Laboratory 9 - Hog Face Detector

Guillaume Jeanneret
Universidad de los Andes
Cra 1 #18a-12, Bogotá, Colombia
g.jeanneret10@uniandes.edu.co

Jorge Madrid
Universidad de los Andes
Cra 1 #18a-12, Bogotá, Colombia
ja.madrid2714@uniandes.edu.co

Abstract

The PHOG face detector is a preliminary approach the face detection task. Using an easier version of the Wider Face dataset, a first approach was done using a basic algorithm. This attempt gave us a result of ***0%*** *within the validation subset.*

1. Introduction

The database [2] Wider Face is a compilation of 32.203 images with 393.703 labeled faces. Each face present a high variability in the pose, scale and occlusion aspect. Also, the dataset is partitioned in training, validation and test subsets, containing each 40%, 10% and 50% respectively of the total images. Furthermore, each subset is composed of 61 event classes. The evaluation metric employed is the same as the PASCAL VOC dataset used: Precision-Recall Curve (PRc) [1].

2. Materials and Methods

2.1. Database

The dataset in which the algorithm was used is the Wider Face dataset [2]. Because of the complexity of the task, an easier version was given. This subset of the original dataset contains 12.242 images of only faces of minimum resolution 80x80 pixels, 7.245 pictures of real life photos (training), 3.226 pictures of validation and 3.604 photos of test. Because of the secrecy of the test annotations, the result is generated in the validation subset.

2.2. Methods

The first method used consisted in training a support vector machine (SVM). The positive data was extracted from the faces given in the database. Re-scaled to a window size of 125x100 each face photo, their HOG was calculated and vectorized. For the negative data, a window was taken randomly and its HOG was produced. A similar number of negatives were produced so the data is balanced.

The second method was more time consuming. Taking a sliding window of 125x100, the hog was calculated and then vectorized. Then, this vector is evaluated in the trained SVM.

The third method was a multi-scale Hog detector. Three different scales were taken: 125x100, 100x80 and 80x80. The training was done similarly to the last method. Each crop was read, re-scaled and vectorized. Three different SVM, each one for a scale, were trained. With this method, because we want to detect anything -we want to have the greatest recall- if the 125x100, 100x80 or 80x80 detector gets a positive on any crop, then it is considered as a detection with a box of the highest resolution.

A way to generate a more general result is to do Negative Mining and re-training the model with the new data acquired and the old one.

2.3. Evaluation

The evaluation of the algorithm was done using the PRc. The *'.m'* file was given, so the result was given automatically.

3. Results

The first method we tried didn't give us any positive result. For the easy, medium and hard evaluation, the area beneath the curve was 0.

For the second method, the results gave us a PRc with values of 7.8 (For the easy one), 3.4 for the medium and 1.6 for the hard one. This values are inconsistent because the maximum value is 1.

Total time: 54 minutes (3268 seconds)

Either we failed, the problem is too difficult, or both.

The effect of the multi-scale Hog: Using only the 125x100 detector: Number on detection can vary from 1200 to 2400 (High recall, low precision).

Using only the 100x80 detector: variation between 4000 and 5200 detections

Using only the 80x80 detector: Variation between 900 and 1200 detections.

Some detection are displayed in figure 9

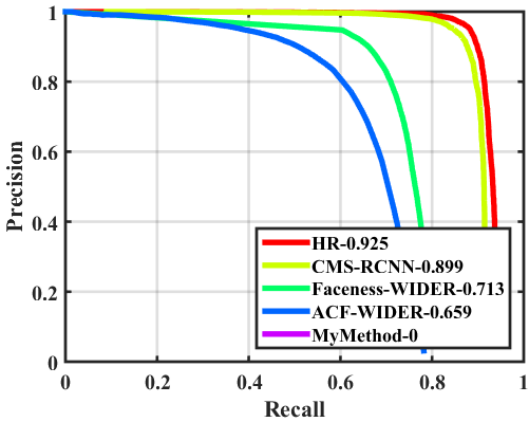

Figure 1: Easy result from the evaluation in the validation test.

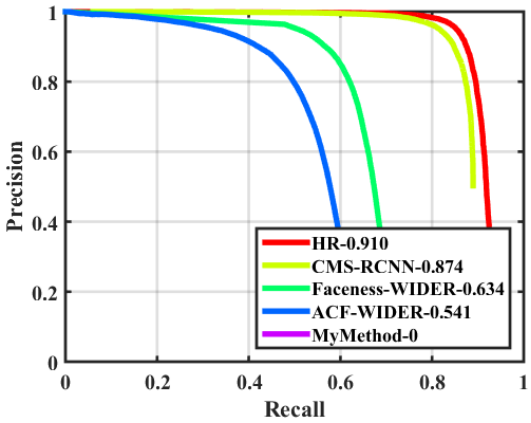

Figure 2: medium result from the evaluation in the validation test.

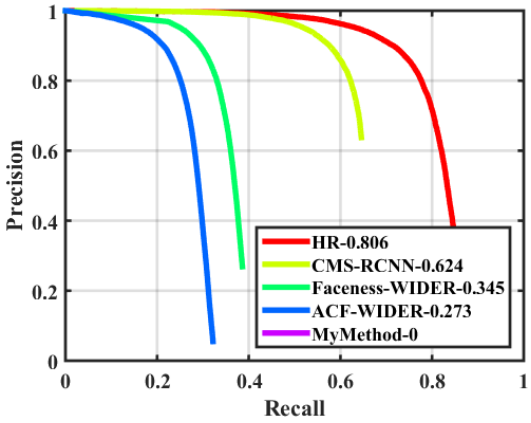

Figure 3: hard result from the evaluation in the validation test.

Figure 4: Wrong detection.

Figure 5: Wrong detection, a texture was detected.

Figure 6: Detection of a face cut in half.

4. Discussion

The problem of classification is challenging when images that are more realistic, heavily cluttered, greater in variety, and which differ in how they were captured, are to be tagged into categories. Only when the problem meets the challenges of real-world imagery, will computers become more useful performing such tasks.

A classification problem can be evaluated using the Jaccard Index as expressed from the equation 1 This expression permits to know how much the ground-truth overlaps the box detected.

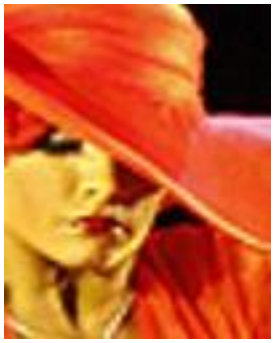

Figure 7: Face detected -Dancing group-.

Figure 8: Face detected -Basketball group-.

$$J(A, B) = \frac{A \cap B}{A \cup B} \quad (1)$$

Another problem that we detected is the SVM we trained. The VL_Feat provides a function that trains a linear SVM. We think that the problem here is that linearity does not provide a special partition that fits the training data.

To get different results, the hyper-parameters of the multi-scale HOG detector can be changed. The first and more important was the scale. As said before, this database contains a wide variety in the occlusion of the faces, scale included. Another hyper-parameter is the number of orientations per cell. Having to many of them can lead to overfitting the model because each vector will describe more precisely each case. Finally, the cell size is the last factor. Big sized cells will get many more orientation with the cost of loosing spatial information.

All the three methods that were implemented use a Histogram of Oriented Gradients as the underlying methods to classify a window as containing a face or not. HOG was enormously succesful when used with the pedestrian-detection dataset, however it was not as succesful with the WiderFace challenge. One immediately wonders why the two results are so different. Pedestrians exhibit very limited variations in pose and orientation, thus their HOGs look alike. Although the SVM trained with the HOG attempts to learn from the variations of the data, it makes underlying assumptions on the structure of a face. While faces are very expressive, HOG does not seem to be expressive enough. It probably yields such modest results because it cannot discriminate windows sufficiently well based on a model of oriented gradients.

5. Conclusion

- The problem of detection is inherently more difficult. Having an enormous number of negatives, the probability of detecting a false positive rises. This explains, at least in principle, the lower current results in this problem if compared to classification.
- Figure 4 is an evidence of how the HOG model can fall short to solve this problem. Although learned from the SVM, the HOG method has strong underlying assumptions about what a face *looks like*. If the model is not expressive enough, the classifier will have trouble distinguishing between images that may result similar to faces for the computer, but that are notoriously different for a human observer, such as a sign with characters. The lack of expressiveness in such models is probably one of the most dominant reasons for the area's shift towards neural networks.

References

[1] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[2] S. Yang, P. Luo, C. C. Loy, and X. Tang. Wider face: A face detection benchmark. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

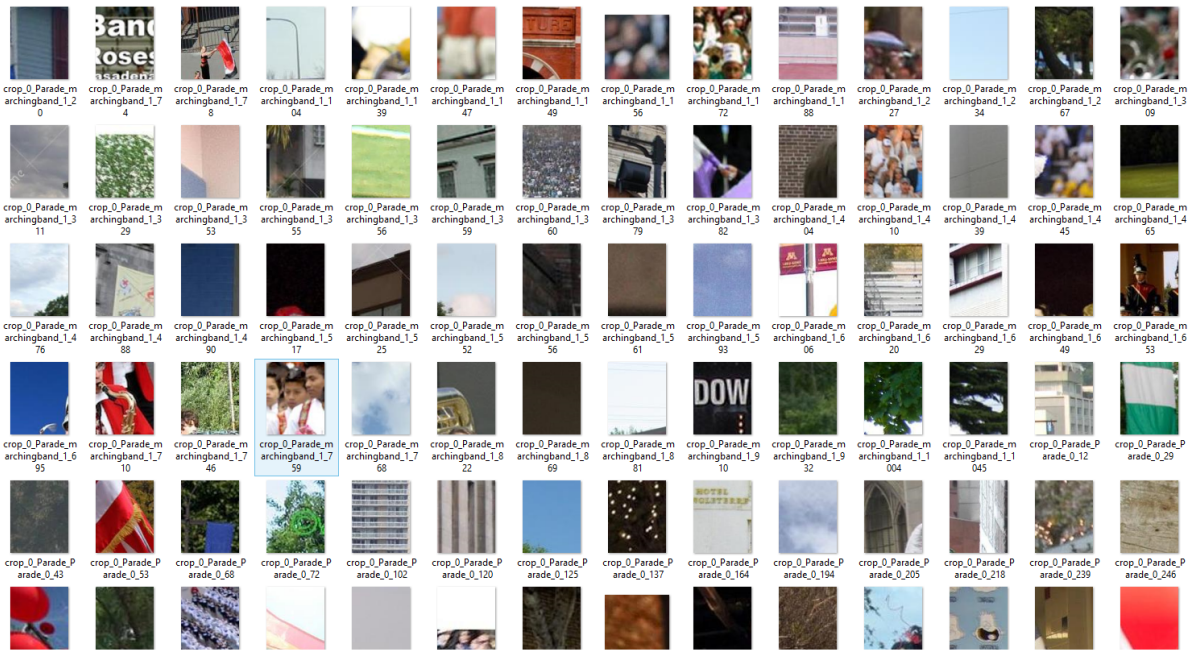

Figure 9: Some detections made by the algorithm.