HISTOGRAMS OF ORIENTED GRADIENTS AND EIGENFACE CUES FOR FACE DETECTION

Dr. Xiaojun Qi, Benjamin Baltazar Vera

Utah State University, Department of Computer Science

ABSTRACT

Face detection is an extensively studied problem in computer vision. We address this problem through a framework which integrates Histogram of Oriented Gradients in combination with Eigenface cues. We illustrate the performance of our framework on an idea case of images and demonstrate its advantages over other face detection implementations.

1. INTRODUCTION

There are many face detection algorithms in the literature. In general, they fall into two categories: Feature-based and Template based. In the former category, features extracted from subwindow located within a detection window are used to describe a face. This approach can be based on different types and combinations of features, such as histogram of oriented gradients (HOG), covariance matrices, combination of several features, and multi-level versions of HOG. On the other hand, the template-based uses the entire template, with generally a sum- comparing metric that determines the best location by testing all or a sample of the viable test locations within the search image that the template image may match up to.

2. OVERVIEW OF METHOD

This section gives an overview of our feature extraction method. This method is based on evaluating well-normalized local histograms of image gradient orientations in a dense grid. We also incorporate a set Eigenface features formed by performing a mathematical process called principal component analysis (PCA). For HOG, the basic idea is that local object appearance and shape can often be characterized rather well by the distribution of local intensity gradients or edge directions, even without precise knowledge of the corresponding gradient or edge positions. This is implemented by dividing the image window into small spatial regions ("cells"), for each cell accumulating a local 1-D histogram of gradient directions or edge orientations over the pixels of the cell. The combined histogram entries form a representation. For better invariance to illumination, shadowing, *etc*., it is also useful to contrast-normalize the local responses before using them. [2] This can be done by accumulation a measure of local histogram "energy" over somewhat larger spatial regions ("blocks") and using the results to normalize all of the cells in the block. We will refer to the descriptors blocks as *Histogram of Oriented Gradient HOG)* descriptors. Tiling the detection window with a dense (in fact, overlapping) grid of HOG descriptors and using the combined feature vector in a conventional SVM based window classifier and gives our first piece of the face detection framework. Each pixel calculates a weighted vote for an edge orientation histogram channel based on the orientation of the gradient element centered on it, and the votes are accumulated into orientation bins over local spatial regions that we call *cells*. Cells can be either rectangular or radial. The orientation bins are evenly spaced over 0°- 180° (.unsigned. gradient) or 0° - 360° (.signed. gradient). The vote is a function of the gradient magnitude at the pixel, representing soft presence/ absence of an edge at the pixel. The Eigenface method is based on linearly projecting the image space to a low dimensional feature space and uses principal components analysis (PCA) for dimensionality reduction. The implementation to create a set of Eigenfaces, one must first compile the training set of images. To reduce the error due to lighting conditions we normalize the set of images using the standard deviation and mean of the set of images. We continue by feeding our set of normalized images to our PCA. This will give is our Eigenvectors and Eigenvalues. The Eigenvectors are our Eigenfaces. We choose 10 (principle components) Eigenvectors with the largest associated Eigenvalues. We then obtain the final EigenFeatures buy subtracting our window by the mean of the set of images and multiply by the principle components. We continue by implementing a sliding window technique that generates a 18 x 27 window on an input image. HOG feature vector and EigenFeature are computed, the concatenated. This continues as the sliding window moves across the entirety of the input image. The results of each window is fed into our SVM and classified as face or non-face. An overlapping verification algorithm is implemented to determine if similar regions can be merged. Each window classified as having a face are compared to all other windows with the same classification. This comparison verifies if any overlapping occurs. If so, the compared windows are merged and displayed depending on the position and coordinates of the windows being merged. The histogram of oriented gradients capture the high interest areas in faces that are rich in gradient information (eyes, nose and mouth) that are quite robust to pose variations, and Eigenfaces captures the holistic appearance of the human face. These two feature channels capture a mix of global and local information about the face, and are robust to variations in pose.

3. DATA SETS AND METHODOLOGY

Our SVM was trained with a face/non-face data containing 69 (frontal) face and 55 non-face images of size 18x27. Before the SVM is trained, when preparing our image database, we apply HOG to the original image and varying flips and circle shifts of the original image to create more training data. This process is done 10 times for the face images and is repeated for the non-face images 4 times. The overall amount of training data for our linear SVM totals to 910 images, 690 face and 220 non-face.

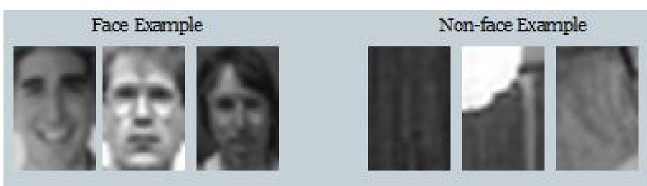

Figure 1. Some sample images from our new human detection database. Each image is 18x27.

We use a sliding window technique to capture feature data across an image. The algorithm creates a 18x27 window at the first pixel of the image and slides across the columns and then moves down one row after all the pixels in that columns have been traversed. Once all the features have been extracted from the windows, they are fed into SVM for classification. These results are stored into a cell that will contain their HOG features, EigenFeatures, window width, window height, coordinate information and block number. With this information, we are able to see where exactly the SVM classified a face. We isolate the face windows and check for overlapping in similar regions. Our window overlapping algorithm we used the stored coordinates of the windows being classified as having a face. To do this, we simply calculate the difference between each of the windows coordinates. Depending on the type overlap, a certain type of merging is applied. If the windows overlap, new coordinates from the merged window are returned and displayed on the input image.

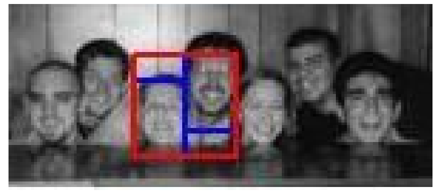

Figure 2. An example of windows merging after verification.

4. EXPERIMENTAL RESULTS

In this section, we demonstrate with experiments how our integration framework performs in terms of accuracy and speed. We achieved the best results when configuring HOG with an orientation binning number of 4 with a 3x3 filter to extract intensity information. We take the top 10 principle components for our Eigenface feature. Our sliding window is size 18x27. In total, there were 10 face candidate windows detected. We were able to achieve 80% accuracy without window merging. However, we achieved 100% on the merged windows. Each merged window contained at least 1 face. The time performance of our algorithm averaged less than 2 seconds in total run time.

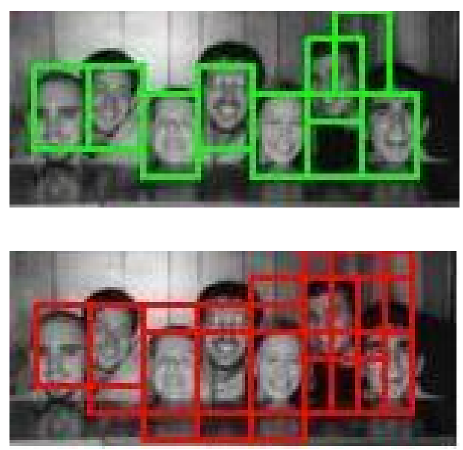

Figure 3. Our final results displaying our candidate window (top) and merged windows (bottom).

4.1. Template versus Feature performance

In this section, we compare two template-based face detection algorithms against our framework. We will evaluate their accuracy and speed again ours. The first template-based face detection algorithm used a Gabor filter with a neural network. A neural network utilizes how the appearance of faces differs from non-faces using training exemplars, and then detects faces by seeing how well the test data fits the learned model. This detector achieved 100% accuracy, however took a total average runtime of 37 seconds. The next detector is similar in the fact that it uses a Gabor filter; yet, this implementation uses a support vector machine to detect faces. This detector also achieves 100% face detection accuracy. In terms of speed, the total average running time was about 26 seconds. This is a definite improvement versus the neural network framework. Nevertheless, our framework tremendously reduces the time it takes to detect faces in an image.

5. CONCLUSION

We have described a novel framework that combines complementing feature extraction techniques that dramatically reduces the time it takes to detect faces in an image. Despite the lack of 100% of candidate faces containing faces, I believe that this approach can be improved and still retain the speed in which it can detect faces.

6. FUTURE WORK

Although our current linear SVM detector is reasonably efficient, there is still room for optimization and to further speed up detections. The inclusion of color cues might be useful and beneficial to this framework. The accuracy of our framework needs improvement. I believe this can be done with a better set of training data. The images in our current training data are too small for any real application of this framework. Thorough testing of this framework is needed to prepare for more challenging detections. I wish to apply this to pedestrian, human and object detection. In all, this framework can be improved to stand up against the best face detection algorithms, even at high resolutions and challenging detection photos.

7. ACKNOWLEDMENTS

This work was supported by the National Science Foundation. I'd like to thank Dr. Xiaojun Qi and everyone that contributed to the success of the University of Utah State Computer Science REU Site Program Computer Vision and Multi-Agent Systems.

8. REFERENCES

[1] William Robson Schwartz, Raghuraman Gopalan, Rama Chellappa, and Larry S. Davis. *Robust Human Detection Under Occlusion by Integrating Face and Person Detectors*

[2] Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. CVPR 1 (2005) 886{893 vol. 1

[3] Ludwig, D. Delgado, V. Goncalves, and U. Nunes. *'Trainable Classifier-Fusion Schemes: An Application To Pedestrian Detection,' In: 12th International IEEE Conference On Intelligent Transportation Systems, 2009, St. Louis, 2009. V. 1. P. 432-437.*