ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 9, September 2015

Raspberry Pi Based Human Face Detection

Swathi .V[1], Steven Fernandes[2]

Department of Electronics & Communication, SDIT, Mangalore, India[1, 2]

Abstract: Face detection is concerned with finding whether or not there are any faces in a given image and, if present, returns the image location and content of each face. Most face detection algorithms are designed in the software domain and have a high detection rate, but they often require several seconds to detect faces in a single image, a processing speed that is insufficient for real-time applications. This paper describes a simple and easy hardware implementation of face detection system using Raspberry Pi, which itself is a minicomputer of a credit card size and is of a very low price. The system is programmed using Python programming language. Both real time face detection and face detection from specific images, i.e. Object Recognition, is carried out and the proposed system is tested across various standard face databases, with and without noise and blurring effects. Efficiency of the system is analyzed by calculating the Face detection rate for each of the database. The results reveal that the proposed system can be used for face detection even from poor quality images and shows excellent performance efficiency.

Keywords: Face detection, Raspberry Pi, Histogram Equalization, Python.

I. INTRODUCTION

Given an arbitrary image, the purpose of a face detection system is to determine if that image contains any faces. As computer science develops, face detection is becoming common place in many applications, such as face recognition, face tracking, facial feature detection, video surveillance, human computer interfaces, and robotics. The problem of face detection is challenging owing to textual differences among the faces, pose, facial expressions, orientation, facial size, lighting conditions, gender, different skin tones and changes in background. Scene changes can also be detrimental to face detection since a background can be simple as well as complex. Faces are not uniform in size and vary with the subject's distance from the camera. Owing to these challenges, researchers are striving to improve face detection by proposing new and more robust algorithms. Many a times, an effective face detection system should be coupled with noise filtering techniques. There are four different types of noise that could be generated in the images: Gaussian noise, Poisson noise, Salt and pepper noise and Speckle noise.

Most face detection algorithms are designed in the software domain and have a high detection rate, but they often require several seconds to detect faces in a single image, a processing speed that is insufficient for real-time applications [1]. Image face detection methods generally include four categories [2]. The knowledge-based methods use human knowledge to derive the rules for identifying a face; these rules are usually based on the relationships between facial features [3]. Face detection applications also use color information, and this has been proved very successful [4]. Template-matching methods use a predefined pattern, which is usually a frontal face. This type of algorithm computes the correlation values of facial characteristics, such as eyes and nose, by assessing patterns to determine the appearance of faces [5]

Appearance-based methods use statistical analysis and machine learningto collect information from a training set. Due to the complexity of the mathematical models involved, these methods usually use dimension reduction to improve computation performance [6]. Gao et al. [7] have proposed an algorithm that uses are configurable fabric to accelerate a field programmable gate array (FPGA) based Haar classifier for face detection. Using a highly pipelined architecture and utilising abundant parallel arithmetic units in an FPGA, the authors have managed to achieve real-time performance of face detection. However, the authors have stressed on the processing speed instead of detection precision.

Previous studies also propose suitable hardware implementations, such as the use of FPGAs, microcontrollers, and multiprocessor platforms [8]. These hardware implementations achieve better performance, but usually at the cost of a large area. Earlier hardware architectures focus on improving detection speed. A boosted cascade face detection architecture reduces the memory access time [9], but requires an embedded system, sacrificing area and speed. A versatile recognition processor employing a boosted cascade classifier is presented in [10]. This design achieves low power for multiple applications, but requires a huge amount of SRAM to store the input data, integral images, and classifier.

In this paper, we come up with a new hardware system for human face detection which makes use of Raspberry Pi. It is a credit-card sized computer with the components mounted on a credit card sized motherboard, running a dedicated version of Linux. It plugs into TV and a keyboard. It is a capable little computer which can be used in electronic devices and for much functionality that a desktop computer can perform. It comes at a very low price.

II. PROPOSED SYSTEM

The proposed system consists of both hardware units and software. A general block diagram of the system is as shown below.

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 9, September 2015

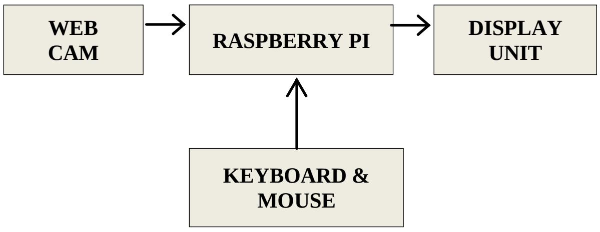

Fig.1. Block diagram of the proposed system

The Raspberry Pi is the heart of the system. We make use of a Model B Raspberry Pi which has a size specification as 85.60 mm × 53.98 mm (3.370 in ×2.125 in), and around 15 mm deep. It has a 512 MB built in RAM and operates at 700MHz. It has 2 USB ports and an Ethernet port.

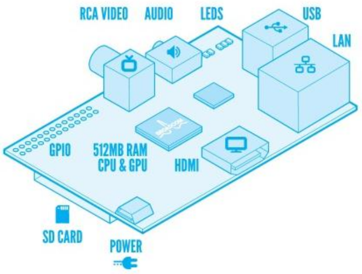

Fig. 2.Raspberry Pi Model B

The system is programmed using Python programming language.We have developed three algorithms, for face detection from a given image, from a folder of images and for real time face detection.

A. Face detection from a given image

Histogram equalization is done on the input image. Haar classifier is used for image calculation process and once face is detected, a red bounding box is drawn on the detected face. Detected face and sub faces are saved and time taken for detection is printed.

B. Face detection from a folder of images

After Histogram equalization of the given image, Haar classifier is again used for image calculation process. The difference from the first algorithm is that in addition to saving the detected face to a specified folder, the algorithm also checks if each image belongs to the source directory. If yes, the current file is named as a valid image with the file name. Otherwise, the file is named as an invalid image.

C. Real time face detection

Video is captured real time using the webcam. As long as a face is detected, a red bounding box is drawn and the video is displayed in the output window. The algorithm is efficient enough to detect multiple faces also.

III.RESULTS

Python codes were developed for face detection from a given image, from a folder of images and for real time face detection.

Sample input image and the detected image and sub faces are shown in Fig.3 below.

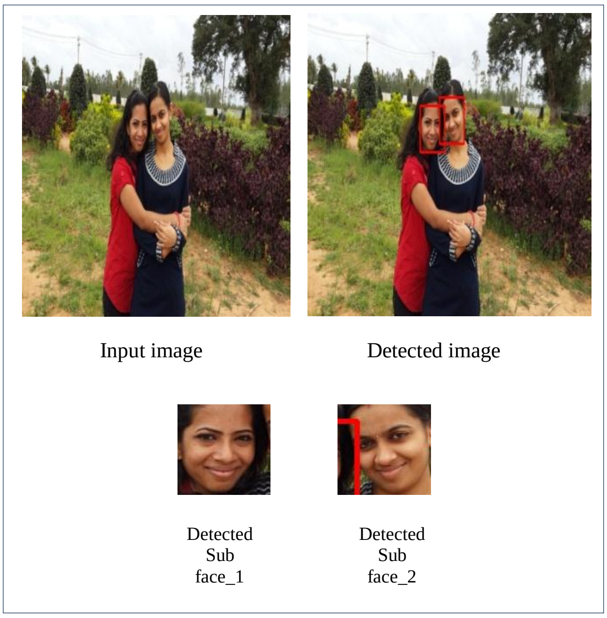

Fig. 3.Face detection from a given image

For face detection from a folder of images, the proposed face detection method was applied to various face datasets, namely Face 94, Face 95, Face 96, ATT Face database, JAFFE, Georgia tech based database, Grimace, IITK, Caltech database, Yale face database B, FGNET, Brazilian FEI database, University of Bern images.

Two types of noises, namely Median noise and Intensity noise were introduced to all the face datasets. Further, two types of Blurring effects, namely Motion blur and Gaussian blur were added to the face datasets. Face detection rate was calculated for all the face datasets, with and without noise and blurring effects. Values are tabulated in Table I and Table II.

Sample set of input images and detected images using the proposed algorithm, for JAFFE database that has major facial expression variation is shown in Fig. 4.

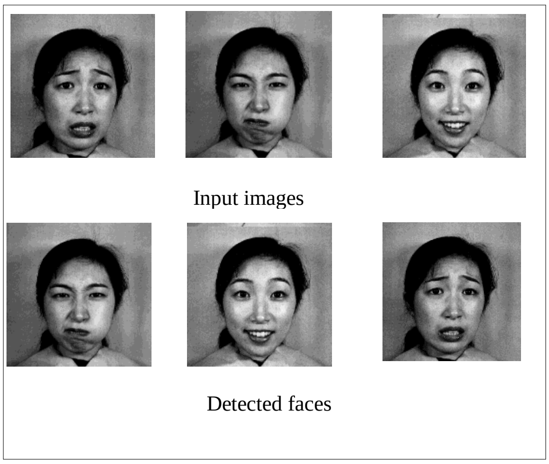

Fig.4. Face detection from a folder of images

Comparison of performance of proposed algorithm on various standard face datasets in the absence of noise and blurring effects is shown in Fig. 5.

DOI 10.17148/IJARCCE.2015.4942

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 9, September 2015

TABLE I FACE DETECTION IN THE ABSENCE OF NOISE & BLURRING EFFECTS

Database	Total no. of images	No. of faces detected	% Detection
FACE 94	3044	3023	99.31%
FACE 95	1439	1414	98.26%
FACE 96	2028	2014	99.30%
JAFFE	213	213	100.00%
GEORGIA TECH BASED DB	750	715	95.34%
GRIMACE	360	355	98.61%
IITK	677	553	81.68%
CALTECH IMAGE DB	450	449	99.78%
ATT	400	191	47.75%
SHEFFIELD	1012	351	34.68%
BRAZILLIAN FEI DB	400	393	98.25%
MUCT FACE DB	3755	3667	97.66%
POINITING HEAD POSE DB	2604	827	31.76%
POSEVARIATION DB	2620	1418	54.12%
SUBJECT	740	356	48.10%
UNIVERSITY OF BERN IMAGES	450	281	62.44%
FGNET	1002	896	89.42%

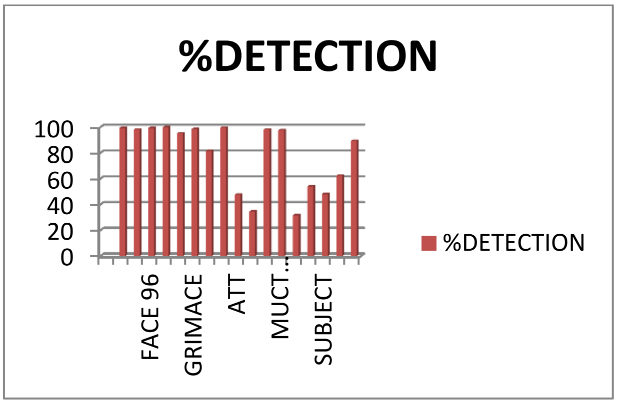

Fig.5. Face detection rate in the absence of noise & blurring effects

Sample input image and detected face for IITK database in the presence of Intensity noise is shown in Fig. 6.

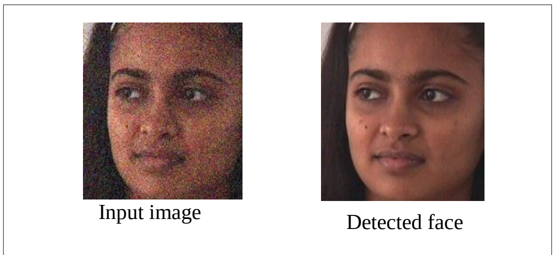

Fig.6. Face detection for image from IITK database in the presence of Intensity noise

Sample input image and detected face for JAFFE database in the presence of Median noise is shown in Fig. 7.

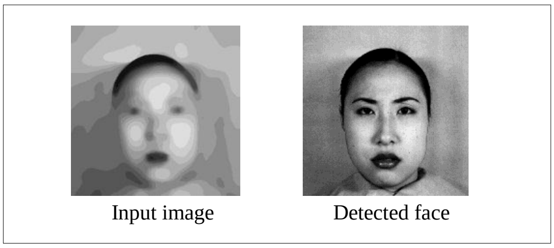

Fig.7. Face detection for image from JAFFE database in the presence of Median noise

Sample input image and detected face for CALTECH database in the presence of Gaussian blur is shown in Fig.8.

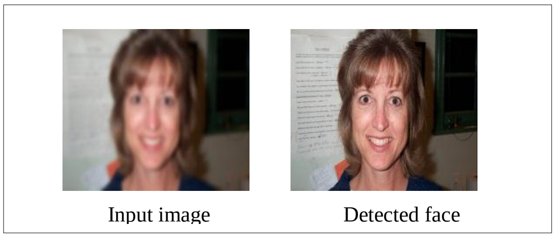

Fig.8. Face detection for image from CALTECH database in the presence of Gaussian blur

Sample input image and detected face for IITK database in the presence of Motion blur is shown in Fig. 9.

Fig.9. Face detection for image from IITK database in the presence of Motion blur

TABLE II FACE DETECTION IN THE PRESENCE OF NOISE & BLURRING EFFECTS

DATA-BASE	Intensity Noise	Median Noise	Gaussian Blur	Motion Blur
ATT	70/200 =35%	17/200 =9%	57/200 =29%	55/200 =28%
GRIMACE	52/72 =72 %	59/72 =82%	59/72 =82%	64/72 =89%
JAFFE	40/40 =100%	40/40 =100%	40/40 =100%	40/40 =100%
IITK	114/150 =76%	103/150 =69%	100/150 =67%	103/150 =69%
CALTECH	74/76 =97%	71/76 =93%	74/76 =97%	70/76 =92%
FACE_95	358/360 =99%	352/360 =98%	340/360 =94%	357/360=97%
FEI_1	341/500 =68%	403/500 =81%	341/500 =68%	440/500= 88%
FEI_2	426/500= 85%	421/500 = 84%	365/500 = 73%	459/500= 92%

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 9, September 2015

Comparison of performance of proposed algorithm on various standard face datasets in the presence of noise and blurring effects is shown in Fig. 10.

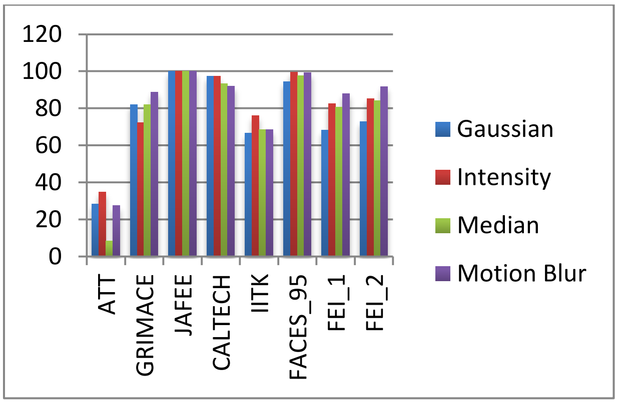

Fig.10. Face detection rate in the presence of noise & blurring effects

For Real time face detection, webcam was connected to the Raspberry Pi to capture the real time video of people in front of it. Output video window was set to defined size and the video was displayed in it. The system is efficient to capture video of multiple faces and a red box was drawn across each of the detected face.

An example of real time face detection is shown in Fig.11.

Fig.11. An example of Real time Face detection

IV. CONCLUSION

A face detection system using Raspberry Pi was developed. The system was programmed using Python programming language. Both Real time face detection and face detection from specific images, i.e object recognition, was carried out. The proposed system was tested across various standard face databases such as At & T, Caltech, Indian Face Database, JAFFE, YALE B, Face94, Face95, Face96, Grimace etc with and without noise and blurring effects. The efficiency of the system was analyzed in terms of face detection rate. The analysis revealed that the present system shows excellent performance efficiency and can be used for face detection even from poor quality images.

V. ACKNOWLEDGMENT

I wish to record my profound and sincere gratitude to our Principal and management of SDIT, Mangalore. I also express my sincere thanks to my family for extending all the support to carry out this research.

REFERENCES

[1] Chih- Rung Chen et al., "A 0.64mm^2 Real-Time Cascade Face Detection Design Based on Reduced Two-Field Extraction," IEEE Trans. VLSI Systems., vol. 19, no. 11, Nov. 2011, pp 1937-1948
[2] M. H. Yang, D. J. Kriegman, and N. Ahuja, "Detecting faces in images: A survey," IEEE Trans. Pattern Anal. Mach. Intell., vol. 24, no.1, , Jan. 2002, pp. 34–58.
[3] G. Yangand, T. S. Huang, "Human face detection in complex background," Pattern Recog., vol. 27, no. 1, 1994, pp. 53–63
[4] K. C. Yowand, R. Cipolla, "Feature-based human face detection," Image Vision Comput., vol.15, no.9, 1997, pp.713–735.
[5] T. Sakai, M. Nagao, and S. Fuji bayashi, "Line extraction and pattern detection in a photograph," Pattern Recog., vol.1, 1969, pp.233–248
[6] S. C. Yan, J. Z. Liu, X. Tang, and T. S. Huang, "Formulating face verification with semi definite programming," IEEE Trans. Image Process., vol.16, no.11, pp.2802–2810, Nov.2007
[7] B. Moghaddam and A. Pentland, "Probabilistic visual learning for object recognition," IEEE Trans. Pattern Anal. Mach. Intell., vol.19, no.7, pp.696–710, Jul.1997.
[8] Phung. S., Bouzerdoum. A., Chai. D "A novel skin color model in Y Cb Cr color space and its application to human face detection". Proc. IEEE Int. Conf. on Image Processing, 2002, vol.1, pp.289–292.
[9] Menser. B., Wien.M, "Segmentation and tracking of facial regions in color image sequences". Proc. SPIE, 2000, vol.4067, pp.731–740.
[10] Chen.W.,Sun.T., Yang. X., Wang. L., "Face detection based on half face-template". Ninth Int. Conf. on Electronic Measurement Instruments, August 2009, pp.4–54–4–58.