Proceedings of the 7th International Conference on Computing Methodologies and Communication (ICCMC-2023)
IEEE Xplore Part Number: CFP23K25-ART; ISBN: 978-1-6654-6408-6

Hand Detection and Gesture Recognition in Complex Backgrounds

S.Gnanapriya[#1], K.Rahimunnisa[*2], M.Sowmiya[#3], P.Deepika[#4], S.Praveena Rachel Kamala[#5]

[#]*Department of Information Technology,*

Easwari Engineering College, Anna University

India

[1]gnanapriya.s@eec.srmrmp.edu.in

[3]sowmiya.m@eec.srmrmp.edu.in

[4]p.deepika@eec.srmrmp.edu.in

[5]praveena.s@eec.srmrmp.edu.in

[*]*Department of Electronics and Communication Engineering*

Easwari Engineering College, Anna University

India

[2]rahimunnisa.k@eec.srmrmp.edu.in

2023 7th International Conference on Computing Methodologies and Communication (ICCMC) | 978-1-6654-6408-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICCMC56507.2023.10084181

***Abstract*— In this paper, a Convolutional Neural Networks (CNN) based hand detection model that, on a major note, focuses and segments only the hands from any complex background using the Open-CV libraries for real-time computer vision, is proposed. Based on the features extracted from the region of interest, the VGG16 CNN Architecture classifies and predicts the gestures, based on the trained data. The system is trained by using binary images, so that the background is eliminated and classification is done only on the edges. This approach increases the performance of the system with respect to time. The major step involved in the proposed system is Background Elimination, which is carried out using a series of Open-CV methods and functions. Hand Detection Systems find applications in various domains ranging from Sign-Language Detection to Human-Computer Interaction.**

***Keywords*— Convolutional Neural Networks, Hand detection, Segmentation, Mixture of Gaussian Model 2, real-time, Background Elimination, Feature Extraction, Gesture recognition**

I. INTRODUCTION

The modern computer and mobile devices have evolved through various interfaces, over the years, from keyboards and mouse - to touch screens-and now to touch less interfaces, being the most worked upon technology in recent times. Touch less interfaces enable "Contactless Interactions" which makes interactions more hygienic, fun and also improves User Experience. Touchless Interfaces depend mainly on two technologies- i)Gesture Recognition and ii)Voice Recognition.

This paper revolves around the development of a Gesture Recognition System to provide an effective and efficient way of conversation between the user and computer. The Hand Gesture Recognition System proposed, involves the calculation and detection of the hand's configuration, orientation, movement and location to recognize the particular gestures, based on the training dataset provided to the Convolutional Neural Networks (CNN) model, and the background cancellation techniques, provided using Open-CV libraries. The existing model produces results only when the gestures are captured in a white background, (ie..) without any noise, distortion or multiple, complex objects in the background. This limitation has been overcome in the proposed system, and an effective way of hand detection is proposed. The paper contributes by providing a F1 score of 0.90, which is basically the test accuracy.

II. EXISTING SYSTEM

Vision based hand gesture recognition system is a great boon to the hearing impaired community and it can be used in real time. They can express their thoughts using sign language. Hand gesture recognition is applied in vast varied areas like ambient assisted living for elderly applications; the background needs to be eliminated in real time. This is achieved in two steps. Step one detects the hand and step two recognizes the gesture. Hand regions need to be segmented from the background region. The probabilistic model MOG2 mixture of Gaussian is used [11] to remove background or extract the foreground object from the image using skin color as threshold. [1] Used Fuzzy C Means Clustering to remove background objects from the input image and [4] used fusion of Gaussian Mixture Model of skin color and Ada boost Classifier based on HAAR is used for background removal. An improved Gaussian Mixture Model based on adaptive recursive algorithm [13, 12] handles change in illumination. [5] Used depth based approach where separate skin threshold for hand and face is defined as depth of hand from camera. To recognize the gesture, the use of SIFT and SURF extracts features and SVM for classification is proposed by [8]. The system performance of Hand Gesture Recognition Database is compared with and without augmentation and the performance of the system is found to be greater[7] where classification is carried out with CNN, and the system performance is measured using precision, recall, F-measure and accuracy. Deep Neural Network classifies the gesture without the need for a hand segmentation stage and gives high performance on objects with simple background compared to complex background [9]. Lightweight CNN is used for gesture recognition after removing background [2] and evaluating the performance using F1 score. CNN architecture VGG16 to recognize and classify 11 hand shapes of type both static and dynamic(left click, right click, zoom, double click, pointer, cursor etc..) using Transfer learning and also compared the performance with AlexNet

architecture[10]. Paper [3] used fusion of multi modal features to recognize gestures, which improved the performance of static gesture recognition. Static gestures are classified using trained classifiers and achieve better accuracy, whereas dynamic considers both hand shape and motion of hand for classification. Compared to traditional algorithms, machine learning based approaches and Deep learning based approaches, deep learning approaches perform better[14].

The existing system explores hand gesture recognition without considering cluttered backgrounds. In reality avoiding a background environment in real time is not possible, and it has been addressed using Deep Learning Technique VGG16, a deeper and accurate network.

III. PROPOSED SYSTEM

This paper mainly contributes by developing a system that is trained by using binary images, so that the background is eliminated and classification is done only on the edges. The performance of the system is improved with respect to time, as it eliminates the complexity of skin color. Most of the previous works provide a way of classifying the gestures from an input in real-time by capturing a sequence of frames only in a simple white background. This makes it difficult for the system to detect the hands when there are other objects in the background, and hence becomes unable to recognize the gestures. The proposed system provides a simple, yet efficient neural network based classifier for detection of the structure of our hands and Gesture Recognition, even in complex backgrounds by deploying the VGG16 CNN architecture.

The Proposed system architecture shown in Fig.1 eliminates the complex background by masking the background and classifies the shown gesture by providing the annotation. The major stages involved in the proposed system are Pre-processing, background Elimination, Feature Extraction and Classification or Gesture Recognition.

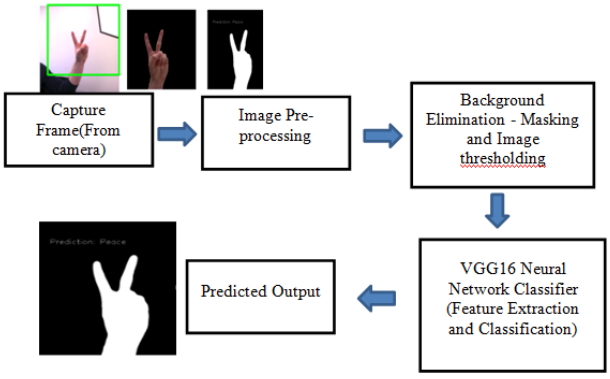


Fig.1 Main Structure of the interactive system.

The system provides a step-by-step method to recognize and classify the gestures in real-time. The model is trained by using images that undergo background-masking and binary thresholding ie., with only the edges of the hand, thereby eliminating background. The trained VGG model then performs feature extraction and classification on the transformed real-time images into binary images.

A. Image Preprocessing

The image captured from the screen is resized and transformed into a NumPy array for it to be understood by the trained VGG model. The system convolves the input image using a bilateral filter - a non-linear, smoothing filter used to preserve the edges of an image, to enhance its quality by reducing noise. Convolutional Neural Networks do not require much of preprocessing compared to the other primitive models, as the model can self-learn characteristics with the convolutional filters.

B. Removal of Background

This paper proposes the concept of masking to eliminate the background, to make hand detection simpler. If an image is captured before placing the hands in the scene, a mask can be created for the background, thus making background elimination very simple.

A series of OpenCV libraries and methods have been used to perform masking and binary thresholding of the images captured, in order to make it similar to the images used for training. The Python Imaging Library (PIL)- Pillow is applied to the input image to convert the RGB images(3-channels) into Grayscale images(1-channel).

1) Background Masking:

As any image can still contain noise, a Gaussian Blur method is used to smoothen the input image. Then the contours are identified to detect the edges of the main object. This is done by using the Chain Approximation function in OpenCV to minimize or eliminate the misidentified contours. The MOG2 Background Subtraction method is used to generate a mask for the background and is combined with the image using Bitwise AND operator to generate the masked images as shown in Fig.2.

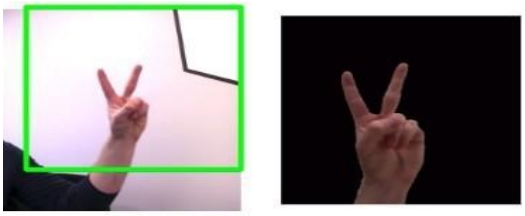


Fig.2 Background Masked Image

2) Binary Thresholding

Each pixel of the masked image is compared to a particular threshold value. If the pixel value is less than the threshold, it is set to 0, else to the maximum value. The system uses Otsu's thresholding function for this purpose, which provides a great advantage of automatically calculating the threshold value by itself, in contradiction to the other threshold functions like Basic or Adaptive thresholds, where the threshold value needs to be provided manually. The Otsu's algorithm identifies the threshold value by minimizing the weighted within-class variance and provides images with better quality. Thus, after thresholding, the areas with maximum value are considered to be the region of interest for further processing by the trained VGG model.

The binary thresholding method has been implemented so as to generate a clear and absolute outline of the hand and makes it efficient to generalize across different skin tones.

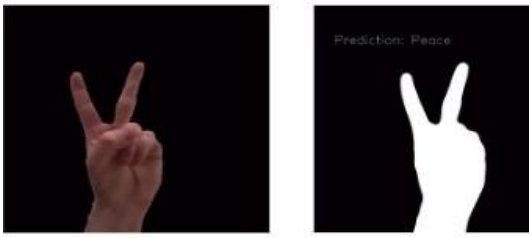


Fig.3 Experimental results after converting masked images into binary images.

Fig .4 Conversion of RGB images to binary images

Fig.3 and Fig.4 shows the experimental results produced after performing Background Masking and applying Binary Image thresholding.

C. Feature Extraction and Classification

Convolutional Neural Networks (ConvNet/CNN) are a part of Deep Learning algorithms that are deployed to take in an input,analyse each aspect of the input, as desired and trained, to extract the features and provide outputs with maximum accuracy by learning on its own.

CNN uses a series of filters on an image to extract its high-level features and then performs prediction or classification. As shown in Fig.5 CNN carries out these functions in 3 major steps:

I. i. Convolutional Layers- to mainly to extract the dominant features

II. ii. Pooling Layers-to reduce the spatial size of the convolved image

III. iii. Fully connected Layers-to perform classification by learning a non-linear function.

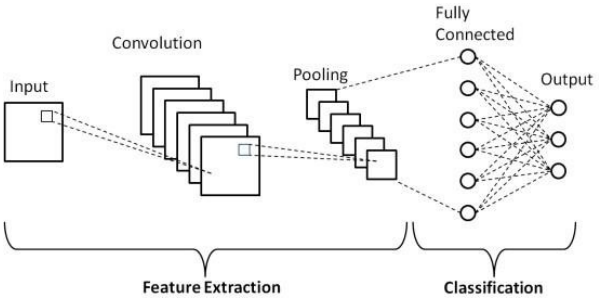


Fig 5. Layers in Convolutional Neural Networks [15]

In this paper, a Convolutional Neural Network has been built using Keras and Tensorflow. An excellent pre-trained VGG16 model has been implemented. Four dense (fully connected) layers have been built on the top, along with a dropout layer, as per the requirements. With the added layers, the CNN model can extract more dominant and high-level features with improved accuracy.

1) Conv2D — is a 2D convolution that inputs an image with a matrix size (300,300) and uses a kernel with input size (3,3) to create 32 feature maps.
2) Batch Normalization — standardizes each batch of data by the mean and variance methods in reference to each of the mini batches of data to train the model faster and converge much more quickly.
3) MaxPooling2D — down samples the 32 feature maps that pass-through from the Conv2D output into the MaxPooling of (2,2) size, whose output is then passed through Conv2D with 128 feature maps and then MaxPooling with (2,2) size.Another feature learning process takes place with Conv2D 32 feature mapping and (2,2) max pooling.
4) Flatten:- This layer converts the 3 dimensions of the feature learning output into a 1D column vector which is connected to a fully connected neural network part.
5) Dropout layer — 25% (0.25) of the neurons are randomly excluded during each of the update cycles.
6) Dense layers —The fully connected layers connected deeply to the preceding layers should have the number of neurons equal to the number of classes. In this case, we have 5 classes and hence 5 neurons.
7) Image Input Layer — Size of feature map- (224,224,3)

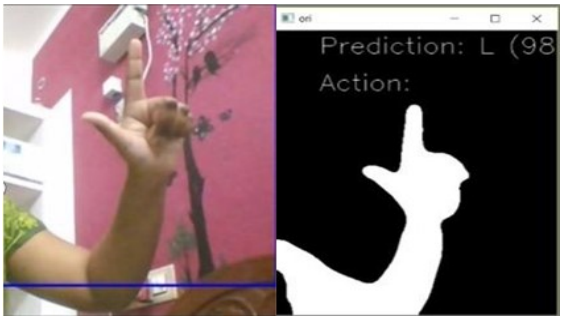


Fig. 6 Experimental results after Feature Extraction and Classification

Table.1 VGG16 Model Configuration details

Layer Type	Feature maps	Size of Filter matrix
1stConvolutional 2D Layer ReLU Layer MaxPooling Layer	32 1	(5,5) (2,2)
2ndConvolutional 2D Layer ReLU Layer MaxPooling Layer	64 1	(3,3) (2,2)

3rd Convolutional 2D Layer ReLU Layer MaxPooling Layer	64 1	(3,3) (2,2)
1st Dense Layer ReLU Layer	128	-
2nd Dense Layer ReLU Layer	128	-
3rd Dense Layer ReLU Layer	128	-
Dropout Layer	25%neurons excluded	-
4th Dense Layer Softmax Layer	5	-

The model has been distinguished into a test-train split model, considering 80% of the images for training and the rest for testing.

Any neural network model requires an activation unit majorly for interaction i.e. to convert any given input into the desired output and for providing a non-linear effectiveness. In the developed model, ReLU (Rectified Linear Activation Unit) is used, as it performs with better accuracy often, and also makes it easier to train the model.
Optimizers are algorithms used in the neural networks model to resolve the optimization problems, reducing the losses in the model by changing the learning rates, weights so that the functions used are minimized. We have used the Adaptive Moment Estimation Optimizer (Adam) in the model as it provides faster adaptive learning of the parameters. It converges faster by using the Momentum and Adaptive Learning rates and provides the best on average, compared to other optimizers like SGD,
Momentum, etc. The Model parameters of VGG16 are provided in Table.1

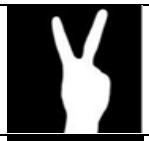
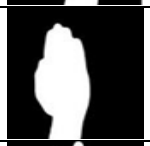
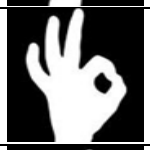
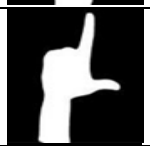
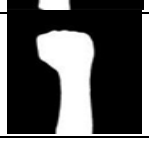
Action	Gesture
Peace	
Palm	
Okay	
L	
Fist	

Fig.7 Images of the Gestures trained

As shown in Fig.6, the system captures the real-time color images and predicts the gesture, based on the classification, as a text message with the accuracy, along with the plotted binary image.

D. Dataset

The VGG model needs to be trained to recognize and classify the gestures in real-time. The model is trained by using the binary images as shown in Fig.7 with only the ground truth, thereby eliminating background.

Fig.8 Training dataset for the gesture 'L'

To train the model 120 training image samples out of 150, of size 640x480 pixels under each class of gesture are used.

The database to test the hand gesture recognition system was created using125 static images with different backgrounds. The static images provided are of size 640x480 pixels collected from a computer-vision camera. The gestures were collected from 10 different people. The system is tested with approximately around 40 to 70 samples under each category with different orientations, size, flips, angle and positions of the palm. Fig 8 provides the dataset used for the recognition of the gesture 'L'.

IV PERFORMANCE EVALUATION

To describe the performance of the classifier, we used the Confusion Matrix function and Classification Report method for the quality of predictions from the SkLearn library of python, applied over the testing datasets.

```
In [86]: print(classification_report(y_pred_classes, y_true))
```

	precision	recall	f1-score	support
0	0.91	0.89	0.90	44
1	0.86	0.95	0.90	39
2	0.90	0.81	0.85	53
3	0.85	0.91	0.88	44
4	0.95	0.93	0.94	76
avg / total	0.90	0.90	0.90	256

```
In [87]: print(confusion_matrix(y_pred_classes, y_true))
[[39  2  0  2  1]
 [ 0 37  1  1  0]
 [ 3  3 43  3  1]
 [ 1  0  1 40  2]
 [ 0  1  3  1 71]]
```

Fig.9 Performance Validation results

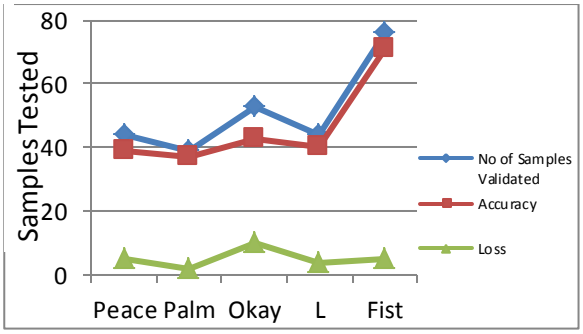


Fig.10 Validation Accuracy and Loss Graph

In statistical analysis of binary classification, the F1_score is a measure of a test's accuracy. It is calculated from the precision and recall of the test. As shown in Fig.9 the system was able to provide an average of 90% precision or accuracy with respect to each of the gestures it has been tested for. The validation accuracy and loss is shown in Fig.10.

V CONCLUSION

The paper proposes a system that uses the Convolutional Neural Networks for Gesture Recognition. Background Cancellation techniques are made efficient using the Python 3.7 OpenCV Libraries and Feature Extraction and Classification is performed using the VGG16 CNN architecture. Although Deep Learning based semantic techniques are available for background elimination, the proposed image processing based approach combined with Machine Learning techniques eliminates the need for sophisticated devices and the large number of background images required for learning. Unlike that used a kinect sensor to collect depth input for real time analysis, the proposed method used a single camera to collect input.

REFERENCES

[1] A Pugazhenthi, G.Sreenivasulu, A Indhirani Background Removal by Modified Fuzzy C-Means Clustering Algorithm",IEEE International Conference on Engineering and Technology(ICETECH) 2015.

[2] Adam Ahmed Qaid MOHAMMED , Jiancheng Lvand MD. Sajjatul Islam, " A Deep Learning-Based End-to-End Composite System for Hand Detection and Gesture Recognition ", Sensors 2019.

[3] JIALI DUAN and JUN WAN,SHUAI IAOYUAN GUO, STAN Z. LI, " A Unified Framework for Multi-Modal Isolated Gesture Recognition"ACM Trans. Multimedia Comput. Commun. Appl. 14,1s, Article 21 February2018.

[4] Jing-Hao Sun, Ting-Ting Ji, Shu-Bin Zhang, Jia-Kui Yang and Guang-Rong Ji, " Research on the HandGesture Recognition Based on Deep Learning",2018, 12th IEEE International Symposium on Antennas, Propagation and EM Theory (ISAPE)

[5] Keshav Sinha et al, "A Computer Vision-Based Gesture Recognition Using Hidden Markov Model"Springer, Innovations in Soft Computing and Information technology,2019

[6] Marouane Benmoussa, Abdelhak Mahmoudi," Machine Learning for Hand Gesture Recognition Using Bag-of-words " international conference on Intelligent Systems and Computer Vision (ISCV),2018.

[7] Md. Zahirul Islam, Mohammad Shahadat Hossainy, Raihan Ul Islamz and Karl Andersson Static Hand Gesture Recognition using Convolutional Neural Network with Data Augmentation, Joint IEEE 8th International Conference on Informatics, Electronics & Vision (ICIEV) & the 3rd International Conference on Imaging, Vision & Pattern Recognition (IVPR), 2019.

[8] Ming Jin Cheok1 · Zaid Omar1 · Mohamed Hisham Jaward2,"A review of hand gesture and sign language recognition techniques", Int. J. Mach. Learn. & Cyber. © Springer-Verlag GmbH Germany 2017.

[9] Peijun Bao, Ana I. Maqueda, Carlos R. del- Blanco, and Narciso García, " Tiny Hand Gesture Recognition without Localization via a Deep Convolutional Network " IEEETransactions on Consumer Electronics, Vol. 63, No. 3, pp.251-257,2017.

[10] Soeb Hussain and Rupal Saxena, Xie Han, Jameel Ahmed Khan, Hyunchul Shin, "Hand Gesture Recognition Using Deep Learning" , International SoC Design Conference (ISOCC), 2017.

[11] Soukaina Chraa Mesbahi, Mohamed Adnane Mahraz, Jamal Riffi, Hamid Tairi,Hand gesture recognition based on convexity approach and background subtraction, 2018, IEEE International Conference on Intelligent Systems and Computer Vision (ISCV).

[12] Zoran Zivkovic, Ferdinand van der Heijden, " Efficient adaptive density estimation per image pixel for the task of background subtraction ", Pattern Recognition Letters Elsevier, 2005.

[13] Zoran Zivkovic,"Improved Adaptive Gaussian Mixture Model for Background Subtraction" , Proceedings of the 17th International Conference on Pattern Recognition,(ICPR)2004.

[14] Indian Sign Language Gesture Recognition using Image Processing and Deep Learning, Neel Kamal Bhagat, Vishnusai Y, Rathna G N, Digital Image Computing: Techniques and Applications (DICTA)IEEE, 2019

[15] https://medium.com/techiepedia/binary-image-classifier-cnn-using-tensorflow-a3f5d6746697