Grenze International Journal of Engineering and Technology, Jul 2019

Analysis of Preprocessing Algorithms for Face Detection using KNN and SVM Classifiers

Sai Sasank Varthakavi [1], Devisetty Rohith Prasanna Babu[2], Likith Kumar Reddy Singam Reddy[3] and Remya Ajai AS[4]
[1-4]Department of Electronics and Communication Engineering, Amrita School of Engineering Amritapuri, Kerala, India
saisasank86@gmail.com[1], rp babu 461@gmail.com[2], singamreddy.likith@gmail.com[3] remya.amrita@gmail.com[4]

***Abstract*—Face detection using computer vision algorithm is undoubtedly an important technology in this modern world. The accuracy of the existing algorithms can be improved by analyzing the best preprocessing technique for a particular classifier. In this work, we implemented different preprocessing algorithms for two main classifiers namely KNN and SVM. With various combinations of existing preprocessing, segmentation and feature extraction algorithms, SVM classifier gives better accuracy of around 98%. The analysis is done using MATLAB 2018b software.**

***Index Terms*— Edge Detection, Feature extraction, HOG, Image Segmentation, K-NN classifier, Preprocessing, and SURF features.**

I. INTRODUCTION

Image processing is a technology which started its journey for Satellite imagery, wire photo standards conversion took a new transformation in the 21st century for various purposes such as biomedical image processing, computer vision etc. Our interest is to explain to you the application of image processing in face detection which is one of the emerging and challenging technologies of computer vision.
Face detection can be used for various purposes such as automatic attendance system, crime detection, biometrics, etc. This technology has become very challenging due to the following reasons:

- Pose change.
- Image quality.
- Facial changes.

In this paper, we discussed about various combinations of algorithms and techniques and also we discussed the accuracy of their combinations so that you can implement it in no matter of time. We also discussed different steps involved in Face detection such as Image acquisition, Image Preprocessing, Image segmentation, Feature extraction,
Image classification etc. one by one as shown in the Fig 1.

II. PREPROCESSING

Preprocessing is a technique which is used to enhance the raw images received from cameras or pictures taken from different databases which are used in various applications like surveillance in public area. In various fields of science and technology significance of image pre-processing[1] had improved in recent times. In this scenario, we mainly focussed on 3 types of preprocessors namely,

Grenze ID: 01.GIJET.5.2.10_1

- Image adjustment
- Image sharpening
- Histogram Equalisation

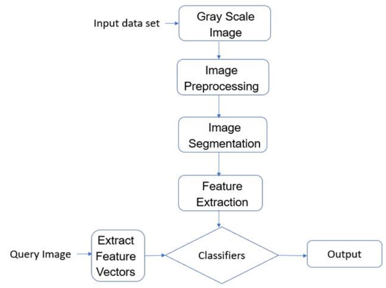

Fig 1: Block Diagram for face detection system

A. Image adjustment

In order to maintain all the images in a database at a certain intensity level, we choose this preprocessor so that it is easy for the next steps involved. Imadjust is the parameter which increases the contrast by saturating the 1% of data at low and high intensities of the image as we referred from [2]and [3].

 

Fig 2: Input image Fig 3 : Output image

Fig 3 is the result of the Matlab generated code which shows that the contrast is increased by maintaining intensity levels.

B. Image sharpening

Another preprocessor is Imsharpen which is frequently known parameter used for its own significance. This preprocessor increases the contrast of the image whenever the two colors of the image meet i.e we are dealing with a grayscale image , whenever the image change color from black to gray and gray to white then there blur will exit on the image in order to remove the blur area this preprocessor use the technique[2] and [3]called Unsharp Masking, in this situation the image get sharpened by filtering out the unsharpened area it is just like subtracting the unsharpened(blur) area from the image itself . Fig 5 is the sharpened image.

 

Fig 4 :Input image Fig 5 :Output image

C. Histogram Equalisation

Histogram Equalization is a technique which is used to increase or decrease the intensity of the image. It will calculate the pixel intensities. After that, it will count no. of pixels which have the same p

Probability of each pixel = No.of pixels with the same intensity /total no. of pixels. (1)

It will find out the cumulative probabilities of pixels. Now it will multiply the desired intensities with cumulative probabilities of the pixels. The output of this technique results in the desired intensity of the image as shown in Fig7.

 

Fig 6 :Input image Fig 7 :Output image

III. IMAGE SEGMENTATION

Image segmentation is a technique which is used to highlight certain regions of the image based on its color, intensity, texture. This highlighted area is the concentrated area which can be further moved for image analysis. Image segmentation includes several techniques [4] such as Sobel, canny, Prewitt in edge detection based segmentation, [6]thresholding, region-based segmentation methods k-means clustering, [7]Fuzzy c-Means (FCM), clustering person are several methods of segmentation. The boundary should be created where there is a maximum difference between pixel value. This can be done easily by Edge detection in image segmentation. Similarly, face can be detected and segmented accurately.

A. Edge detection

In this survey, we concentrated on Edge detection[8] because in Face detection since our point of interest is only on the facial features of a specific person. In Edge detection,[9] the image is segmented according to the intensity values of image or discontinuity of image. The portion which is different from the other portion is to be identified. It should be separated by a boundary as shown in the figure 8

Fig 8: Image showing equally divided intensity levels

In figure 8, boundary can be easily identified since there is an exact separation between area filled with white and black which is impossible in Fig 9 there comes Edge detection into picture.

Fig 9: Image showing an abrupt change in the intensity levels

Types of edge detectors:

- Sobel Edge Detector.
- Canny Edge Detector.
- Prewitt Edge Detector

B. Sobel Edge detector

Sobel Edge detector [10]is used to find the change in contrast or the gradient of contrast which is equal to change in intensity or change in state of pixel either 1 to 0 or 0 to 1. Fig 10 gives the Horizontal Gradient of the Image and Fig 11 results in the Vertical Gradient, which will helps us to calculate Orientation given as (2).

 

Fig10 :Gx - Horizontal Gradient Fig 11 :Gy - Vertical Gradient

Orientation = $tan-1$ vertical gradient /Horizontal gradient) (2)

C. Canny Edge detector

The Canny Edge detector is considered to be a good detector for already existing edges. It uses the technique of localization which means the distance between real edge pixels is minimized in output edge pixels which gives a thin edge. It filters noise [5] before detecting any edges by the Gaussian filter. Then the Gaussian smoothing will be performed after a suitable mask has been calculated. Then it uses the Sobel kernel to find the gradient in the horizontal direction(Gx) and vertical direction (Gy).

$$\text{Edge Gradient} = \sqrt{Gx^2 + Gy^2} \qquad (3)$$

$$\text{Angle}(\Theta) \quad = \quad tan - 1 \;\; Gx \div Gy \qquad (4)$$

It will take all real edges and suppress it if it is not an actual edge and sets threshold value say higher and lower.The higher value is to start an edge and the lower edge is to continue it. The result will be a binary image with thin edges

D. Prewitt Edge Detection

Prewitt Edge detector can be Viewed as an approximation of Sobel Operator. But it has its own limitations. It can be used only for noiseless and contrasted images otherwise it won't come up with desired Output rather comes out as a noisy image. In real time that might be a problem to be dealt with.

Fig 12 :Input image

Fig 13: Gradient Magnitude

Fig 14: Gradient Direction

IV. Feature extraction

Feature extraction is an approach which reduces the dimensionality of an image and is used to locate interest points of an image, as a compact feature vector. To complete tasks like image matching and retrieve more quickly, we use this approach to reduce the feature representation when the image sizes are large.
In this paper, we mainly discussed two feature extraction techniques to get combinations and to calculate accuracy.

- Histogram of oriented gradients (HOG)
- Speed up Robust features (SURF)

A. Histogram of oriented gradients (HOG)

HOG returns a feature vector of the 1xN matrix, where N is the length of the feature vector from a grayscale input image. Then it will return a converted and encrypted local shape from interest regions within the image as in the figure 16.

Fig 15 :Input image

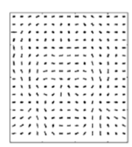

Fig 16: Output image

The HOG feature is arranged by HOG blocks.

Block 1	*Block 2*	*Block 3*	*Block 4*	*Block 5*

H(c12)	*H(c22)*	*H(c23)*	*H(c24)*

Fig 17: The Arrangement of Histogram in HOG Feature V vectors

C11	*C12*	*C13*
C21	*C22*	*C23*

Fig 18: Recognized image pixels in cell size of 2x3

When we recognize the face and if the recognized part is in cell size of 2x3, which is represented in the above diagram, then it would make the size of each HOG block, 2x2 cells. Therefore, we have a size of the cells as pixels[11]and [12].

Block-1

H(C11)	*H(C21)*	*H(C12)*	*H(C22)*

Block-2

H(C12)	*H(C11)*	*H(C13)*	*H(C23)*

As a part of our work, we will need feature vectors only and later we can use this information for classification.

A. Speed Up Robust Features (SURF)

SURF returns interest points which are having different properties in a digital image and interest points are selected from distinctive locations from the image. For example, corners and T- junction. The repeatable property of an interest points results in us find out the same interest points in different viewing conditions like image having high brightness and contrast etc., and these interest points will be taken as a feature vector as shown in Fig 20.

Fig 19: Input image

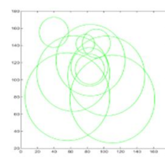

Fig 20: Output image

To detect SURF points, in this paper, first we have to filter the image by the equation (5) so that the filtered image has local points to find out these local points it will use basic approach named Hessian-matrix approximation.

$$S(x,y)= \sum_{i=0}^{x} \sum_{j=0}^{y} I\,(i,j) \qquad (5)$$

The above expression (5) is used to calculate intensities and to make sure that calculation is independent of the size of the image, The Hessian-Matrix[13][14] is given as below:

$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1\,\partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1\,\partial x_n} \\ \frac{\partial^2 f}{\partial x_2\,\partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2\,\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n\,\partial x_1} & \frac{\partial^2 f}{\partial x_n\,\partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}. \qquad (6)$$

To point out interest points in the image, we must suppress the maximum octaves [15]. The maxima of the determinant of the Hessian matrix are plotted with the method explained [16]. Finally, we acquired the interest points as feature vectors and later on, we can use this result for the classification.

V. CLASSIFICATION

So far,our discussion is about how the Feature vectors are extracted from different combinations from an Image, Now our aim is to classify the query image to which class it belongs to So we will use a classifier.

A. K-NN Classifier

To classify we can use metric distance techniques between the query test set points and dataset points. Namely K-NN classifier[17] will use this technique. We are dealing with the multiclass datasets so we might be having a problem with small cost (i.e ‘cost’ refers to the matrix namely cost(i,j) where I refers to the class in which it belongs to, and j belongs to predicted class), in order to overcome this problem K-NN classifier will break ties,

ties occur whenever we are dealing with multiclass, to classify the dataset K-NN use the K- nearest neighbor[18] and also smallest index among the tied groups. K-NN classifier has the property[19],i.e it stores the index values of the columns of the query data which has categorical predictors and gives result empty set whenever the predictors are not categorical.[20]. After the classification, in order to get the accuracy we will use the inbuilt Matlab function C=confusionmat(train_label, Group), (i.e Group is an n$\times$ 1 matrix which has predicted testset) Finally we calculated accuracy for different combinations that we have explained so far is given the table.

B. SVM Classifier

SVM classifier is also called Support Vector Machine algorithm is generally used for training supervised models in machine learning. It can be implemented for both classification and regression models of machine learning. It generally classifies the training data into two regions by creating a linear line or boundary which is well known as hyperplane. This hyperplane is unique for every model and it is constructed in a way that it reduces the maximum error and also maximize the geometric margin. Testing data label is predicted by calculating the distance between training data features and testing data features and the label which has got the minimal distance will be assigned to the testing dataset.

VI. RESULTS AND ANALYSIS

Face detection algorithm is implemented in matlab with various combinations of preprocessing, segmentation, feature extraction and classifier algorithms. The accuracy is calculated and tabulated as below:

TABLE I: ACCURACY OBTAINED USING VARIOUS PREPROCESSING AND CLASSIFIER ALGORITHMS

S.NO	Preprocessor	Segmentation	Feature Extraction	Classifier	Accuracy in %
1	Image adjustment	Sobel Edge Detection	HOG	SVM	98.4
2	Image sharpening	Sobel Edge Detection	HOG	SVM	98.4
3	Histogram Equalisation	Sobel Edge Detection	HOG	SVM	98.4
4	Image adjustment	Canny Edge Detection	HOG	SVM	98.4
5	Image sharpening	Canny Edge Detection	HOG	SVM	98.4
6	Histogram Equalisation	Canny Edge Detection	HOG	SVM	96.4
7	Image adjustment	Prewitt Edge Detection	HOG	SVM	96.4
8	Image sharpening	Prewitt Edge Detection	HOG	SVM	96.4
9	Histogram Equalisation	Prewitt Edge Detection	HOG	SVM	96.4
10	Image adjustment	Sobel Edge Detection	SURF	K-NN	94.2
11	Image sharpening	Sobel Edge Detection	SURF	K-NN	94.2
12	Histogram Equalisation	Sobel Edge Detection	SURF	K-NN	94.2
13	Image adjustment	Canny Edge Detection	SURF	K-NN	96.3
14	Image sharpening	Canny Edge Detection	SURF	K-NN	95
15	Histogram Equalisation	Canny Edge Detection	SURF	K-NN	95
16	Image adjustment	Prewitt Edge Detection	SURF	K-NN	95
17	Image sharpening	Prewitt Edge Detection	SURF	K-NN	95
18	Histogram Equalisation	Prewitt Edge Detection	SURF	K-NN	96.3

The above combinations has been assigned
L1,L2,L3,L4,L5,L6,L7,L8,L9,L10,L11,L12,L13,L14,L15,L16,L17,L18 for the convenience in figure 22.

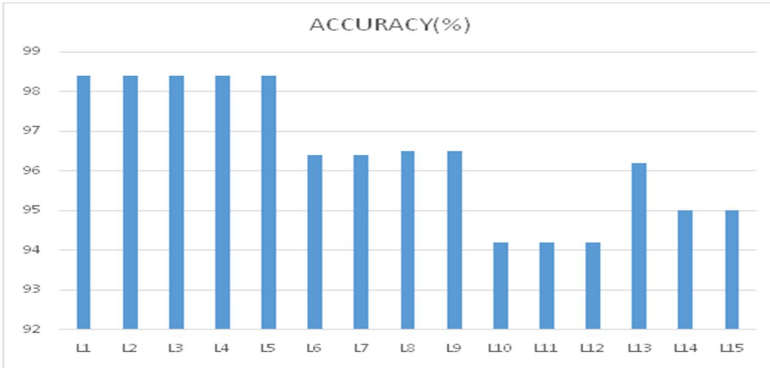

Fig 21: Graph showing different models and their accuracies

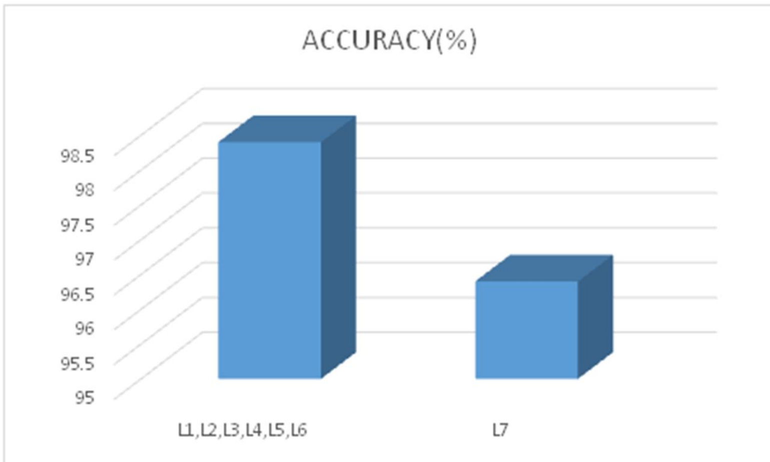

Fig. 22 Graph showing Accuracies of best fitted models

VII. Conclusion

In this paper, we explained different combinations of algorithms for face detection and accuracy was calculated using the confusion matrix. We explained different stages of face detection in a clear fashion and also in a simplified version and also we emphasized on different kinds of algorithms at each stage pictorially using flowcharts and images which we executed and tested in Matlab. From the results obtained the combination of Imadjust, Imsharpen, Histogram Equalisation, with Sobel edge detection and Canny edge detection with HOG feature extraction with SVM classifier had over beaten other combinations and stood at first with an accuracy of 98.4%. In our upcoming research paper, we will explain the different techniques used for real-time face detection which can be implemented in developing advanced intelligent systems.

References

[1] Manoj D. Chaudhary, Jitendra B. Chinchore, "Towards multiple license plate localization in Indian conditions: An edge density based approach", Control Instrumentation Communication and Computational Technologies (ICCICCT) 2014 International Conference on, pp. 60-65, 2014.

[2] Emrah Irmak, Ahmet H. Ertas, "A review of robust image enhancement algorithms and their applications", Smart Energy Grid Engineering (SEGE) 2016 IEEE, pp. 371-375, 2016.

[3] Ozan Erat, Karen Panetta, Sos Agaian, "Contrast enhancement for underwater images in maritime border protection", Technologies for Homeland Security (HST) 2017 IEEE International Symposium on, pp. 1-5, 2017.

[4] A. Sana, J. L. Perry and N. Tabrizi, "A Review of Image Segmentation Techniques for Tracking the Velum," 2015 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, 2015, pp. 433-436.doi: 10.1109/CSCI.2015.99

[5] J. Lu, H. Pan and Y. Xia, "The weld image edge-detection algorithm combined with Canny operator and mathematical morphology," Proceedings of the 32nd Chinese Control Conference, Xi'an, 2013, pp. 4467-4470

[6] R. Kohler, “A segmentation system based on thresholding,” Graph. Models Image Process., vol. 15, pp. 319 - 338, 1981

[7] R. Samet, Ş. E. Amrahov and A. H. Ziroğlu, "Fuzzy Rule-Based Image Segmentation technique for rock thin section images," 2012 3rd International Conference on Image Processing Theory, Tools and Applications (IPTA), Istanbul, 2012, pp. 402-406.doi: 10.1109/IPTA.2012.6469555

[8] A. Sharma, M. D. Ansari and R.Kumar, "A comparative study of edge detectors in digital image processing," 2017 4th International Conference on Signal Processing, Computing and Control (ISPCC), Solan, 2017, pp. 246-250.doi: 10.1109/ISPCC.2017.8269683

[9] Zhang Jin-Yu, Chen Yan and Huang Xian-Xiang, "Edge detection of images based on improved Sobel operator and genetic algorithms," 2009 International Conference on Image Analysis and Signal Processing, Taizhou, 2009, pp. 31-35.doi: 10.1109/IASP.2009.5054605.

[10] Mrityunjay Kumar Ray, D. Mitra and S. Saha, "Simplified novel method for edge detection in digital images," 2011 International Conference on Signal Processing, Communication, Computing and Networking Technologies, Thuckafay, 2011, pp. 197-202.doi: 10.1109/ICSCCN.2011.6024543

[11] T. T. Yu and N. War, "Condensed object representation with corner HOG features for object classification in outdoor scenes," 2017 18th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), Kanazawa, Japan, 2017, pp. 77-82. doi:10.1109/SNPD.2017.8022703

[12] Dalal, N. and B. Triggs. "Histograms of Oriented Gradients for Human Detection", IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 1 (June 2005), pp. 886–893.

[13] Mikolajczyk, K., Schmid, C.: An affine invariant interest point detector. In: ECCV. (2002) 128 – 142

[14] Florack, L.M.J., Haar Romeny, B.M.t., Koenderink, J.J., Viergever, M.A.: General intensity transformations and differential invariants. JMIV 4 (1994) 171–187

[15] Brown, M., Lowe, D.: Invariant features from interest point groups. In: BMVC (2002)

[16] J. M. Patel and N. C. Gamit, "A review on feature extraction techniques in Content Based Image Retrieval," 2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), Chennai, 2016, pp. 2259-2263. doi: 10.1109/WiSPNET.2016.7566544

[17] Bailey, T. & Jain, A. (1978) "A note on distance-weighted k-Nearest Neighbor rules", IEEE Trans. Systems, Man, Cybernetics, 8: 311-313

[18] Bax, E. (2000) "Validation of nearest neighbor classifiers", IEEE Trans. Inform. Theory, 46: 2746–2752

[19] Praveen, K & Poornachandran, Prabaharan & Ajai, A. (2012). Implementation of DES using pipelining concept with skew core key scheduling in secure transmission of images. ACM International Conference Proceeding Series. 637-641. 10.1145/2393216.2393322.

[20] Ajai, A & Nagaraj, Nithin. (2012). A Novel Methodology for Memory Reduction in Distributed Arithmetic Based Discrete Wavelet Transform. Procedia Engineering. 30. 226-233. 10.1016/j.proeng.2012.01.855.