

Face Detection Using Information Fusion

Parham Aarabi, Jerry Chi Ling Lam, Arezou Keshavarz

Modiface Inc. and University of Toronto

Toronto, Ontario

parham@ecf.utoronto.ca, jerry.lam@utoronto.ca, arezou.keshavarz@gmail.com

Abstract – The fundamental point of this paper is that the fusion of several simple, somewhat unreliable, and somewhat inefficient frontal face detectors results in an efficient and reliable frontal face detector which, without any training, performs similarly to a state-of-the-art neural network based face detector trained on 60,000 images. The simple detectors used include a skin detector, symmetry detectors, as well as structural face detectors. On a test set of 30 color images containing frontal faces, the fused face detector had an accuracy of 93% with a RMSE of 4.96 pixels, as compared to an accuracy of 87% and a RMSE of 8.00 pixels for the neural network based face detector. On the Caltech Face Database, the fused face detector had a 90% detection rate which is on par with state-of-the-art face detection methods that utilize extensive prior training, including the neural network approach which achieves a detection rate of 94%.

Keywords: face detection, image fusion, detector fusion

1 Introduction

A central notion of information fusion is that the combination or fusion of a set of information sources can attain more reliable information than any individual source. In this paper, we will attempt to illustrate and quantify this in the context of a face detection task. Our application, called ModiFace, which is running live on modiface.com, utilizes a face detector to automatically detect the face of a person in an image followed by a set of image operations to simulate plastic surgery or other simple cosmetics operations. This is the culmination of the work that was initially presented in [2], though the face modification algorithms have been substantially improved since the publication of [2].

Since the face detection task is running live on the web, it is required to be fast (since the user expects the results in real-time), efficient (since many instances of the detection may be running depending on the number of users on the site), accurate (since accuracy is needed in the plastic surgery visualization task), and reliable. Furthermore, the domain of the face detection application is any image submitted by any user, which implies that the face detector

may encounter a wide variety of image formats, color situations, lighting conditions, etc.

In this paper, we will propose several extremely simple face detection approaches. These approaches are not based on any prior training, but are instead based on common sense metrics. It is shown that the combination or fusion of these approaches results in a relatively accurate, face, and reliable face detector which outperforms all of the individual detectors and, without any training, performs similarly to state-of-the-art face detectors which require extensive training.

2 Face detection prior work

Face detection, which is the problem of automatically detecting and locating a face or a set of faces in an image, has been the focus of extensive research in the past few decades [1,4,5,8,9]. It has been used as a precursor for face recognition [11], or as the basis for features extraction applications such as lip tracking [3] or face feature visualization [2]. A complete and thorough description of face detection algorithms appears in [12].

In this section we will provide a brief overview of three popular face detection approaches. They consist of template matching, neural networks, and linear subspace methods.

2.1 Template based face detection

Perhaps the simplest method of finding a face is to first develop a face template which detects the likelihood of a face at a certain location of the image, and to then check for the existence of the face at all possible locations within the image using various face box sizes [1]. A face template can be an RGB model of the face, or can be a probabilistic model of a face in other domains (such as in the edge domain or the HSV domain).

Template-based methods can be very reliable, though computationally inefficient. In order to find a single face, all possible locations of the image must be searched, with a variety of face sizes, and possibly with different rotations. This search process makes the simple template

models unattractive for computationally-limited applications.

Also, the choice of a template is an extremely important parameter in the design of template-based face detectors. Edge or gradient based models are less sensitive to lighting conditions, and as a result, are most commonly used [1].

In this paper, the fusion of multiple template-based face detectors is proposed. The details of the individual detectors as well as the fused detector will be described in section 3.

2.2 Neural Network based face detection

Neural networks are a popular choice for face and visual object recognition applications [6,7]. For the application of face detection, a multilayer feed-forward neural network whose input layer is directly connected to the image pixels learns to detect faces by providing it numerous face and non-face examples and by training the network using back-propagation.

In this paper, the neural network methodology of [6] was implemented for comparison with our novel fusion-based face detector. The neural network face detector takes an image of size 35x35 pixels as input and consists of a total of six layers. A shared weight neural network architecture such as the one described by [10] was utilized.

The first layer (i.e. input layer) is a convolutional layer with 4 feature maps, each of size 32x32 with each unit being connected to a 4x4 pixel grid of the input image. The second layer is a sub-sampling layer with four feature maps, each of size 16x16. The third layer is a convolutional layer with 15 feature maps each of size 14x14 with each unit being connected to a 3x3 neighborhood of the feature maps in the second layer. The fourth layer is a sub-sampling layer with 15 feature maps of size 7x7 with each unit being connected to a 2x2 area in the previous layer's corresponding feature maps.

In the fifth layer, each of the 15 neurons is fully connected to all units of only one corresponding feature map of the fourth layer. The single neuron in the output (i.e. sixth) layer is fully connected to all the neurons of the fifth layer. This weighted sum is then passed through a logistic function to produce the state of the unit in between 0 and 1 with 0 representing the absence of a face and 1 the presence of a face.

The neural network face detector synthesizes feature extractors and builds classifiers using the extracted features from a training set of face and non-face patterns in a single integrated scheme. All parameters in the system are trained with the mini-batch back-propagation algorithm

with momentum [7] on 30000 face and 30000 non-face images.

To localize faces of different sizes in images, the input image is repeatedly sub-sampled by a factor of 0.9, resulting in a pyramid of images. Each image of the pyramid is then filtered by the network. After processing by this detection pipeline, face candidates (pixels with positive values in the resulting image) in each scale are mapped back to the input image scale. They are then grouped according to their proximity in the image. Each group of face candidates is fused in a representative face whose center and size are computed as the centroids of the centers and sizes of the grouped faces, weighted by their corresponding network responses. For this paper, only one face is assumed to appear in each image and therefore, the face candidate with the highest response is returned as the final face estimate.

2.3 Subspace methods for face detection

Subspace methods for face detection divide images into several overlapping patches (of possibly different sizes) and calculate the projection of each patch onto a set of vectors, which serve as the bases of the "face" space. The projection error for each of these patches is thresholded to identify whether a patch has face or non-face characteristics.

The eigenface method, which was first presented in [11], has been used in several face detection and face recognition applications. This method forms a face subspace by calculating the eigenvectors from the face images in the training set. Given a training set of M grayscale mean-subtracted images, each of dimensions X by Y , a matrix A of size (XY) by M is constructed; each column in the matrix A is created by reshaping each training image into a column vector. Principal Component Analysis (PCA) is utilized to reduce the dimensionality of the problem. Thus, only the N prominent eigenvectors (with higher eigenvalues) of A are chosen, corresponding to the main contributors to the image reconstruction (i.e. the "face" space bases).

Each patch is projected onto the face space and then reconstructed in that subspace using only the prominent eigenvectors. The resulting reconstruction error is defined as the mean square difference between the original patch and the reconstructed patch. This reconstruction error can be used as an inverse face likelihood function, where lower errors are indicative of a higher likelihood that the current patch is a face. In this paper, the eigenface-based face detection method was implemented for comparison purposes. For this purpose, a database of 10 faces was used to compute the face-space eigenvectors.

3 General architecture of the proposed fusion-based face detector

Face search inside a user submitted image using the proposed algorithm starts with a large box with the height to width ratio being 4 to 3, and with the width being 60% of the image width. The box is moved to all locations inside the image one pixel at a time for greatest accuracy (this can obviously be improved for further efficiency). After the entire image has been searched, the width of the box is decreased (again one pixel at a time) with the height to width ratio remaining fixed at 4 to 3.



Figure 1 – The basic procedure for face detection, starting with a search over the entire image with a large face box, and continuing with smaller face boxes until a suitable hit or a minimum face box threshold is reached.

For all box sizes and box locations, a face score is computed which corresponds to the likelihood of a face at that location. The box with the highest score is chosen as the face location and size estimate. This is similar to most template-based face detection algorithms [1,4].



Figure 2 – An example of the search space for face detection. Note that in practice the candidate face boxes will have significant overlap and will be of various sizes.

The most difficult and sensitive part of this template based face detection is the score computations. In the section below, we will outline several candidate face score metrics.

4 Face Detection Metrics

4.1 Preliminaries

Before we describe our choices for the face score metrics, we first need to define a few variables. In this paper,

$I(x,y)$ will denote the original image at location (x,y) . The content of each location are a 3-element vector of red, green, and blue components (i.e. RGB) for each pixel. $\Psi(x,y)$ will denote the gradient magnitude (in the vertical direction) of the image $I(x,y)$, and consists of a non-negative value for each location (x,y) . Finally, $T(x,y)$ will be a binary template image used to fit a binary template to the face.

Please note that all face score metrics are a function of the location and size of the current box. The top left corner of this box is denoted as (x_0, y_0) , while the width and height of the box are denoted as W and H , respectively.

It should also be noted that for visual simplicity, we have used $E[.]$ to denote the expected value of a certain variable, where the expectation is usually performed over the x and y variables. This theoretical expectation is practically estimated as follows:

$$E[Z(x, y)] \cong c \cdot \sum_y \sum_x Z(x, y)$$

where $Z(x,y)$ is some function of x and y , and where c is a normalization constant.

4.2 Symmetry-Based (SYM) Face Detection Metric

Our first metric is simply a mirror mean square error measure applied to the image gradient. It consists of folding the current face box from the middle and taking the average of the pair-wise square difference of the points inside the box that overlap, as shown in the figure below.

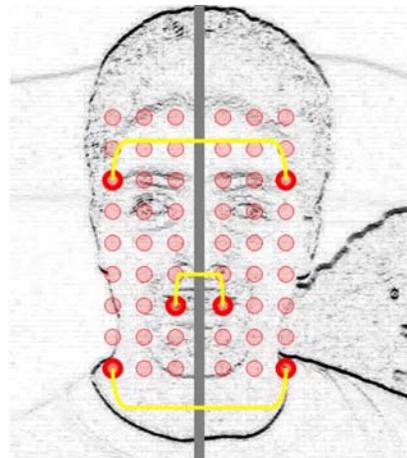


Figure 3 – Face symmetry calculation where the average pair-wise square error between mirror pixels is used as an indication of the face asymmetry (or, the inverse of it as an indication of the face symmetry).

The motivation here is that if the box is perfectly symmetrical in the vertical axis running through the middle of the box, then it will have the smallest mean

square difference. In other words, this is a measure of horizontal symmetry of the current face box.

For the actual metric, we add a constant (in this paper, this constant is 1) to the mean square difference and invert, so that a higher metric is indicative of greater symmetry. The final symmetry metric is shown below:

$$F_{\text{sym}}(x_0, y_0, W, H) = \left(1 + E\left[\left|\Psi(x, y) - \Psi(W + 2x_0 - x, y)\right|\right]\right)^{-1}$$

$$\cong \left(1 + \frac{2}{HW} \sum_{x=x_0}^{x_0+W/2-1} \sum_{y=y_0}^{y_0+H-1} \left|\Psi(x, y) - \Psi(W + 2x_0 - x, y)\right|\right)^{-1}$$

4.3 Template Subtraction (TS) Face Detection Metric

The template subtraction metric measures the degree to which the current face box resembles a face. It consists of applying a binary face template which is used to differentiate between the high gradient and the low gradient regions of a face.

In this paper a very simple template (below left) was utilized, though other choices (such as below right) would yield similar results. These models were based on general characteristics of the face (i.e. drawn by the author in a matter of seconds), and were not in any way trained or optimized for the face detection task

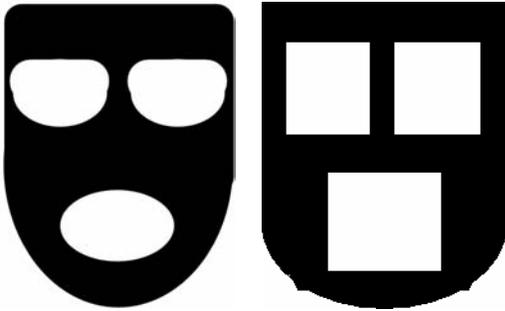


Figure 4 – Two possible templates for face detection purposes are shown. Although they both performed similarly, the template on the left was used throughout the experiments in this paper.

The template subtraction metric can be simply stated as the average gradient magnitude of the pixels corresponding to the white (1) template pixels, minus the average gradient magnitude of the pixels corresponding to the black (0) template pixels. In other words, the template subtraction metric can be defined as:

$$F_{\text{TS}}(x_0, y_0, W, H)$$

$$= E[\Psi(x, y) | T(x, y) = 1] - E[\Psi(x, y) | T(x, y) = 0]$$

$$\cong \frac{\sum_{x=x_0}^{x_0+W-1} \sum_{y=y_0}^{y_0+H-1} \Psi(x, y) \cdot T(x, y)}{\sum_{x=x_0}^{x_0+W-1} \sum_{y=y_0}^{y_0+H-1} T(x, y)} - \frac{\sum_{x=x_0}^{x_0+W-1} \sum_{y=y_0}^{y_0+H-1} \Psi(x, y) \cdot (1-T(x, y))}{\sum_{x=x_0}^{x_0+W-1} \sum_{y=y_0}^{y_0+H-1} (1-T(x, y))}$$

This simple operation can be best represented as follows:

$$F_{\text{TS}} \approx \sum \text{[Face Template]} - \sum \text{[Mask Template]}$$

4.4 Template Ratio (TR) Face Detection Metric

The template ratio, which is another template based metric, is simply the average gradient magnitude of the pixels corresponding to the white (1) pixels of the template divided by the sum of both the average gradient magnitude of the white (1) template pixels and the average gradient magnitude of the black (0) template pixels, as defined below:

$$F_{\text{TR}}(x_0, y_0, W, H)$$

$$= \frac{E[\Psi(x, y) | T(x, y) = 1]}{E[\Psi(x, y) | T(x, y) = 1] + E[\Psi(x, y) | T(x, y) = 0]}$$

$$\cong \frac{\left(\frac{\sum_{x=x_0}^{x_0+W-1} \sum_{y=y_0}^{y_0+H-1} T(x, y) \cdot \sum_{x=x_0}^{x_0+W-1} \sum_{y=y_0}^{y_0+H-1} \Psi(x, y) \cdot (1-T(x, y))}{\sum_{x=x_0}^{x_0+W-1} \sum_{y=y_0}^{y_0+H-1} T(x, y)} \right)^{-1}}{\left(\frac{\sum_{x=x_0}^{x_0+W-1} \sum_{y=y_0}^{y_0+H-1} (1-T(x, y)) \cdot \sum_{x=x_0}^{x_0+W-1} \sum_{y=y_0}^{y_0+H-1} \Psi(x, y) \cdot T(x, y)}{\sum_{x=x_0}^{x_0+W-1} \sum_{y=y_0}^{y_0+H-1} (1-T(x, y))} \right)^{-1}}$$

This simple operation can be best represented as follows:

$$F_{\text{TR}} \approx \frac{\sum \text{[Face Template]}}{\sum \text{[Face Template]} + \sum \text{[Mask Template]}}$$

4.5 Skin-Detector-Based (SKIN) Face Detection Metric

A simple pixel skin detector was employed to find the skin-like regions inside the image (based on the skin detection approach of [8]), and to use the amount of skin in each test patch as an indication of the likelihood of a face.

The pixel (x,y) of image $I(x,y)$ is skin, or $\text{skin}(I(x,y))=1$, if the red (R), green (G), and blue (B) components of that pixel obey the following conditions:

$$R>95 \text{ AND } G>40 \text{ AND } B>20 \text{ AND } R-G>15 \text{ AND } R>B$$

OR

$$R>220 \text{ AND } G>210 \text{ AND } B>170 \text{ AND } |R-G|\leq 15 \text{ AND } R>B \text{ AND } G>B$$

Please note that the above rules are functionally identical to the skin criteria of [8], though the logic functions have been simplified.

The skin-based face detection metric can thus be defined as:

$$F_{\text{skin}}(x_0, y_0, W, H) = E[\text{skin}(I(x, y))] \\ \cong \frac{1}{HW} \sum_{x=x_0}^{x_0+W-1} \sum_{y=y_0}^{y_0+H-1} \text{skin}(I(x, y))$$

4.6 Eye-Lip Total Symmetry (ELTS) Face Detection Metric

The ELTS metric measures the ratio of the sum of gradients in the top half of the face to the sum of gradients in the whole face, as defined below:

$$F_{\text{ELTS}}(x_0, y_0, W, H) = \frac{E[\Psi(x, y) | x, y \text{ in top half of face}]}{E[\Psi(x, y)]} \\ \cong \frac{\sum_{x=x_0}^{x_0+W-1} \sum_{y=y_0}^{y_0+H/2-1} \Psi(x, y)}{\sum_{x=x_0}^{x_0+W-1} \sum_{y=y_0}^{y_0+H-1} \Psi(x, y)}$$

Ideally, a face should have strong gradients around the eyes and the lips/nose, making the ideal ELTS measure at around 0.5. As a result, the following adjustment is done to the final ELTS measure:

$$\hat{F}_{\text{ELTS}}(\cdot) = \min(F_{\text{ELTS}}(\cdot), 1 - F_{\text{ELTS}}(\cdot))$$

4.7 Eye Total Symmetry (ETS) Face Detection Metric

Similar to the ELTS, the ETS measures the symmetry of the total gradients in the top half of the face. It is the ratio of the gradient sum in the top left quadrant of the face to the gradient sum of the top half of the face, as defined below:

$$F_{\text{ETS}}(x_0, y_0, W, H) = \frac{E[\Psi(x, y) | x, y \text{ in top left quadrant}]}{E[\Psi(x, y) | x, y \text{ in top half of face}]} \\ \cong \frac{\sum_{x=x_0}^{x_0+W/2-1} \sum_{y=y_0}^{y_0+H/2-1} \Psi(x, y)}{\sum_{x=x_0}^{x_0+W-1} \sum_{y=y_0}^{y_0+H-1} \Psi(x, y)}$$

As before, in an ideal case the ETS measure should be a 0.5. Consequently, we perform the following adjustment to the ETS measure to ensure that its maximum value is 0.5:

$$\hat{F}_{\text{ETS}}(\cdot) = \min(F_{\text{ETS}}(\cdot), 1 - F_{\text{ETS}}(\cdot))$$

4.8 Lip Total Symmetry (LTS) Face Detection Metric

Just like the ETS, the LTS measure the symmetry of the gradient sums in the bottom half of the image, as defined below:

$$F_{\text{LTS}}(x_0, y_0, W, H) = \frac{E[\Psi(x, y) | x, y \text{ in bottom left quadrant}]}{E[\Psi(x, y) | x, y \text{ in bottom half of face}]} \\ \cong \frac{\sum_{x=x_0}^{x_0+W/2-1} \sum_{y=y_0+H/2}^{y_0+H-1} \Psi(x, y)}{\sum_{x=x_0}^{x_0+W-1} \sum_{y=y_0+H/2}^{y_0+H-1} \Psi(x, y)}$$

As before we adjust the LTS such that its maximum and ideal value is 0.5, as follows:

$$\hat{F}_{\text{LTS}}(\cdot) = \min(F_{\text{LTS}}(\cdot), 1 - F_{\text{LTS}}(\cdot))$$

4.9 Fusion (FUSION) Face Detection Metric

It was determined that combinations of the above parameters results in the most reliable face detection results. The following combined detector, which is a fusion of five of the above metrics, is our final detection metric:

$$F_{\text{fusion}}(\cdot) = F_{\text{skin}}(\cdot) \cdot F_{\text{sym}}(\cdot) \cdot F_{\text{TS}}(\cdot) \cdot F_{\text{TR}}(\cdot) \cdot F_{\text{ELTS}}(\cdot)$$

The FUSION face detection metric, while only utilizing five detectors in its face score calculation, utilizes other metrics for optimization, as described in section 6.

5 Results and Discussion

A face detection experiment was performed on a set of 30 celebrity faces. These faces were mostly frontal without any rotation. Also, each image contained exactly one face. As a result, the reported results include only the detection

rate, since ROC curves, number of false positives, and number of false negatives here are unnecessary. In essence, the number of false negatives (i.e. the missed faces) will be the same as the number of false positives (i.e. the incorrect face position estimates for the missed faces) and equal to 100% minus the detection rate.

The many face detectors discussed in this paper were tested based on their reliability and accuracy. Reliability was measured as the percentage of correctly detected faces (based on the manual markings of the face in each image). A face was deemed to be correctly detected if the left, top, right, and bottom boundaries of the detected face were all less than 10 pixels away from the boundaries of the manually marked faces.

Another measure, related to the accuracy of the detected faces, consisted of the root mean square error (RMSE) of the face box coordinates. This value was calculated as the square root of the mean square distance error of the top-left corner of the face box plus the mean square distance error of the bottom-right corner of the face box. The RMSE was measured separately for the correctly detected faces and the incorrectly detected faces.

The table below summarizes the results of the experiment.

Face Detection Algorithm	Detection Rate	RMSE for correct detections (in pixels)	RMSE for incorrect detections (in pixels)
Symmetry	0%	N/A	133.99
Template Subtraction	83.33%	5.33	57.59
Template Ratio	33.33%	4.70	139.15
Skin Detector	10%	10.05	57.55
Eye-Lip Total Symmetry	0%	N/A	92.82
Eye Total Symmetry	0%	N/A	83.46
Lip Total Symmetry	0%	N/A	93.00
EigenFace-based	23.33%	5.03	42.48
Convolutional Neural Networks	86.67%	8.00	23.23
Fusion	93.33%	4.96	69.38

Table 1 – The correct face detection rates for various face detectors using a set of 30 celebrity images (shown in Figure 5).

As shown, the fusion of the simple face detectors outperforms all competing algorithms that were tested.

The only two errors that were made by the fused detector are shown in the figure below (as the rightmost images in the bottom row).

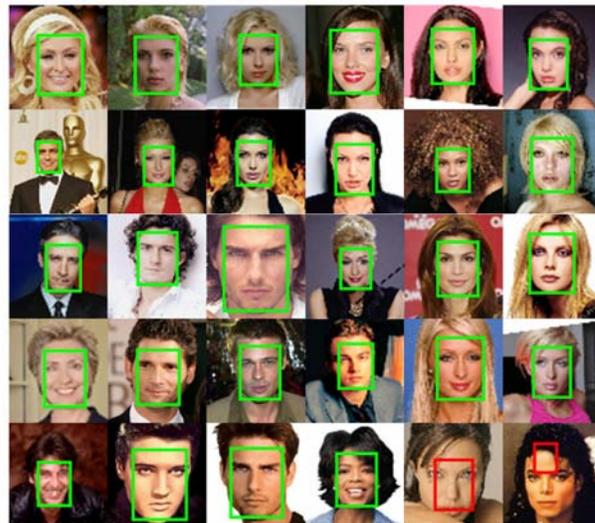


Figure 5 – The zoomed in images of 30 celebrity faces used to test the various face detectors. The face detection results of the fused detector are shown on top of the images. Out of 30 images, only two detection errors (based on the face box coordinates) were made. The two errors are the rightmost two images in the bottom row.

The fused detector not only has the highest detection rate of any approach, but it also has a very low RMSE measure. It has a detection rate that is 10% higher than any of the simple approaches and almost 7% higher than the neural network based approach which was extensively trained.

The eigenface-based detector performed quite poorly, though the rather low RMSE for the incorrect detections indicates that the performance of this detector is underrepresented by the 23% detection rate. In fact, if we increase our correct detection tolerance from 10 pixels to 20 pixels, the detection rate for the eigenface detector increases to 37%. Nevertheless, it is clear that both the neural network and the fused detectors significantly outperform the eigenface-based detector.

The RMSE of the neural network based face detector for the incorrectly detected faces is also very small, which again tells us that the neural network based face detector is most likely detecting some elements of those faces but not very accurately. Fine tuning the results of the neural network based detector or using a more carefully selected training set will most likely improve the results of this detector. Regardless, our goal was to show that the fusion of simple detectors does comparatively well as compared to current state-of-the-art face detection algorithms. The results of this experiment support this claim.

6 Optimizations

The fusion of multiple detectors not only can improve the reliability and accuracy of the face detector but it can also improve its efficiency. By performing simple checks to test the validity of each test patch, further and more complicated computations can be avoided for non-face patches.

By running the face detectors on a variety of faces, it was determined that for correct face patches the following conditions are almost always met:

1. $F_{\text{skin}}(.) > 0.65$
2. $F_{\text{TR}}(.) > 0.5$
3. $F_{\text{ELTS}}(.) > 0.4$
4. $F_{\text{ETS}}(.) > 0.4$
5. $F_{\text{LTS}}(.) > 0.4$

As a result, at every point of the computation if the appropriate parameter did not surpass its corresponding condition, further computations on the current face box were skipped. This was done for a selective subset of the conditions as well as for all conditions combined, as shown in the table below. The timing data below is based on a GNU C implementation of the face detection algorithm running on an Intel P4 2.2GHz processor with 1GB RAM. All images were resized to a width of 100 prior to performing face detection.

Optimization	Average Execution Per Face Time	% improvement
None	12.17 s	0%
$F_{\text{skin}}(.) > 0.65$	5.65 s	54%
$F_{\text{TR}}(.) > 0.5$	10.42 s	14%
$F_{\text{ELTS}}(.) > 0.4$ $F_{\text{ETS}}(.) > 0.4$ $F_{\text{LTS}}(.) > 0.4$	11.25 s	8%
All of the above conditions	4.92 s	60%

Table 2 – Performance of the fused face detector with various continuation conditions which reduce the detection time.

By avoided a costly search in regions where the likelihood of a face is small, the conditions above increase the speed of the face detection process by 60%.

With further optimizations, including searching every 2-4 (instead of 1) pixels depending on the box size, the average execution time per image becomes slightly more than 1.5s, which is in the range of acceptability for live

web applications. We are currently working on an FPGA (Field Programmable Gate Array) implementation of the above algorithm, similar to our previous implementation described in [1]. It is estimated that the algorithm described in this paper running on a single state-of-the-art FPGA will be able to handle 1000 face detection requests per second.

7 Experiments with Other Image Databases

In order to compare the proposed face detector with other detectors reported in the literature, several experiments were performed using the Caltech Frontal Face Database [15]. This database consists of 450 color images of 27 individuals in various lighting conditions.

The fused detector correctly detected 404 of the 450 images, again without any training or preprocessing, which corresponds to a 90% detection rate.

Since many of the detection errors were due to poor lighting conditions, or the result of faces that were smaller than the smallest search box, the following 24 ill-conditioned images were removed from the dataset and a second experiment involving the remaining 426 images was performed.



Figure 6 – A subset of the 24 ill-conditioned images from the Caltech database for which the fused face detector fails. The six faces in the bottom left are smaller than the minimum face box, and the remainder of the faces fail to pass the $F_{\text{skin}}(.) > 0.65$ condition.

Since all of the ill-conditioned images were incorrectly detected by the fused detector, the new detection rate was 404 out of 426 images, which corresponds to a 95% detection rate.

The convolutional neural network face detector had a 94% (i.e. 421/450) detection rate on the Caltech database, which implies that this method was less affected by the ill-conditioned images and the smaller face sizes.

Other state-of-the-art face detection algorithms tested on the same (Caltech) database also performed similar or slightly better than the fused detector [13,14].

For example, the face detector of [13] utilized 103 training images and was tested on 386 of the Caltech face images, resulting in a 97% detection rate. It is unknown what this detection rate would have been had it been tested on the entire 450 image dataset. The system of [14], which utilized genetic algorithms for face detection, was pre-trained and resulted in a 90% detection rate on the Caltech database.

8 Conclusion

The proposed face detection method illustrates the effectiveness of fusing or combining multiple detectors. The performance of this fused detector outperforms all sub-detectors and performs, without any training, similarly to current state-of-the-art face detection algorithms which utilize extensive training. It should also be mentioned that the fused face detector is built from extremely simple image processing blocks which are very well suited for custom hardware implementation, which is our current research focus. The current face detector is currently running live on the *modiface.com* website making it easily and directly available for testing and experimentation.

References

- [1] Nguyen, D., Halupka, D., Aarabi, P., Sheikholeslami, A., Real-time Face Localization Using Field Programmable Gate Arrays, IEEE Transactions on Systems, Man, and Cybernetics, Part B, Vol. 36, No. 4, pp. 902-912, August 2006.
- [2] Rabi, A., Aarabi, P., Face Fusion: An Automatic Method For Virtual Plastic Surgery, Proceedings of the 2006 International Conference on Information Fusion (Fusion'06), Florence, Italy, July 2006.
- [3] Aarabi, P., Mungamuru, B., The Fusion of Visual Lip Movements and Mixed Speech Signals for Robust Speech Separation. Information Fusion (Special Issue on Robust Speech Processing), Vol. 5, No. 2, pp. 103-117, June 2004.
- [4] Fang, W. M., Aarabi, P., Robust Real-Time Audiovisual Face Localization, Proceedings of the 2004 SPIE Conference on Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications, Orlando, April 2004.
- [5] Viola and Jones, Robust Real-Time Face Detection, International Journal of Computer Vision, 2004.
- [6] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, Gradient-Based Learning Applied to Document Recognition, Proc. IEEE, vol. 86, no. 11, pp. 2278-2324, 1998
- [7] Phansalkar, VV and Sastry, PS (1994) Analysis of the Back-Propagation Algorithm with Momentum, IEEE Transactions on Neural Networks 5(3):pp. 505-506.
- [8] Peer, P., Kovac, J., and Solina, F., Human skin colour clustering for face detection, EUROCON 2003.
- [9] Vezhnevets V., Sazonov V., Andreeva A., A Survey on Pixel-Based Skin Color Detection Techniques, Proc. Graphicon-2003, pp. 85-92, Moscow, Russia, September 2003.
- [10] Vaillant, R., Monrocq, C., and LeCun, Y., Original approach for the localization of objects in images, IEE Proceedings on Vision, Image, and Signal Processing, vol. 141, pp.245-250, 1994.
- [11] Turk, M. and Pentland, A. (1991). Eigenfaces for recognition. *The Journal of Cognitive Neuroscience*, 3(1): 71-86.
- [12] R. L. Hsu, M. A. Mottaleb, and A. K. Jain. Face detection in color images, IEEE Transactions on Pattern Analysis and Machine Intelligence, 24:696-706, 2002.
- [13] Claudio A. Perez and Juan I. Vallejos, Template generation by component maximization for real time face detection, Optomechatronic Sensors, Instrumentation, and Computer-Vision Systems, Yasuhiro Takaya, Jonathan Kofman, Editors, Proceedings of SPIEm Volume 6375, Nov. 2006.
- [14] Cheng-Jian Lin, Ho-Chin Chuang, and Yong-Ji Xu, Face detection in color images using efficient genetic algorithms, Optical Engineering, Volume 45, Issue 4, April 2006.
- [15] Weber, M., Frontal Face Dataset, California Institute of Technology, <http://www.vision.caltech.edu/html-files/archive.html>, 1999.