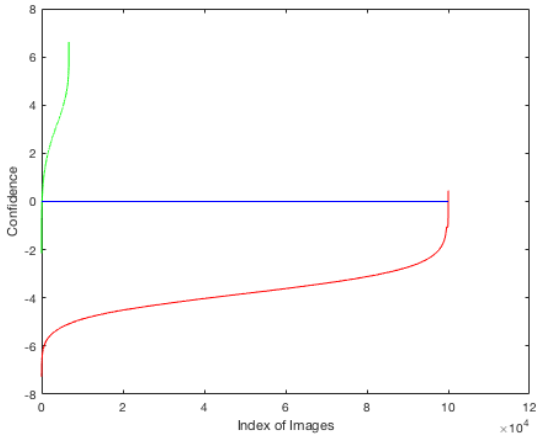

Figure 3. Linear separation between positive and negative training examples.

A very limited number of positive training images has a negative confidence value. This result shows that even with a huge negative training some test images will not be identified. The maximum value of negative examples is less than 0.5 so that in principle, with little cost a high value identification is obtainable. However, some feautures in the test images may resemble facial structures and hence a confidence value closer to 2 could decrease false positives.

4.2. HOG template

The following figure shows HOG templates obtained under the more restrictive conditions described before. (In the Appendix, we also show HOG templates for less restrictive conditions.) Here, we are able to identify the silhouette of eyes, nose and mouth as well as a round face shape. Even though the training set has faces all of shapes and even slightly oriented (but not from a profile angle) the average of this gradients converges to a symmetric face with common proportions.

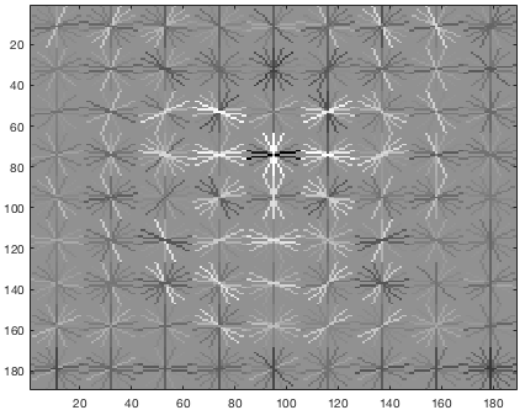

Figure 4. HOG Template with a training set of 100010 negative images and cell size of 4.

4.3. Accuracy as a Function of Scale

Scale is a determinant parameter of the quality of face identificaction. In its original conception, the HOG algorithm identifies faces that are on equal scales that the training images. However, in the current evaluation task the training images are of size 36×36 .

With any given scale, the number of faces that coincide in each set is very low. Here we study the accuracy given by specific scales. The scale of 0.75 is the best suited for the identification task since most faces on the dataset correspond to this scale. Even though this results per se are unsatisfactory, they allow us to explore better the range of scales.

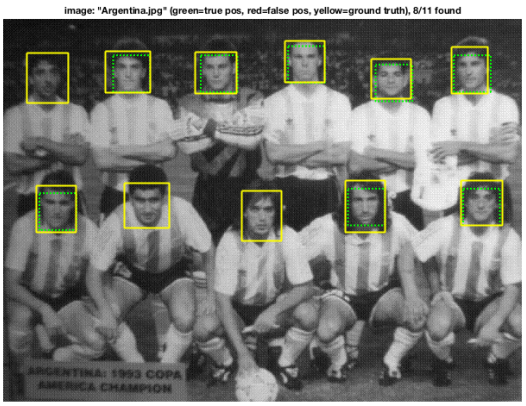

Figure 5. 8 of 11 faces identified in an image when using the 0.75 scale.

Figure 6. A rare instance of a face identified in the 1.0 scale. Few faces in the dataset are as small as our positive training examples.

Scale	Recall
1.0	0.09
0.75	0.20
0.50	0.08
0.25	0.02

Table 1. Recall measures for fixed scales. The values are proportional to the of faces in the dataset with similar size. With a threshold value of 2.0 a precision very close to 1 was obtained for all scales.

4.4. Multiple Scales in One Feature

Given the previous results, we now use a multitude of scales ([1,0.9,0.8,0.75,0.7,0.6,0.5,0.4,0.3,0.2,0.1]) to identify faces of all sizes. The balance between recall and precision is determined by the threshold parameter.

The PR curve displays an extended region of precision

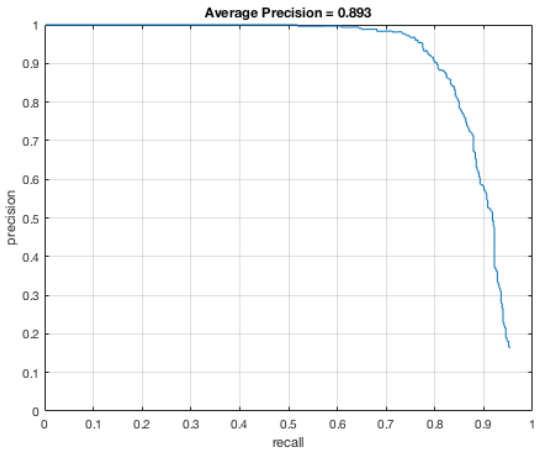

Figure 7. Precision-Recall curve.

close to 1 that rapidly decays in the high recall region. The algorithm has difficulties finding faces with shadows, smiling, of slight orientation and of black people. For example, experiment with the threshold parameter we found that the algorithm is most likely to identify a bright object (like a knee) as a face than the face of a black player.

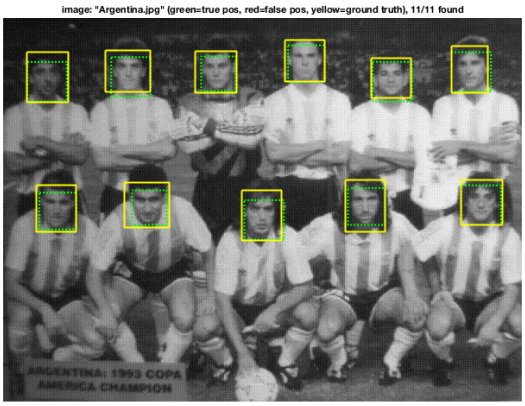

Figure 8. All faces are identified when choosing a threshold value of 1.0.

5. Conclusions

The multiscale HOG face identification is a succesful algorithm that allows a F1 measure as high as 0.90. The algorithm exhibits a bias towards high precision regions that could be suitable for some applications. It faces difficulties when handling faces in conditions such as covered by shadows, smiling, slightly oriented and of black people. This is an indication that strongly suggests that the training set does not contain sufficient quality exaples so as to general-

Figure 9. Qualitative results of the Viola-Jones algorithm on the test set. The green boxes correspond to the annotations and the blue boxes to the predictions.

ize facial detection to the world's population. Enlarging the training dataset is a necessity.

To further improve the performance of the algorithm, it would be worth trying out a convolutional neural network. It has been proven in a plethora of problems in computer vision that neural networks are the best perfo ming. This aproach might prove succesful in thrsiproblem as well.

6. Extra credit

6.1. Viola Jones

The Viola-Jones face detection algorithm [2] was implemented using the openCV library[1]. Figures 9 and 10 are examples of qualitative results on the test dataset. The precision-recall curve obtained with the algorithm can be seen in figure 11. It is clear that there is a significant decrease in performance with respect to the HOG algorithm in Figure 7. Considering that the original viola-jones algorithm was constructed to optimize efficiency over accuracy, this result makes a lot of sense. It is tailored to find differences in light intensity along specific geometric orientations. This decreases computation time significantly but sacrifices overall performance in turn.

6.2. Waldo

Running the HOG algorithm using only Waldo as a positive example did not work. At first, the algorithm didn't even correctly label the Waldo in the training set. We had to replicate the waldo sample image 3000 times for the SVM to classify it correctly.

[1]Details on the implementation can be found directly on their website: `https://docs.opencv.org/3.4.3/d7/d8b/tutorial_py_face_detection.html`

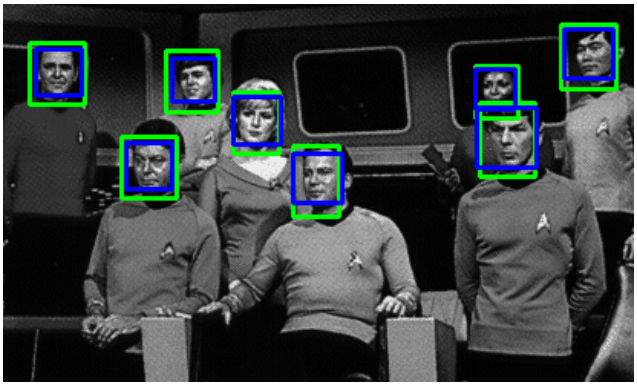

Figure 10. More qualitative results of the Viola-Jones algorithm on the test set. The green boxes correspond to the annotations and the blue boxes to the predictions.

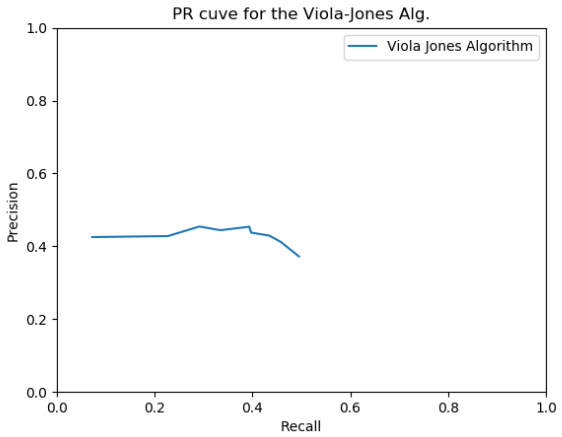

Figure 11. Precision-recall curve for the viola-Jones algorithm.

However, even after tweaking with the sliding window algorithm and exhaustively searching the hyper-parameter space, the image could not be found.

Nevertheless out of curiosity, we listed all of the items by date on the Waldo folder and found that the only image that had a different timestamp was: `Waldo/13--Interview/13_Interview_Interview_On_Location_13_558.png` corresponding to Figure 12.

It would be wise to modify the time stamp of the image to match the entire database so that future students can't easily find the image, cheat and overfitt their algorithms to find it.

References

[1] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009*

Figure 12. Waldo.

IEEE conference on computer vision and pattern recognition, pages 248–255. Ieee, 2009.

[2] P. Viola and M. J. Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004.