Low Power Affordable and Efficient Face Detection in the Presence of Various Noises and Blurring Effects on a Single-Board Computer

Steven Lawrence Fernandes and Josemin G. Bala

Dept. Electronics & Communication Engineering, Karunya University,
Tamil Nadu, India
steva_fernandes@yahoo.com, josemin@karunya.edu

Abstract. Till today face detection is a burning topic for the researchers. In the areas like digital media, intelligent user interface, intelligent visual surveillance and interactive games. Various noises and blurring effects face images captured in real time. Single board computer for efficient face detection system is introduced in this paper which works well in the presence of Gaussian Noise, Salt & Pepper Noise, Motion Blur and Gaussian Blur. Raspberry Pi based single-board computer is used for the experiments, because it consumes less power and is available at an affordable price. The developed system is tested by introducing varying degree of noises and blurring effects on standard public face databases: GRIMACE, JAFEE, INDIAN FACE, CALTECH, FACE 95, FEI – 1, FEI – 2. In the absence of noise and blurring effects also the system is tested using standard public face databases: GRIMACE, JAFEE, INDIAN FACE, CALTECH, FACE 95, FEI – 1, FEI – 2, HEAD POSE IMAGE, SUBJECT, and FGNET. The key advantage of the proposed system is excellent face detection rates in the presence of noises, blurring effects and also in the presence of varying facial expressions and across age progressions. Python scripts are developed for the system, resulted are shared on request.

1 Introduction

Pixel color is taken directly in conventional face detection methods as information cues. For illumination and cue changes, these collected data are very sensitive [1-2]. These problems are addressed by many researches. They introduced transform features. The transform features convert the pixel color (or intensity) with the help of nonlinear transform function. These are classified into two groups. Among them, the first one is intensity based transform feature and the second one is gradient based transform feature. First one convert pixel color (or intensity) into and encoded value. It compares pixel value and neighboring pixels. Papageorgiou and Poggio [3] found out Haar like features. In this, two rectangular regions are taken and their average intensities are calculated. The Haar like feature encodes these differences between average intensities. Irrespective of pixel color (or the intensity) the Haar like feature is applied to extraction of texture. To detect faces, Harr like features are utilized by

S.C. Satapathy et al. (eds.), *Emerging ICT for Bridging the Future – Volume 1,*
Advances in Intelligent Systems and Computing 337, DOI: 10.1007/978-3-319-13728-5_13

Viola and Jones[4], [5]. Integral images [5] is been utilized by them to calculate Haar like feature. The integral images are also used to compute efficient scheme to construct a vigorous classifier. This is achieved by cascading several impuissant classifiers which utilizes AdaBoost training. Binary Haar like feature is proposed by Yen et al.[6] which keep only directional relationship for Haar feature computation. But for only robust classification Haar feature was too impotent. Based on general definition texture in local neighborhood Ojala et al [7] proposed local binary pattern(LBP). But in LBP, since there are many different patterns of local intensity, the sensitivity variation makes the training process more tedious in case of AdaBoost. Local gradient Patterns (LGP) are used to overcome this problem. It produces constant patterns, even though local intensity variations are present along edges. Similar to LBP is Cnesus Transform (CT), introduced by Zabin and Woodfill [8 -14]

In this paper, we made an effort to apply hybrid feature that combines local transform features: LBP, LGP, and BHOG by means of the AdaBoost feature selection method on single-board computer – Raspberry Pi. The main advantage of Raspberry Pi based single-board computer is that it consumes less power and it is available at an affordable price. With a total cost of less than $200, the system can be easily built; the system consumes only 3 watts of power for 17 pages/sec. In the proposed system we have used local transform feature among several local transform features specifically, LBP, LGP, and BHOG having the lowest classification error is sequentially selected until we obtain the required classification performance. The developed system is tested with different facial expressions, varying poses, across age progressions and in presence of various noises: Gaussian Noise, Salt & Pepper Noise and various blurring effects: Motion blur, Gaussian blur. Standard public database used are: GRIMACE, JAFEE, INDIAN FACE, CALTECH, FACE 95, FEI – 1, FEI – 2, HEAD POSE IMAGE, SUBJECT, FGNET.

2 Single Board Computer-RASPBERRY PI

In educational institutes, to give hands on experience, Raspberry Pi was invented by Raspberry foundation [15-30]. It is a single board computer with two models. One model consist of 256MB RAM, single USB port. It doesn't contain any network connection. The second model has 512 MB RAM and two USB port. The second model also has Ethernet port. We have used second model. It contains Broadcom BCM 2835 system on a chip. This chip includes one ARM1176JZ-F700 MHz low processor. The GPU on the board is Broadcom BCM2835 system on a chip which includes an ARM 1176JZ-176JZ-F 700 MHz low processor. The board contains 3.5mm HDMI audio output. It supports SC, MMC, SDIO card slots. Blue ray quality playback can be obtained by GPU using H.264 at 40MBits/s.

3 Local Binary Patterns (LBP)

Original LBP operators are used to label pixels along with decimal numbers. The labelled pixels are called LBPs or as LBP codes. They include other pixels around the selected pixel, encode them and form as local structure. Around the selected pixel, 8 neighbors are selected in a 3x3 matrix form. The center pixel value is deducted with the value that of neighbor pixel values. The results are verified. The pixel negative value is encoded as zero. Others are encoded as one. Now, all the encoded values are concatenated in clockwise fashion. The starting point is left top most neighbor. The result so obtained (binary number) will be taken as label of the center pixel. Like this the derived binary numbers of the region is taken as LBPs or LBP codes.

4 Local Gradient Patterns (LGP)

The eight neighbors gradient values are used by LGP operator. The average gradient value of the neighbors is calculated and are substituted to the given pixel. It is used as threshold value for LBP encoding. The neighbor pixels gradient value is checked with the current pixels gradient value. If the neighbors gradient value is greater than the current then it is assigned a value 1 or else 0 is assigned. The LBP code will be concatenation of binary 1s and 0s. This will form as binary code. This method can be utilized for different sized neighbor hoods. As an example consider a circle with radius r. Let it take p sampling points.

5 Binary Histogram of Oriented Gradients (BHOG)

Following are the steps to generate BHOG features. Take the square of the gradient magnitude. Square it. Identify the block, take the orientation of all the pixels in the block. Square it. In the same now build the orientation histogram HOG(b). Encode the orientation histogram in to 8 bit vector. Here each bit is determined by thresholding. Now if this thresholding is greater than the given threshold, assign 1 bit; else assign 0 bit. Compared to HOG, there are several advantages of BHOG. In BHOG, to calculate gradient magnitude, we need not have to calculate square root. The reason for this is, it just compares the threshold and value of histogram bin. Next is, an essential step in original HOG, normalization of orientation histogram, is not present in BHOG. Instead, it just needs relative comparison of given threshold value and value of histogram bin. By AdaBoost training, BHOG feature can be obtained. This is because the BHOG feature can be represented as scalar value of one dimensional.

6 Hybridized Local Transform Features (HLTF) for Face Detection on Single Board Computer – Raspberry Pi Model B

The HTLF algorithm is as shown below:
Step 1: Start
Step 2: Prepare the positive and negative training image
Step 3: For this positive and negative training image, calculate weight values.
Step 4: For LBP, LGP and BHOG local transform features, obtain positive, negative training feature images.

Step 5: For all feature images, calculate classification errors.
Step 6: The transform feature which has minimum classification error need to be selected as best local transform feature.
Step 7: Large weight values are given to the training images that are incorrectly classified by selected feature. Small weight values are given to the training images that are correctly classified as selected feature in subsequent iterations. This update is done for all training images.
Step 8: Share the weight values among LBP, LGP and BHOG features on Respberry PI Model B in order to prevent reselection of previously selected feature by other feature types.
Step 9: Stop.

7 Results and Discussions

We have considered 7 standard databases to test HLTF on a single board computer – Raspberry Pi. We have introduced various noises: Gaussian noise, Salt & Pepper noise to all the 6 standard databases: GRIMACE, JAFFE, INDIAN FACE, CALTECH, FACE 95, FEI – 1, FEI – 2. This process is followed by introducing blurring effect: Motion blur and Gaussian blur to same 7 standard databases. Varying degree of noises and blurring effects were added to face images using paint.net tool. The values are tabulated in TABLE 1 for GRIMACE, JAFFE, and INDIAN FACE databases and in TABLE 2 for CALTECH, FACE 95, FEI – 1, and FEI – 2 databases. The noise value will indicate the amount of noise added to the face image. Blur value indicate the amount of blurriness added to the face image.

Table 1. Details of various noises and blurring effects added to face images for GRIMACE, JAFEE and Indian face databases

DATABASE	GRIMACE	JAFFE	INDIAN FACE
Gaussian noise values	50,55,60,65,70	50,55,60,65,70	50,55,60,65,70
Salt & Pepper noise values	50,60,70,80,90	50,60,70,80,90	50,60,70,80,90
Motion blur values	10,20,30,40,50	10,20,30,40,50	10,20,30,40,50
Gaussian blur values	5,10,15,20,25	5,10,15,20,25	5,10,15,20,25

Table 2. Details of various noises and blurring effects added to face images for CALTECH, FACE 95, FEI – 1, FEI - 2 databases

DATABASE	CALTECH	FACE 95	FEI - 1	FEI - 2
Gaussian noise values	50,55,60,65,70	50,55,60,65,70	50,55,60,65,70	50,55,60,65,70
Salt & Pepper noise values	50,60,70,80,90	50,60,70,80,90	50,60,70,80,90	50,60,70,80,90
Motion blur values	10,20,30,40,50	10,20,30,40,50	10,20,30,40,50	10,20,30,40,50
Gaussian blur values	5,10,15,20,25	5,10,15,20,25	5,10,15,20,25	5,10,15,20,25

FRR = (Total number of face images recognized accurately / Total number of face images available in the database) x 100. Face detection rate obtained using Hybridized Local Transform Features (HLTF) algorithm on Single Board Computer – Raspberry PI Model B in the presence Gaussian Noise, Salt and Pepper Noise, Motion Blur, Gaussian Blur, Facial Expression variations, Pose variations and Age variations are tabulated from Table 3 to TABLE 9.

Table 3. Face detection rate in the presence of Gaussian noise

DATABASE	**Total no. of images in the database**	**Total no. of face images detected appropriately**	**Total no. of images detected as face but not faces**	**Total no. of faces not detected**	**Face detection rate**
GRIMACE	18*4=72	59	00	13	81.94
JAFFE	10*4=40	40	00	00	100
INDIAN FACE	30*5=150	111	00	39	74
CALTECH	19*4=76	74	00	02	97.36
FACE 95	72*5=360	352	00	08	97.77
FEI - 1	100*5=500	403	00	97	80.6
FEI - 2	100*5=500	421	00	79	84.2

Table 4. Face detection rate in the presence of motion blur

DATABASE	**Total no. of images in the database**	**Total no. of face images detected appropriately**	**Total no. of images detected as face but not faces**	**Total no. of faces not detected**	**Face detection rate**
GRIMACE	18*4=72	61	00	11	84.72
JAFFE	10*4=40	40	00	00	100
INDIAN FACE	30*5=150	110	00	40	73.33
CALTECH	19*4=76	74	00	02	97.36
FACE 95	72*5=360	357	00	03	99.16
FEI - 1	100*5=500	440	00	60	88
FEI - 2	100*5=500	459	00	41	91.80

Table 5. Face detection rate in the presence of Gaussian blur

DATABASE	Total no. of images in the database	Total no. of face images detected appropriately	Total no. of images detected as face but not faces	Total no. of faces not detected	Face detection rate
GRIMACE	18*4=72	59	00	13	81.94
JAFFE	10*4=40	40	00	00	100
INDIAN FACE	30*5=150	102	00	48	68
CALTECH	19*4=76	74	00	02	97.36
FACE - 95	72*5=360	340	00	20	94.44
FEI - 1	100*5=500	341	00	159	68.20
FEI - 2	100*5=500	365	00	135	73

Table 6. Face detection rate in the presence of facial expression variations

DATABASE	Total no. of images in the database	Total no. of face images detected appropriately	Total no. of images detected as face but not faces	Total no. of faces not detected	Face detection rate
FACE 94	19*20+113*20+20*20= 3040	3020	01	19	99.34
FACE 95	72*20=1440	1415	00	25	98.26
FACE 96	147*20=2940	2895	00	45	98.46
JAFFE	10*20=200	200	00	00	100
GEORGIA TECH	50*15=750	718	05	27	95.73
GRIMACE	18*20=360	352	07	01	97.77
INDIAN FACE	22*11+ 39* 6= 476	356	00	120	74.78
CALTECH	19*18=342	342	00	00	100
BRAZILLIAN FEI	100*2+100*2=400	394	00	06	98.50

Table 7. Face detection rate in the presence of pose variations

DATABASE	Total no. of images in the database	Total no. of face images detected appropriately	Total no. of images detected as face but not faces	Total no. of faces not detected	Face detecti on rate
HEAD POSE IMAGE	13*186=2418	761	00	1657	31.47
SUBJECT	10*74=740	352	10	378	47.56

Table 8. Face detection rate in the presence of age variations

DATABASE	Total no. of images in the database	Total no. of face images detected appropriately	Total no. of images detected as face but not faces	Total no. of faces not detected	Face detection rate
FGNET	1002	921	00	81	91.91

8 Conclusion

This paper introduces a single board computer - Raspberry Pi for efficient face detection system in the presence of Gaussian Noise, Salt & Pepper Noise, Motion Blur and Gaussian Blur. We apply hybrid feature that combines local transform features: LBP, LGP, BHOG by means of the AdaBoost feature selection method on single-board computer – Raspberry Pi because it consumes less power and it is available at an affordable price. The developed system is tested by introducing varying degree of noises and blurring effects on standard public face databases: GRIMACE, JAFEE, INDIAN FACE, CALTECH, FACE 95, FEI – 1, FEI – 2. The developed system is also tested in the absence of noise and blurring effects using standard public face databases: GRIMACE, JAFEE, INDIAN FACE, CALTECH, FACE 95, FEI – 1, FEI – 2, HEAD POSE IMAGE, SUBJECT, and FGNET. Thus, the key advantage of the proposed system is excellent face detection rates in the presence of noises, blurring effects and also in the presence of varying facial expressions and across age progressions

References

1. Togashi, Haruo and Kanagawa, Face recognition apparatus, face recognition method, gabor filter application apparatus, and computer program. U. S. Patent 8077932 (December 13, 2012)
2. Li, S., Gong, D., Liu, H.: Face identification method based on multiscale weber local descriptor and kernel group sparse representation. C. N. Patent 102722699A (October 10, 2010)

3. Papageorgiou, C., Poggio, T.: A Trainable System for Object Detection. Int'l J. Computer Vision 38(1), 15–33 (2000)
4. Viola, P., Jones, M.: Robust Real-Time Face Detection. Int'l J. Computer Vision 57(2), 137–154 (2004)
5. Viola, P., Jones, M., Snow, D.: Detecting Pedestrians Using Patterns of Motion and Appearance. Int'l J. Computer Vision 63(2), 153–161 (2005)
6. Yan, X.C.S., Shan, S., Gao, W.: Locally Assembled Binary (LAB) Feature with Feature-Centric Cascade for Fast and Accurate Face Detection. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 1–7 (2008)
7. Ojala, T., Pietikainen, M., Harwood, D.: A Comparative Study of Texture Measures with Classification Based on Feature Distributions. Pattern Recognition 29(1), 51–59 (1996)
8. Zabih, R., Woodfill, J.: Non-Parametric Local Transforms for Computing Visual Correspondence. In: Eklundh, J.-O. (ed.) ECCV 1994. LNCS, vol. 801, pp. 151–158. Springer, Heidelberg (1994)
9. Zhang, L., Chu, R., Xiang, S., Liao, S., Li, S.Z.: Face Detection Based on Multi-Block LBP Representation. In: Lee, S.-W., Li, S.Z. (eds.) ICB 2007. LNCS, vol. 4642, pp. 11–18. Springer, Heidelberg (2007)
10. Lowe, D.: Distinctive Image Features from Scale Invariant Keypoints. Int'l J. Computer Vision 60(2), 91–110 (2004)
11. Ke, Y., Sukthankar, R.: PCA-Sift: A More Distinctive Representation for Local Image Descriptors. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 511–517 (2004)
12. Bay, H., Ess, A., Tuytelaars, T., Gool, L.: Surf: Speeded Up Robust Features. Computer Vision and Image Understanding 110(3), 346–359 (2008)
13. Dalal, N., Triggs, B.: Histograms of Oriented Gradients for Human Detection. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 886–893 (2005)
14. Zhu, Q., Avidan, S., Yeh, M., Cheng, K.: Fast Human Detection Using a Cascade of Histograms of Oriented Gradients. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 1491–1498 (2006)
15. T. R. P. Foundation, About us, `http://www.raspberrypi.org/about` (Online; accessed June 2013)
16. Arcia-Moret, A., Pietrosemoli, E., Zennaro, M.: WhispPi: White space monitoring with Raspberry Pi. In: Global Information Infrastructure Symposium, pp. 28–31 (2013)
17. Ali, M., Vlaskamp, J., Eddin, N., Falconer, B., Oram, C.: Technical development and socioeconomic implications of the Raspberry Pi as a learning tool in developing countries. In: 5th Computer Science and Electronic Engineering Conference (2013)
18. Nagy, T., Gingl, Z.: Low-cost photoplethysmograph solutions using the Raspberry Pi. In: IEEE 14th International Symposium on Computational Intelligence and Informatics (2013)
19. Milenkovic, A.M., Markovic, I.M., Jankovic, D.S., Rajkovic, P.J.: Using of Raspberry Pi for data acquisition from biochemical analyzers. In: 11th International Conference on Telecommunication in Modern Satellite, Cable and Broadcasting Services (2013)
20. Edwards, C.: Not-so-humble raspberry pi gets big ideas. Engineering & Technology 8(3), 30–33 (2013)
21. Mitchell, G.: The Raspberry Pi single-board computer will revolutionise computer science teaching (For & Against). Engineering & Technology 7(3), 26 (2012)
22. Batur, A.U., Hayes, M.H.: Adaptive active appearance models. IEEE Trans. Image Process. 14(11), 1707–1721 (2005)
23. Edwards, C.: ICT lessons get the raspberry. Engineering & Technology 7(4), 76–78 (2012)

24. Tso, F.P., White, D.R., Jouet, S., Singer, J., Pezaros, D.P.: The Glasgow Raspberry Pi Cloud: A Scale Model for Cloud Computing Infrastructures. In: IEEE 33rd International Conference on Distributed Computing Systems Workshops, pp. 108–112 (2013)
25. Sundaram, G.S., Patibandala, B., Gaddam, S., Alla, V.K.: Bluetooth communication using a touchscreen interface with the Raspberry Pi. In: Proceedings of IEEE Southeastcon, pp. 1–4 (2013)
26. Ionescu, V.M., Smaranda, F., Diaconu, A.-V.: Control system for video advertising based on Raspberry Pi. In: RoEduNet International Conference 12th Edition Networking in Education and Research, pp. 1–4 (2013)
27. Abrahamsson, P., Helmer, S., Phaphoom, N.: Affordable and Energy-Efficient Cloud Computing Clusters: The Bolzano Raspberry Pi Cloud Cluster Experiment. In: IEEE 5th International Conference on Cloud Computing Technology and Science, pp. 170–175 (2013)
28. Jain, S., Vaibhav, A., Goyal, L.: Raspberry Pi based interactive home automation system through E-mail. In: International Conference on Optimization, Reliabilty, and Information Technology, pp. 277–280 (2014)
29. Banerjee, S., Sethia, D., Mittal, T.: Secure sensor node with Raspberry Pi. In: International Conference on Multimedia, Signal Processing and Communication Technologies, pp. 26–30 (2013)
30. Shams, E., Thiessen, J.D., Bishop, D.: A PET detector interface board and slow control system based on the Raspberry Pi, pp. 1–3 (2013)