WEAKLY TRAINED DUAL FEATURES EXTRACTION BASED DETECTOR FOR FRONTAL FACE DETECTION

Wael Louis *, K.N. Plataniotis †

Multimedia Lab, The Edward S. Rogers Department of Electrical and Computer Engineering, University of Toronto, 10 King’s College Road, Toronto, Canada

ABSTRACT

This paper investigates the inconvenience of using huge number of features, enormous training dataset and lengthy training session to achieve a good performance frontal face detector. The proposed face detector is based on a novel idea which proposes using joint decision from two parallel different features trained detectors, one detector is trained with Local Binary Patterns (LBP) features and the other with Haar-like features. Both detectors are trained with few features using not a huge face/non-face dataset and within relatively short period of time. Hence, both detectors agree on the face image but seldom agree on the non-face image. The result is significantly improved using a multi-detections merging algorithm using simple clustering method. The robustness of the detector is examined once using a face/non-face dataset and compared to Lienhart frontal face detector, and secondly using a real-life sequence.

Index Terms— Frontal face detection, Weakly trained detectors, Small non-face images dataset, Feature extraction.

1. INTRODUCTION

Extensive research for face detection is conducted since the 1970s as it can be used for surveillance purposes, human tracking, human-computer interaction, etc. Many techniques are presented in Yang et al. [1]. As many appearance based face detectors (i.e. detectors that use feature extraction and classifiers for the detection criterion), a tremendous amount of time is spent in the training stage [2][3]. This time is spent on modifying the images by cropping and aligning them, collecting thousands of non-face images, deciding how many layers—in case of cascaded classifiers—are enough, how many features should each layer be trained with, etc. Also little research is conducted to find the relation among these problems or find an optimum solution [4]. Therefore, the decision of choosing the parameters of these problems is usually carried on based on trial and error, so much time is spent in this stage.

One of the most famous face detection techniques is Viola and Jones [2] which uses Haar-like features. This technique is considered as a breakthrough in the face detection due to its high detection rate and high speed of processing using the AdaBoost algorithm with cascaded classifiers. To achieve high detection rate, speed and low false positives, Viola and Jones used 6,060 Haar-like features distributed on 38 layers. The faces’ features were extracted using 4,916 face images and their vertical mirror. Despite the simplicity to implement the algorithm, weeks are required to train the system. Due to long training time, and the limitation of the Haar-like features in lighting variation, another approach was introduced in Hadid et al. [3] using the Local Binary Patterns features, which is texture descriptor features that has the capability to resist illumination changes [5], this technique required big faces/non-faces dataset of 12,000 faces and 20,560 non-face images. Furthermore, Improved Local Binary Patterns is used in Rodriguez [4] where 450 LBP features was distributed on 3 cascaded classifiers and was trained using the AdaBoost in order to achieve an effective system. Even though, this method succeeded in reducing the number of cascaded classifiers but it required huge face images and huge negative dataset selected from a collected database that consist of billions of non-face images. Another approach is also introduced by Chen et al. [6], instead of targeting the type of features or the design of the system, it tries to reduce the number of training set by obtaining an optimal subset of the full dataset. This technique also requires constructing a big faces dataset in order to choose the optimum subset.

Due to all explained problems, the method we are proposing is to implement a frontal face detector trained with few features in short training time. The desired technique is to be used for surveillance purposes using a 2D information from static camera mounted in a position where mostly frontal faces are captured. The novel frontal face detection method (hereafter parallel face detector) is to extract two types of highly proven face discriminant features. Unlike the series cascaded classifiers face detectors, the introduced method uses only two classifiers, cascaded in parallel, each is trained by few number of one type of features. The final decision is made by a logical AND between the decision of each detector. Hence, this method is designed for finding the faces instead of discarding the non-faces. The paper is organized such that section 2 describes the technical details including the feature extraction and the detector design, followed by section 3 where the conducted experiments as well as the training and evaluation phases are discussed. Finally the conclusion is in section 4.

2. TECHNICAL DETAILS

A novel approach is proposed for face detection using parallel detection and joint decision from two individually weakly trained detectors, each detector is trained with different types of features. A complete block diagram of the parallel face detector is in Fig.1. One of the weak detectors is trained with LBP features and the other with Haar-like features. These features were specifically chosen not only because they give discriminant face features but because they can be extracted quickly so they can be used for real-time detection (i.e. 15 frames/second) as in [2][4]. Also each of these features target different image structure such that the LBP features are used as a texture
descriptor by having the ability to detect corners, edges, spots and flat ends [3], also it has a high tolerance to illumination changes [7] while Haar-like features are excellent for small details structure, edges and bars [2] and are capable to be computed very quickly using the integral images. Hence, each will have different detection criteria to detect the face. Similar approach of using these types of features together is recently investigated in [8].

Based on Viola and Jones [2] or LBP [4] detectors, both use the AdaBoost algorithm [9], each layer of the cascaded classifiers, regardless of the number of face features is trained with, is capable of detecting \( \approx 100\% \) of the faces; however, the more the number of features it is trained with the less the false positive detections. The AdaBoost method uses weak classifiers \( h_i(X) \), where each \( h_i(X) \) is a single feature. The AdaBoost algorithm weighs and select the best \( n \) weak classifiers, where each weak classifier can minimize the classification error to construct a strong classifier \( H(X) = \sum_{i=1}^{n} \alpha_i h_i(X) \). At each iteration of the boosting, the best weak classifier \( h_i(X) \) is chosen, and the weight \( \alpha_i \) is increased to the wrongly classified samples for the next iteration.

Consequently, choosing only small number of highly distinctive features for each detector using the AdaBoost algorithm will results a high percentage of faces detected in the image. Hence, each detector individually is adequate to find the face; however it produces many false positives due to the fact that many non-face images might have the detector’s trained features since they are only few ones. Therefore, both detectors will agree when a face image is passed and disagree when a non-face image is passed. This statement is true since every face image contains best LBP extracted features and best Haar-like extracted features that each detector is trained with; however most of false positive detections detected by one detector don’t have both best LBP extracted features and best Haar-like extracted features. Therefore the idea which was originally proposed by Rowley et al. [10] of using logical AND is used to drastically reduce the false positives and preserve the true positives.

2.1. LBP features extraction

Simple LBP features extracting algorithm operates by thresholding. It takes the value of the center pixel in a \( 3 \times 3 \) pixel window in grayscale image– and consider it the threshold. Then assign 1 to any of the neighboring pixels with a value greater than the threshold or 0 if it is less than the threshold. The resulted binary value is then converted to decimal. The histogram of the LBP values from the image works as a descriptor for the image.

The LBP features don’t get extracted on only 1 pixel neighbor or a square window, but as in Ojala et al. [7], with a circular neighbor with different radii and points. Where points are considered as the number of equally spaced points that construct the LBP operator and the radius is how far the points from the central pixel. This circular operator is symbolized as \( LBP_{P,R} \), where \( P \) and \( R \) correspond to the value of points and radius respectively. There are \( 2^P \) binary words for each LBP window. Also in Ojala et al. [7], it is found that there is a subset of the \( 2^P \) that spans most of the texture descriptor, this subset is called uniform LBP, \( LBP_{P,R}^{u} \). The uniform LBP are the words that have only two bit flipping from 0 to 1 and 1 to 0 (e.g. 01110000).

The LBP features extraction in the parallel face detector is designed to extract features in two phases, first phase is to extract the features based on the entire face image to obtain the overall description. For this purpose two LBP features extraction are used, one is the \( LBP_{P,R}^{u} \) and the other is \( LBP_{P,R}^{u} \). The second phase is to detect the smaller face description; therefore, \( LBP_{P,R}^{u} \) is used with overlapping windows to target many places of the face. The LBP features extraction procedure is illustrated in Fig. 2.

In Hadid et al. [3] only 2 LBP were used so almost same preference is given for information obtained from within the face region (i.e. using small overlapping windows) and information obtained from the entire face (i.e. no overlapping window). However, to get more distinctive features, 3 LBP features extraction combination is used, where 2 of them are for the overall face. The explained combination gives the best results on the tested data. Having higher number of LBP features in the combination increases the computation complexity and wouldn’t increase performance significantly.

2.2. Haar-like features extraction

The Haar-like features extraction is based on the work of Viola and Jones [2] which subtracts the sum of pixels of grayscale image in two adjacent rectangles. These two rectangles, one is considered as black and the other is white region. The final feature result is the subtraction of the sum of the white region from the sum of the black region. These adjacent rectangles can be in different templates such as being 2 or 3 horizontally or vertically adjacent, 4 adjacent regions in a square shape, etc. The Haar-like features work well in places where there are differences in brightness in the object as in the human face such as the area between the eyes and the forehead or the area between the eyes and the bridge of the nose. These two areas are found to be the most discriminant areas in the human’s face [2]. The feature extraction starts by a feature of smallest size (i.e. 2 pixels in the 2 rectangle template) and exhaust the image many times by keep

---

Fig. 1. Parallel face detector system.

Fig. 2. LBP features extraction diagram.
increasing the size of feature until it reaches one of the dimensions of the image. This search is performed quickly using the integral image.

2.3. Multi-detection merging

Multi-detections issue is considered as a drawback in many appearance based face detector due to the fact that the features are insensitive to small error changes [2][4]. Hence while the scanning window scans the image, many detections might be located for the same detection. However, due to the fact that the parallel face detector will rarely agree on a false positive, then the number of detections in false positive regions are significantly less than those multi-detections in the faces regions.

The algorithm is based on only one parameter. It finds the centroid position of each detection, then cluster these positions with mode finding using Euclidean distance from the points to clusters' centroid regions as a similarity measure. The threshold $\beta$ is considered as the minimum number of particles in each cluster to be considered as a detection. All the clusters that don’t pass this test get deleted. Afterwards, the leftover clusters’ centroid positions mean values are taken as the centroid of that detection. The size of the detected face is calculated as the average size of all the detections of the centroids that constituted that cluster.

3. EXPERIMENTAL RESULTS

Following the objective of this paper of having small training set. The system is trained using Viola and Jones faces dataset, it consists of 4,916 grayscale face images of size 24 \times 24 pixels. No extra cropping, resizing and aligning are performed on the data. Also only 7,872 grayscale non-face images of size 24 \times 24 pixels are used in the training. This is considered as the baseline face size. In the literature, many face template sizes were used such as 19 \times 19 pixels in [3][4][10], 24 \times 24 pixels in [2][11]. The reasons behind using the 24 \times 24 pixel template size is based on keeping the Viola and Jones dataset as it is since one of the implemented weak detectors uses Viola and Jones approach, and they proved [2] that this size gave them the best results.

The performance of the face detector is examined in two experiments, first is based on part of the Ole Jensen dataset [11]. This dataset contained 5,000 grayscale face images and 10,000 of grayscale non-face images. The results are illustrated using the Performance Rate (PR), True Positive Rate (TPR), and False Positive Rate (FPR). $PR = \frac{TP}{TP + FN} \times 100$ where $TP$ is the number of correctly classified faces as faces, $TN$ is the number of correctly classified non-face as a non-face, and $T$ is the total number of images in the dataset. $TPR = \frac{TP}{24^2} \times 100$, where $NP$ is the total number of face images in the dataset. $FPR = 1 - \frac{FP}{24^2} \times 100$, where $FP$ is the number of wrongly classifying a non-face image as a face image and $Nn$ is the total number of non-face images in the dataset. Second experiment is using a real-life example which is based on a footage from a realistic environment where data became available to the University of Toronto team for research purposes. The footage is taped by a camera mounted on the ceiling in vantage to capture frontal faces. The footage is in RGB colormaps of Codec Video 1 format with video rate of 5 frames/second. Also, the sequence is of low resolution of 300 \times 243 pixel, decent illumination, and multiple faces but non-crowded area. Faces appear in different sizes up to 60 \times 80 pixel width \times height. The performance evaluation for first experiment is based on whether a detection occur or not; however, similar to Lienhart et al. [12] approach is used to consider the detections in the footage frames as a correct detection. A face is considered as correct detection when the Euclidean distance between the actual measured face and the detector’s face is less than 40% of the width of the detection’s size, also the detection’s height and width are less than 40% of the actual face width and height. The actual face is considered as the region from above the eyebrow to the end of the chin since the dataset that trained the system was originally cropped such that the bounding box from above the eyebrow to below the mouth and is increased by 50%.

Due to the fact that the more features the classifiers are trained with the better the detection with tradeoff of increasing the computational complexity. Hence, the weak LBP detector is trained using the AdaBoost algorithm with 50 face features. Only 1 strong classifier is trained. As explained in section 2.1 that $LBP_{8,12}^u$ and $LBP_{12,2}^u$ are used for full face description; hence, window of size 24 \times 24 pixels is used whereas for the smaller $LBP_{1,1}$ feature, a scanning window of size 12 \times 12 pixels and shifting of 2 pixels are used. For each image there are 59 features from the $LBP_{8,1}^u$, 135 features from $LBP_{12,2}^u$, 784 features from $LBP_{8,1}$. The total number of features extracted from each 24 \times 24 pixel image is 978 features. The 784 features of $LBP_{8,1}$ is calculated as there are $LBP_{8,1} = 2^8 = 16$ features in each 12 \times 12 pixel window; therefore, the 24 \times 24 pixel image is subdivided into 49 windows each of 12 \times 12 pixels overlapped by 10 pixels. Therefore, these features are 49 \times 7 = 784. Since $LBP_{12,2}^u$ and $LBP_{12,2}^u$ features don’t have overlapping windows, so they are based only on extracting the uniform features. The smaller the scanning window and smaller the shifting the more the features to be extracted hence more computation is required. Training this detector took 9 minutes using Intel(R) Core(TM)2 Duo CPU T9400 @ 2.53 GHz and 3.00 GB RAM, in comparison to hours in [4]. Also only few features are chosen to train the weak Haar-like detector. Training a strong classifier with only 40 features took \approx 23 hours on a computer with specification of Intel(R) Pentium(R) 4 CPU 3.00GHz and Memory of 1024 MB RAM. In comparison to weeks of training in the actual Viola and Jones detector [2].

The final face detector is the logical AND of the result of both detections. The ROC curve for the performance on the Ole Jensen dataset is in Fig. 3. A comparison is conducted between the parallel detector and Lienhart detector [12] on the same Ole Jensen dataset, the result is tabulated in Table 1. Since most appearance based face detectors require a huge dataset and a very long training time then it won’t be possible to duplicate the results. Lienhart detector is considered for comparison for two reason, first its implementation is very close to the Viola and Jones detector. Also it is available on OpenCV, so most of the papers in the literature compare their results with Lienhart face detector. Due to the explained limitations, the parallel face detector is only compared to Lienhart detector.

\begin{table}
\centering
<table>
<thead>
<tr>
<th>Detectors</th>
<th>Results</th>
<th>TPR %</th>
<th>FPR %</th>
<th>PR %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lienhart detector</td>
<td>94.90</td>
<td>0.22</td>
<td>96.82</td>
<td></td>
</tr>
<tr>
<td>Parallel face detector</td>
<td>97.40</td>
<td>0.31</td>
<td>98.97</td>
<td></td>
</tr>
</tbody>
</table>
\caption{Comparison between the parallel face detector and Lienhart detector.}
\end{table}

Five scenarios from the footage are shown in Fig. 4 to illustrate the behavior of the system in real-life scenarios. Three scenarios are for single frontal face in different positions in order to examine the performance with different face sizes. Fourth scenario is for a frontal slightly occluded face, to show the capability of handling slight occlusion, and the fifth scenario when more than one person
which have the features that the detector is trained with. Consequently, when these detectors are placed in parallel and logical AND is made, then they agree on the face window and disagree on the non-face window. Furthermore, the parallel detector is insensitive to small errors, so multi-detections occur in the face region, while less number of multi-detections occur in the non-face region; hence, multi-detections merging algorithm is implemented that could significantly improve the results since it does not only merge detections on the face region but also discard detections that belong to non-face regions.

5. REFERENCES


4. CONCLUSION

Novel frontal face detector is implemented based on a joint decision between two parallel weakly trained detectors. One of the detectors is based on the LBP features extraction and the other is based on Haar-like features extraction. Both detectors are implemented with 1 strong classifier and trained using the AdaBoost algorithm. The LBP detector is trained with 50 features whereas the Haar-like detector is trained with 40 features. Despite the fact these detectors individually would not perform well but having a joint decision between them produces a promising results. The reason behind having a good results is that each detector is trained with only few best face distinctive features represented differently. Hence, each detector individually is capable of detecting the faces but would also make many false positive detections due to the non-faces images appear in the screen, to illustrate the ability of detecting more than one face. The detection is conducted using a baseline subwindow of size $24 \times 24$ pixels that scans the entire image with a shift of 2 pixels. The window is scaled by 1.25 each time and its content is downsampled to $24 \times 24$ pixels.