International Journal of Engineering & Technology, 7 (3) (2018) 1227-1232

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET
doi: 10.14419/ijet.v7i3.12842
Research paper

Performance comparison of segmentation algorithms for hand gesture recognition

Priyanka Parvathy D [1] *, Dr. Kamalraj Subramaniam [2]

[1] Research Scholar, Karpagam Academy of Higher Education, Coimbatore, India
[2] Associate Professor, Department of ECE, Karpagam Academy of Higher Education, Coimbatore, India
**Corresponding author E-mail: priyanka_pd@yahoo.com*

Abstract

The gestures presented in diverse backgrounds have to be accurately processed and segmented, for it to be classified precisely by the hand gesture recognition system. This study compares performance of the proposed Image Segmentation Algorithm with a standard Canny Edge Detection Algorithm by comparing the statistical values of the features obtained from the feature extraction stage, thus validating the importance of having a robust preprocessing stage for the hand gestures. The proposed algorithm uses Non-local Mean filter for noise removal and then an improved Global Swarm Optimization based Canny edge detection for extracting the edges. Features are extracted using two dimensional Multi-resolution Discrete Wavelet Transform (2D-DWT) combined with Gray-level Co-occurrence Matrix. The efficiency of the proposed Image Segmentation Algorithm is evaluated using Radial Basis Function Neural Network as the classifier.

Keywords: *Hand Gestures; Preprocessing, Feature Extraction; Edge Segmentation; Non-Local Mean Filtering; Otsu Thresholding; 2d-Discrete Wavelet Transform (Dwt); Particle Swarm Optimization; Artificial Neural Network*

1. Introduction

The past few years have seen a rapid improvement in the design and development of advanced and sophisticated means of Human Computer Interaction (HCI). Though the range of HCI techniques for basic tasks is still dominated by traditional input methods like the keyboard and mouse, or touch based systems like touch pads, Hand Gesture Interaction (HGI) is gaining popularity [1] as an attractive alternative-especially in gaming, virtual reality and medical applications. With the advance of Vision based Hand Gesture Recognition (HGR), users can communicate with computers without physically touching them, just as one would communicate naturally with another. The main things which reduces the quality of HCI is size and speed variations from humans to the computer, poor performance against the complex backgrounds, varying lighting conditions and unreliable detection of gesturing phase have limited the use of hand gestures as a reliable modality in the interface design [2].The HCI interpretation of the gesture require proper preprocessing which is done by the static configurations of the hand that has been properly defined in the system [2].
The segmentation of the hand from its background is a very crucial stage in the HGI system, as the similarity of the segmented hand with that of the original hand will decide the efficiency of the recognition system. Most researches are conducted in a manner so as to simplify and to increase the efficiency of the segmentation phase, either using plain background [3], [4], fixed set of gestures under controlled environments. In this study too we have specified the type of gestures which is on a plain dark background and uniform lighting conditions.
This work compares two algorithms, one which uses a complicated series of filtering, background subtraction and edge detection and the other a very easy and simple preprocessing and segmentation stage. The efficiency of both segmentation techniques is compared by the image feature vectors obtained in the feature extraction stage by the application of the 2D-DWT.

2. Literature review

Gestures are the meaningful expressions of the human body and they are the powerful means of communication. In this section we discuss similar researches conducted in this area. Asanterabi Malima et al [8] states that hand gesture recognition is a challenging problem in research to confront. So they considered a fixed set of manual commands and a reasonably structured environment to develop a simple, effective procedure for gesture recognition. They exemplified the efficiency by segmenting the hand region, locating the fingers, and classifying the gesture. The method is invariant to translation, rotation, and scale of the hand.
Hand gesture pointing location detection was implemented in [5]. Haar classifier is used for feature extraction but due to poor illumination, the skin color identification proved to be difficult. Variation of illumination, rotation, size and position of the gesture images, efficient feature representation and classification are the main challenges in the development of real time gesture recognition system.
Support Vector Machine (SVM) was used to train feature vectors and testing is done with previously learned SVM by comparing it with same gesture recognition at different lightning conditions but again problems arises when there is a complex background [7].
Song.W et al [6] has used background subtraction and frame difference techniques for recognizing hand gesture areas. The moving part is detected by subtracting the current image and eliminating the image background. Then dynamic threshold method is used to detect the moving hand gesture.
Dardas et al., [9] discuss that real time tracking and detecting system using SIFT features and SVM. The hand gestures are captured

in different scale, direction and size and the captured image Eigenvectors are extracted and trained. The test image gestures are captured and the edges are detected and compare with the trained features by using the Euclidean Distance. The minimum distance and minimum weight based features are identified and the hand gestures also recognized. The performance results showed a very high accuracy for the multi class SVM classifier

3. Methodology

3.1. Image collection

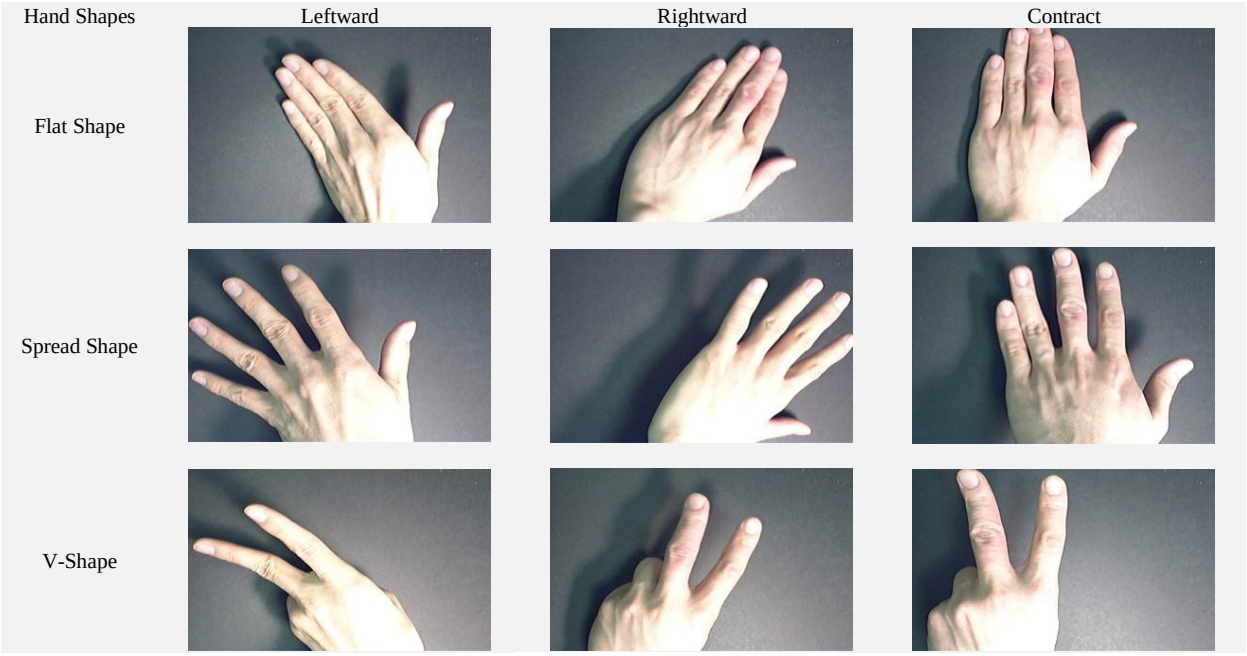

Fig. 1: Nine Types of Hand Gestures.

The set of 100 image sequences were recorded at a 30 fps rate with a resolution of 320 x 240, by using a fixed camera directly over the hand, on a dark background. The Leftward and Rightward class covers a full rotation of the gesture either along the left direction or along the right direction. In addition, this has been taken under 5 illumination conditions. The illumination settings are such that light source is projected from all 4 corners and from above. The Contract class consists of the same gestures with a contracted finger movement, again taken under [5] illumination conditions. Therefore, for each illumination setting we have 20 images that spans different angle of rotation of the image.

3.2. Pre-processing

The quality of an image is dependent on many factors like the equipment used to capture the image, lighting conditions and other environmental aspects. So it is imperative that the image pass through preprocessing step like noise removal, scaling, RGB to gray conversion or a combination of these methods before the image can be segmented. The Cambridge database has RGB images with the resolution of 240 × 320 pixels Fig 1. These images are converted to gray-scale and then resized into 256 × 256 pixels.

3.3. Image segmentation using IGSOCED

Canny edge detection is considered as an optimal algorithm when it comes to edge detection. But it has shown to exhibit poor performance in instances of impulse noise and uniform background. Canny edge detection is implemented in five steps:

1) Image smoothening using Gaussian filter.
2) Determining the gradients, which mark the direction of the edges.
3) Non-maxima suppression for thinning the edge.
4) Double thresholding for determining the strong and weak edges.
5) Hysteresis for tracking the strong and weak edges.

The Cambridge hand gesture database is a widely used dataset and as seen in Figure1 has various gestures on a dark background. The fact that the gestures were taken on a plain background and that each class had around 100 gestures were the main reasons for choosing this database. The dataset consists of 900 images which are classified into 9 different gestures classes. For every class, dataset has 100 image sequences, based on hand rotation and contraction, taken under five different illumination conditions, 10 arbitrary motions and with 2 subjects.

The Canny edge detection algorithm gives good results but shows poor detection in the presence of impulse noise and composite edges, and is prone to false edge detection where there are smooth backgrounds. The main drawbacks of the traditional algorithm (Canny Edge Detector)are:

- Using Gaussian filter to remove noise would result in losing the high frequency edge components.
- The gradient amplitude calculation using the 2 x 2 neighborhood windows makes it sensitive to noise and prone to fake edge detection
- The edges have a tendency to produce multi point responses.

A novel improved canny edge detection was proposed in [11] which uses Non-Local Means filter and Particle Swarm Optimization [14] based edge detection in place of the Sobel operator. The algorithm improves the standard Canny [15] by:

1) Replacing the Gaussian filter with the Non-local Mean filter.
2) Using PSO for gradient detection.

This ensures detection of continuous and smooth edges. Here we have used the Non Local Mean Filter (NLMF) [10],[13] for noise removal, where all the pixels in the image is considered for calculating the mean which is weighted by the similarity to the required pixel. In this filter the weighted average of the local neighborhood is not taken but rather the whole image will be scanned for pixels that are similar and then the average of the pixels around those similar pixels is taken. Every pixel in the image which has the same neighborhood will be replaced with this new calculated average value.

If we consider a pixel p that has similar neighborhood with two other pixels p1 and p2 in the image, then after one scan of the whole image p1 and p2 will be assigned a higher weight $w(p,p1)\ and\ w(p,p2)$ while those pixels whose neighborhood is not similar will be assigned lower weights. So the least similar pixels weight would almost be negligible. After estimating the self-similarity value of pixel, the NLM filter calculates the value of pixel p as an average of all pixels in the image whose neighborhood is similar to p and it is computed as follows,

$$NLM(V)(p) = \sum_{q\varepsilon V} w(p,q)V(q) \quad (1)$$

Where, $(V)(p)$ is the noise free image and weights $w(p,q)$ meet the subsequent conditions $0 \le w(p,q) \le 1$ and $\sum_q w(p,q) = 1$.

It's a highly powerful and efficient filter which preserves the edges to a large extent. The background is subtracted using Otsu Thresholding method because of its simplicity and efficiency in separating images that have bi-modal histograms. The examined pixel intensity value maximizes the inter class variance at the same time minimizes the intra class variance between the pixels. Edges are then detected using the proposed Improved Global Swarm Optimization based Canny Edge detection (IGSOCED) [11]. The stages of the Image Segmentation process is shown in Fig2. Here we have used a combination of two Mathematical models to further improve the location of edges in the image.

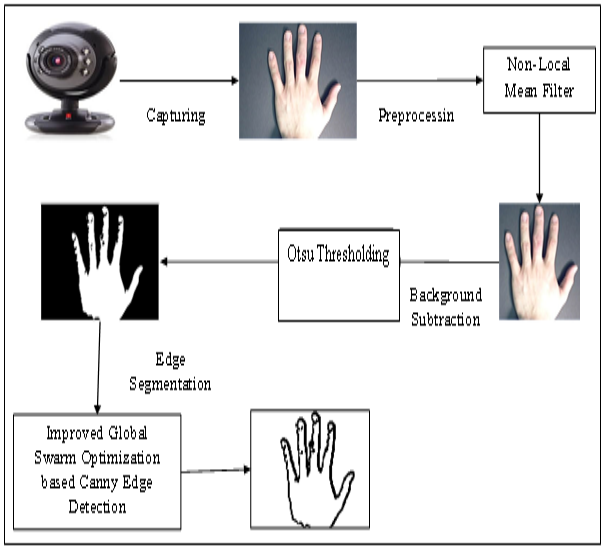

Fig. 2: Image Segmentation Using IGSOCED.

In Particle Swarm Optimization (PSO) process, all particles in an n-dimensional space are randomly placed and initialized with an initial velocity, and the inclusion of the particle into a particular group is determined by its satisfying a fitness criteria. Each particle has a global value G_{best}, which will be the best value of all the particles in the group and a personal best value, E_{best}.

The particle space comprises of the total number of edge points and for every iteration the particle is assigned a value depending on the direction of the edge. The particle velocity is updates as follows:

$$v_i^{k+1} = \omega^k v_i^k + c_1 r_{1i}\left(P_{best-i}^k - x_i^k\right) + c_2 r_{2i}\left(G_{best-i}^k - x_i^k\right) + c_3 r_{3i}\left(E_{best-i}^k - x_i^k\right) \quad (2)$$

Where, $c_3 r_{3i}\left(E_{best-i}^k - x_i^k\right)$ is defined as the velocity Improvement Factor (IF), v_i is the velocity of i[th] particle and $c_1\ and\ c_2$ are learning factors. In the given equation k is defined as the number of iterations and $r_1\ and\ r_2$ are defined as independently distributed random numbers from the range 0 to 1.

3.4. Feature extraction

The feature extraction process was carried out in two steps: firstly the wavelet coefficients were extracted using Multiresolution 2D-DWT and then gray-level co-occurrence matrix is used to extract texture based statistical features. In this section we discuss about the feature extraction stages that were used for evaluating the performance of the system.

3.4.1. Multi-resolution DWT

The wavelet is a powerful mathematical tool for feature extraction, and has been used to extract the wavelet coefficient from images. Wavelets are localized basis functions which are shifted and scaled types of certain fixed mother wavelets. The main benefit of wavelets is that they offer localized frequency information about a function of a signal, which is mainly beneficial for classification. An image will have a combination of varying statistics of high resolution, that correspond to the finer details like the edges, and coarse statistics that will represent the homogenous regions. This property becomes very useful for multi resolution wavelet transforms to be used for feature extraction from images. The multi resolution analysis is implemented using a tree structure of high pass and low pass filters. Initially we input the whole image and we get 4 sub-images as output, which are the diagonal, vertical and horizontal details and the approximation components. And then the approximation component is further subdivided into four sub-images. The number of level of decomposition depends on the type of image and application that we require. In this research Coiflet4 Wavelet has been used as the mother wavelet.

3.4.2 Statistical feature extraction

In first-order statistical texture analysis, information on texture is extracted from the histogram of image intensity. This approach measures the frequency of a particular grey-level at a random image position and does not take into account correlations, or co-occurrences, between pixels. In the analysis of second-order statistical texture, information on texture is based on the probability of exploring a pair of gray-levels at random orientations and distances over a whole image. The statistical features from images are acquired using GLCM. In this technique, Gray level co-occurrence matrix was created and the statistical texture features such as energy, contrast, correlation and homogeneity were found from the LH and HL sub-bands of the three levels of wavelet decomposition. Due to its large dimensionality, GLCM tends to be very sensitive to the size of the samples that are analyzed. There are two ways to work around this problem: the number of gray levels can be reduced or various metrics of the matrix can be derived from the matrix values. Reducing the size often affects the performance and the ranking of the features extracted, so in this thesis GLCM values have been used to compute metrics that give a good representation of the image. The following features are used:

Homogenity - Homogeneity refers to the closeness of distribution of elements and is computed as:

$$\frac{M(p,q)}{1+|p-q|} \quad (3)$$

Contrast - Contrast is the measure of gray level variation and is given by:

$$\sum_{p,q}|p-q|^2 M(p,q) \quad (4)$$

Correlation – Correlation is the measure of linear dependence between the pixels, given by:

$$\frac{(p-\mu p)(q-\mu q)M(p,q)}{\sigma_p \sigma_q} \quad (5)$$

Where μp and μq are the means and σ_p, σ_q are the standard deviations of M_p and M_q.

Energy - Energy is computed as the sum of squared pixel values, computed as:

$$\sum_{p,q} M(p,q^2) \quad 6)$$

Using these metrics we get a 4 x 1 feature vector which will be used as input to the classifier.

3.5. Classification

Having done the image segmentation and feature extraction, the input hand gestures are recognized using Radial Basis Function Neural Network (RBFNN). RBFNN is a classification of the Feed Forward Network, which can be used for approximating functions and recognizing patterns. RBFN typically has three layers: the input layer which will correspond to the dimension of the feature vector, the hidden layer which is the radial basis functions and the output layer which would correspond to the number of classes. The Radial Basis Functions will transform the input space into hyperspheres (which is the number of nodes in hidden layer) by applying a nonlinear transfer function. At the output layer a linear transformation occurs, where hidden layer outputs are linearly combined to generate a probabilistic value at the output layer.
A two-phase learning methodology was used to train the RBFNN: k-means clustering was used to determine the centers and then output weights were calculated using a supervised learning algorithm, gradient descent. The classifier was trained with 450 images from the database (50 images per class) and tested using 100 images, which was chosen randomly from all classes. While training and testing, the input image will be first rescaled and converted to grayscale and then image segmentation is done using NLM filter and IGSOCED. The statistical features of the segmented image are extracted and then given as input to the RBFNN for classification. The RBFNN will have four input nodes which corresponds to the features extracted using the GLCM and 6 hidden nodes. The numbers of output nodes are nine which correspond to the 9 types of gesture classes that have to be classified.

4. Results and discussions

We have compared the performance of the IGSOCED algorithm with that of the traditional Canny edge detection algorithm. We have processed 3 images, shown in Fig 3, using IGSOCED and standard Canny Edge Detection Algorithms and then multi-level 2D-DWT is applied to extract the feature vector, using MATLAB 2012a. Coiflet4 Wavelet has been used as the mother wavelet for the 2D- DWT, after applying 3 level of DWT, the statistical features from the images are obtained using Gray-Level Co-occurrence Matrix (GLCM).

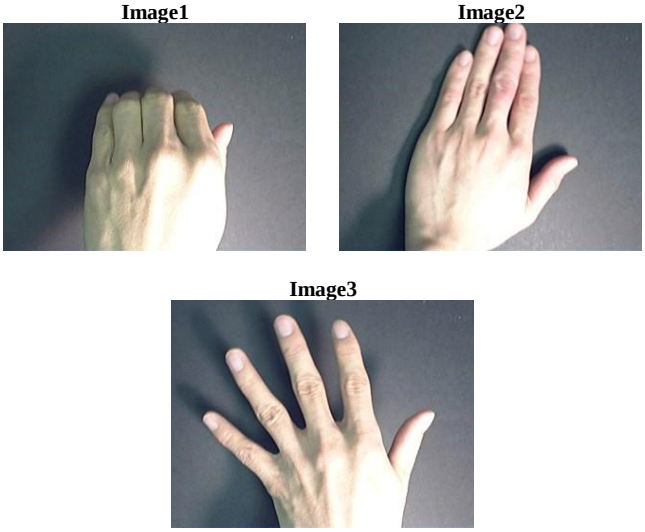

Fig. 3: Hand Gesture Images.

The input to the feature extraction is 256 x 256 edge segmented image. After applying 3 levels of Multi resolution Coiflet Wavelet, the output approximation image will have 32 x 32 pixels. Features are then extracted using GLCM. The GLCM extracts energy, contrast, correlation and homogeneity statistics. These would form the feature vector which could be given as feature vector for gesture classification.
Performance metrics such as Energy, Entropy, Standard deviation and Variance are calculated and analysed for three images using both the proposed Image Segmentation algorithm that uses IGSOCED and Image Segmentation that uses standard Canny, shown in Table1. The plot of the Energy and Entropy is shown in Fig 4

Table1: Metrics Comparison for IGSOCED and Canny

S. No	ENTROPY		ENERGY	
	IGSOCED	Canny	IGSOCED	Canny
Img 1	1.68747	2.63766	0.98124	0.95470
Img 2	1.75221	2.7260	0.97453	0.95514
Img 3	2.41130	2.8593	0.96949	0.95850
S.No	STANDARD DEVIATION		VARIANCE	
	IGSOCED	Canny	IGSOCED	Canny
Img 1	0.043350	0.043352	0.001879	0.00187969
Img 2	0.043350	0.043351	0.001879	0.0018796
Img 3	0.043350	0.43352	0.001879	0.0018796

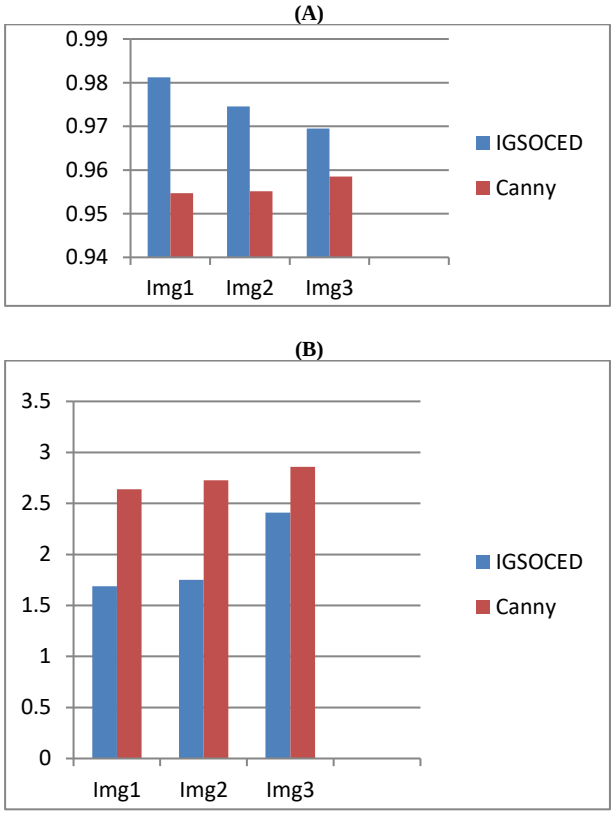

Fig. 4: A) Entropy Plot for Img1, [2] and [3] B) Energy Plot for Img 1, 2 and 3.

Table1 shows the comparison of entropy, variance, standard deviation and energy of the signals after applying Coiflet wavelets The Energy and Entropy plot clearly shows that Segmentation using IGSOCED gives better values.
RBFNN has been used for the purpose of classification and is implemented using MATLAB 2012a. The NN will have 4 input nodes, one hidden layer with 6 nodes and output layer with 9 nodes. 450 images have been used for training the network and 100 images have been used for the purpose of testing. Since we have used the two-phase learning methodology we have three parameters to determine: the centers, c, and the spread of the basis function, σ , and the weights, w, of the output layer. The selection of the centers is done using k-means clustering algorithm. Then

the scaling parameters (width) are computed using p-nearest neighborhood heuristics. The final step is to train the output weights using gradient descent. Once the centers, c_j, and the width, σ, are obtained then the weights of the output layer can be computed. The gradient descent is a supervised learning algorithm that is simple and gives performance better than RBFNN that are trained conventionally. Cross-validation has been used in the training phase to finalize the parameters.

The performance of the classifiers is dependent on many factors like the number of training and testing samples used; the effectiveness of the features extracted and the type of classifier that is chosen. The performance of the classifier is analyzed using the following metrics:

$$Accuracy = \frac{gestures\ rightly\ classified}{total\ number\ of\ gestures} * 100 \quad (7)$$

$$Sensitivity = \frac{TP}{TP+FN} \quad (8)$$

$$Sensitivity = \frac{TN}{TN+FP} \quad (9)$$

Where, TP is True positive, FN is False Negative, TN is True Negative and FP is False Positive. Accuracy gives the probability of a correct classification, Sensitivity gives the percentage of images that have been truly classified into its true class and Specificity is the false positive rate. TP (sensitivity) can then be plotted against FP (1 – specificity) for each threshold used. The resulting graph is called a Receiver Operating Characteristic (ROC) curve (Figure 2).

The following table (Table 2) gives the performance comparison of the classifier when the gestures were segmented using IGSOCED and Canny, and corresponding plot is shown in Figure5.

Table 2: Recognition Ratio of Both Segmentation Stages

Methods	Data set		Recognition Ratio (%)		
	Training	Testing	Training	Validation	Testing
Image Segmentation using IGSOCED	450	100	99.48	98.32	96.26
Image Segmentation using Canny Algorithm	450	100	96.32	95.43	92.7

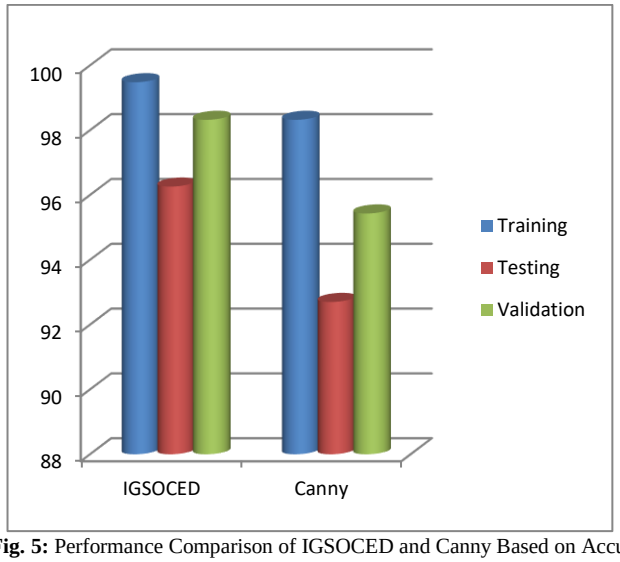

Fig. 5: Performance Comparison of IGSOCED and Canny Based on Accuracy of Classifier.

From the results table, it is clearly seen that the classifier performed much better with 96.26% recognition for IGSOCED as compared to 92.70% for canny algorithm, thus proving the efficiency of the proposed Image segmentation algorithm. The ROC plot for the RBFNN classifier for IGSOCED is shown in Figure 6. Class 1, which corresponds to the V gesture has shown 100% recognition while other classes have shown varied sensitivity and specificity.

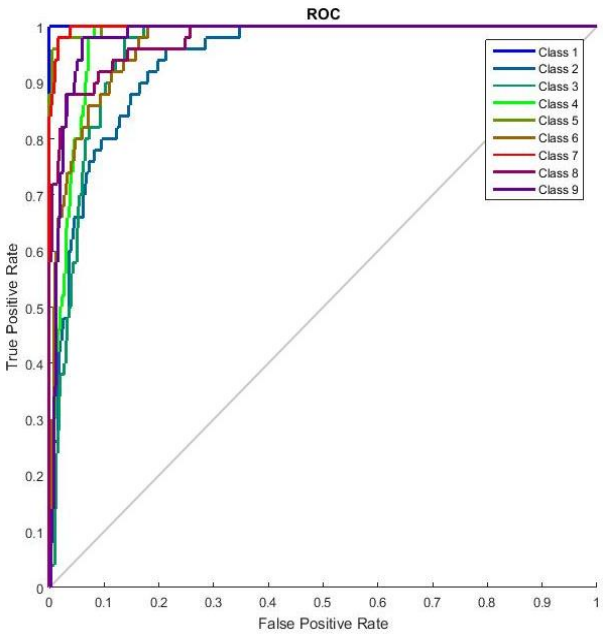

Fig. 6: ROC Curve for RBFNN.

5. Conclusion

In this paper, the efficiency of the proposed Image Segmentation algorithm that uses an improved Canny algorithm has been evaluated with a standard Canny Edge Detection Algorithm. The proposed algorithm uses Particle Swarm Optimization at the gradient detection stage as compared to the Sobel operator that is used in standard Canny detection. The performance is verified using a Radial Basis Function Neural Network for classifying the input hand gesture images. The classifier gives a Recognition ratio of 96.26% when IGSOCED is used for segmenting the images as compared to 92.70 % for standard Canny. The higher classification accuracy, when images are segmented using IGSOCED, clearly shows that the proposed algorithm retains more edge information than the Canny.

Acknowledgement

The author would like to thank her guide, Dr Kamlraj Subramaniam, for guidance and support throughout the research.

References

[1] Zhou, Y., Jiang, G., & Lin, Y. (2016). A novel finger and hand pose estimation technique for real-time hand gesture recognition. *Pattern Recognition*, *49*, 102-114. https://doi.org/10.1016/j.patcog.2015.07.014.

[2] Pisharady, P. K., &Saerbeck, M. (2015). Recent methods and databases in vision-based hand gesture recognition: A review. *Computer Vision and Image Understanding*, *141*, 152-165. https://doi.org/10.1016/j.cviu.2015.08.004.

[3] Ravikiran J, Kavi Mahesh, Suhas Mahishi, Dheeraj R, Sudheender S, and Nitin V Pujari," Finger Detection for Sign Language Recognition", Proceedings of the International MultiConference of Engineers and Computer Scientists (IMECS 2009), vol. I, Hong Kong, March 2009.

[4] J. Mackie, B. McCane, "Finger Detection with Decision Trees", University of Otago, Department of Computer Science.

[5] Raheja, J. L., Chaudhary, A., &Maheshwari, S. (2014). Hand gesture pointing location detection. *Optik-International Journal for Light and Electron Optics*, *125*(3), 993-996. https://doi.org/10.1016/j.ijleo.2013.07.167.

[6] Song, W., Lu, Z., Li, J., Li, J., Liao, J., Cho, K., & Um, K. (2014). Hand Gesture Detection and Tracking Methods Based on Background Subtraction. In *Future Information Technology* (pp. 485-490). Springer Berlin Heidelberg https://doi.org/10.1007/978-3-642-55038-6_76.

[7] Feng, K. P., & Yuan, F. (2013, December). Static hand gesture recognition based on HOG characters and support vector machines. In *Instrumentation and Measurement, Sensor Network and Automation (IMSNA), 2013 2nd International Symposium on* (pp. 936-938). IEEE.

[8] Asanterabi Malima, Erol Ozgur, and Mujdat Çetin, "A_Fast_Algorithm_For_Vision-Based Hand Gesture Recognition for Robot Control", available at., http://people.sabanciuniv.edu/mcetin/publications/malima_SIU06.pdf.

[9] Dardas, Petriu, "Hand gesture detection and recognition using principal component analysis", 2011 IEEE International Conference on Computational Intelligence for Measurement Systems and Applications (CIMSA).

[10] Hongjun Li, Ching Y. Suen, "A novel Non-local means image denoising method based on grey theory", JournalPattern Recognition in ACM, 2015.

[11] Priyanka Parvathy D, Hema C.R, "Hand Gesture Identification Using Preprocessing, Background Subtraction and Segmentation Techniques", 2016 IJAER (pp 3221-3228)

[12] Kamalraj Subramaniam, Renjith V Ravi, "Optimized wavelet filters and modified Huffman Encoding based Compression and Chaotic encryption for Image data" *(ICPR), 2017 IJAER on* (pp. 3961 - 3977). IEEE.

[13] Anoopa JoseChittilappily, Kamalraj Subramaniam, "SVM based defect detection for industrial applications", 2017 IEEE 4th International Conference on Advanced Computing and Communication Systems.

[14] Mahdi Setayesh, M. Z. (2011). Detection of Continuous, Smooth and Thin Edges in Noisy Images Using Constrained Particle Swarm Optimisation. *GECCO'11*. Dublin: ACM.

[15] Canny, J. (1986). A computational approach to edge detection. *Pattern Analysis and Machine Intelligence*, 679–698 https://doi.org/10.1109/TPAMI.1986.4767851.