FPGA Implementation of Face Detection using Skin Segmentation and AdaBoost Algorithm

Vinay P[1],

[1]PG Student, Department of Electronics and Communication,
East Point College of Engineering and Technology
Bidarahalli, Bangalore, Karnataka, India

***Abstract*—The following paper proposes an improved face detection engine which uses skin colour information. As the skin colour is a fundamental feature of human faces, skin colour detection can be performed on the input image to reduce the computational complexity[4]. If skin colour is not detected, then there are no chances of a face in the image. Thus relaxing any further processing. This design proposal will form a powerful and efficient face detection for mobile devices which can work in real time. This implementation works for different skin colours or tones. Following skin detection, the face is detected using adaboost algorithm. This becomes an accurate technique to avoid being cheated by fake faces., or even other face like objects. The hardware architecture for real-time processing will be implanted on a FPGA. FPGA makes an ideal use, because it provides good performance with low power dissipation at marginally low cost. All of which is essential for a power constraint mobile units.**

***Keywords*— Adaboost, Skin Segmentation, Haar features, Weak Classifier, Strong Classifier.**

I. INTRODUCTION

Face detection technology is an important part of everyday technology. It's applications tick the wide areas of biometrics, authentication, photography, entertainment etc. The biometric technology measures and analyses the human characteristics, such as eye and iris, voice, facial details, fingerprints for authentication. Among these, face detection systems carry out a major role in biometric authentication. Face detection and recognition forms an important tool in the field of computer vision system. It is used for object tracking and pattern detection. Numerous different techniques have been proposed and developed contributing to more efficient face detection and recognition systems. These face detection systems are usually used in places where there is a primary requirement high security, such paces include government offices, banks, and research centres, organizations. When we consider robotics and artificial intelligence, face detection for two- or three- dimensional implementation can be seen. Such kind of implementation is used for access control systems, high end digital cameras and advanced automotive systems.

This paper proposes a system for face detection using Adaboost algorithm. This system also uses the skin colour information. However, colour is not a physical phenomenon. Colour varies with light, illumination and environment. Thus face detection is known to be highly influenced by these conditions. The proposed face-detection engine renders high performance face detection rate by producing reliable and optimized learning data through the Adaboost learning algorithm. In order to reduce false detection by a non face or other face like objects, skin colour and depth data should be considered. However the depth data can be omitted for being more complex. Hence we can opt for skin tone detection.

II. FACE EXTRACTION

The process of face detection begins by segmenting the skin area in the given input image. The possible face regions are then detected by the adaboost algorithm.

A. Skin Segmentation

The process of skin segmentation in colour images is implemented as a pre-processing step in several applications. The skin segmentation filters out the non-skin colour part from the given input colour image [3]. The face detection system extracts human faces. Since skin colour being a fundamental trait of the humans, the skin segmentation is used a preliminary process in several image processing applications.

The input image is in RGB colour space. However RGB colour space is not suitable for skin colour models [5] because: (1) RGB components are subject to lighting conditions.
(2) High correlation between the three components.

Hence an alternative colour space, YCbCr is used. Y gives the luminance and Cb and Cr gives the chrominance of blue and red respectively. The chrominance components are independent of luminance component in the space. Hence it makes suitable for skin detection.

The following equations are used to convert RGB components into YCbCr components [6].

$$Y = 0.257R + 0.504G + 0.098B+16$$
$$Cb = -0.148R - 0.291G + 0.439B+128$$
$$Cr = 0.439R - 0.368G - 0.071B+128$$

Each pixel of an image will have different values of Y,Cb and Cr, and that pixel will be said to be a skin pixel if it satisfies the skin threshold conditions for YCbCr space. The Skin segmentation is show in Fig. 1.

(a) (b) (c)

Fig. 1 boy.jpg image in (a) RGB Colourspace (b) Skin segmented binary image (c) Skin area converted to grayscale.

B. Haar Features

On face detection process, images are classified based on values of sample features rather than pixels. These feature values are calculated using haar features, originally given by given by Viola & Jones. This technique quickly rejects the regions which are highly unlikely to contain the face. The different types of features are shown in Fig. 2. [7]

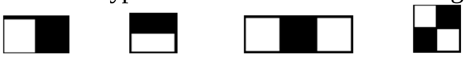


Fig. 2 Different types of haar features.

The haar feature value can be calculated by subtracting the sum of pixels in black region from sum of pixels in the white region. The feature yields a single valued result. Considering different sizes and position there can be as many as 160000+ features.

Viola & Jones suggested that feature detector of base resolution 24*24 gives satisfactory results.[1]

C. Integral Image

The feature values for different sizes can be rapidly calculated using an intermediate form of the input image. This is called integral image. The integral image at location (x, y) contains the sum of pixels above and to left of (x,y), including (x,y), show in Fig 3.1[1]

1	1	1
1	1	1
1	1	1

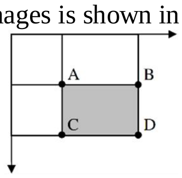
(a)

1	2	3
2	4	6
3	6	9

(b)

Fig 3.1 (a) Input Image (b)Integral Image

This allows for the calculation of sum of pixels inside given feature rectangle using only four values [1]. The integral images is shown in Fig. 3.



Sum of gray rectangle=D-B-C+A

Fig. 3 Sum calculation from Integral image

III TRAINING ADABOOST

As mentioned earlier, there can be as many as 160000+ features [1]. Among these, only few features are relevant and only few gives optimum results. These relevant features are selected by Adaboost learning algorithm.

A. Adaboost Algorithm

Adaboost is short form for Adaptive Boosting, and it is used in conjunction with other learning algorithms to improve the performance. Formulated by Yoav and Robert Schapire, the adaboost is now a widely used machine learning algorithm[2].

Adaboost constructs a collection of weak classification functions which is called as a Strong Classifier. Each feature is considered to be a potential weak classifier which correctly classifies in little more than half of the cases[2].

The mathematical representation of a weak classifier is

$$h(x, f, p, t) = \begin{cases} 1 \text{ if } pf(x) > pt \\ 0 \text{ otherwise} \end{cases}$$

Where, f is applied feature, p is parity and t is threshold which tells whether x should be classified into a face or non-face.

The best feature is selected by evaluating the feature on a set of training images. This process determines the best feature, threshold and parity based on weighed error it produces. Weighed error is a function of weights of each training images. The weight of correctly classified image is decreased while the weight of misclassified images unchanged.

The pseudo code of Adaboost algorithm for face detection is shown in Fig. 4. [1]

1. Given example images $(x_i,y_i),\ldots,(x_n,y_n)$, $y_i \in \{+1,-1\}$ for face and nonface example images respectively.
2. Initialize weights, $w_i(t)=\frac{1}{n}$
3. For t=1…T
   A. Normalize weights,
      $w_i(t)=\frac{wi(t)}{sum(wi(t))}$
   B. Select the best weak classifier with respect to weighed error:
      $h_t=\min_{f,p,t}\sum w_i|h(f,x,p,t)-y_i|$
   C. Choose $h_t(x)$, with lowest error ε_t.
   D. Update weights:
      $w_i(t+1)= w_i(t)*\beta^{1-e_i}$
      where, e_i=0, if x is classified correctly else 1 and $\beta_t=\frac{\varepsilon t}{(1-\varepsilon t)}$
      $\alpha=0.5*\log(\frac{1}{\beta t})$
4. The final strong classifier is,
   $H(x)=\sum_{t=1}^{T}\alpha_t h_t(x)$

Fig. 4 Pseudo code for Adaboost Algorithm

B. Cascade of Classifiers

The usual approach for face detection is to run the detector on the given image window. However running all the detector for the same window is not sensible as it gives negative results if the existence of face is not likely. Hence, instead of detecting faces in the image, the algorithm should discard the non-faces. This can be accomplished with the help of cascade of classifiers [1]. The idea of cascaded classifiers is illustrated in Fig. 5.

In the tree decision diagram show in Fig.7, each stage may contain 5 to 15 detectors. The image will be discarded if any detectors results a 'false' at any stage. This approach will quickly discard the non-faces without even having to run all the detectors. It also improves the accuracy by reducing false detection.

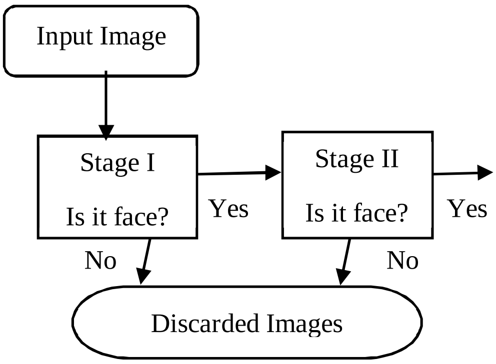


Fig. 5 Cascade of Classifiers.

IV. IMPLEMENTATION & EXPERIMENTAL RESULTS

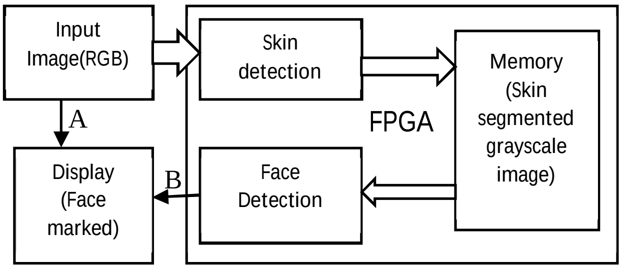
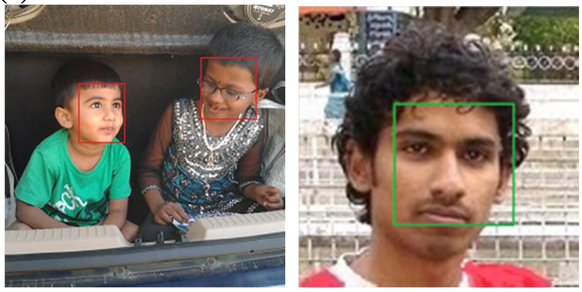


Fig. 6 Block diagram of FPGA implementation.

A. Display the original image using VGA interface.
B. Number of faces detected and their location in the image.

As the first step, the Adaboost algorithm is trained with 100 face images and 400 non face images. All the training images are taken in size of 24x24 pixels. The haar features were generated based on these images. The processing of skin segmentation followed by face detection was found to be fast. While considering accuracy and performance, the detection result are found to be satisfactory. Elimination of major portion of non-face regions by skin segmentation reduces complexity on the adaboost algorithm. The image after face detection is shown in Fig. 7(a).

In another experiment using spartan3, a 128x128 input image is taken. The experimental block diagram is shown in Fig.6.The skin detection is performed to obtain an image similar to Fig.1(c). This image is stored in an intermediate memory and it will be accessed for the face detection by haar features and adaboost. The result of this system is number faces detected in the image and their x,y coordinates in the input image. Hence the original input image is displayed on the monitor using VGA port. With the help of result from the face detection system, the faces will be marked during the display process. The resultant displayed image is shown in Fig.7(b)



(a) (b)

Fig. 7 (a) Face detection showing 2 faces marked in red rectangle
(b) Face detected image from the Spartan3 FPGA.

V. CONCLUSIONS

The proposed approach is found to be reasonably accurate given its simplicity. The skin colour segmentation reducing the adaboost complexity, the face detection system becomes more robust and efficient.

With satisfactory results on hand, the proposed approach can be adopted to simplify the process of face recognition, face expression detection or any other object detection.

ACKNOWLEDGMENT

The work has been carried out in United technologies Ltd located in Bangalore on behalf of East Point College of Engineering and Technology. I express my sincere thanks to Mrs. Padmini M for being my mentor and for all the support. I am thankful to United Technologies Ltd. for letting me to carry out the work.

REFERENCES

[1] P.Viola, M.J.Jones, " Robust Real-Time Face Detection", International Journal of Computer Vision, Vol57, N02, 37-154, pp.1 2004.
[2] R. Schapire and Y. Singer. "Improving boosting algorithms using confidence-rated predictions",1999.
[3] Sanjay Kr. Singh, D. S. Chauhan, Mayank Vatsa, Richa singh, "A Robust Skin Colour Based Face Detection Algorithm", TKJSE, Vol. 6, No. 4, pp. 227-234 (2003).
[4] Cheol Hun Han and Kwee-Bo Sim, "Real-Time Face Detection Using AdaBoot Algorithm", International Conference on Control, Automation and Systems 2008
[5] Alajel, K..M., Xiang, W., and Leis, J. Face detection based on skin colour modeling and modified Hausdorff distance. In Proc. of IEEE Int. Conf. on Consumer Communications and Networking, 2011, 399-404.
[6] Prathibha. E, Dr. Siva Yellampalli and Prof. A. Manjunath, "Design and Implementation of Colour Conversion Module RGB to YCbCr and Vice Versa", IJCSI, ICVCI-2011, Vol. 1, Issue 1, November 2011.
[7] Rainer Lienhart and Jochen Maydt, "An extended set of Haar-like features for rapid object detection," in IEEE International Conference on Image Processing, 2002, vol. 1, pp. 900–903.