International Journal of Computer Information Systems and Industrial Management Applications.
ISSN 2150-7988 Volume 15(2023) pp. 341-350
© MIR Labs, www.mirlabs.net/ijcisim/index.html

Submitted: 1 November, 2022; Accepted: 11 May, 2023; Publish: 23 June, 2023

Ensemble Learning for Static Hand Gesture Recognition using HOG and LBP Features on RGB-D Data

Dayananda Kumar N.C [1], K.V Suresh [2] and Dinesh R [3]

[1]Dept. of Electronics and Communication Engineering, Siddaganga Institute of Technology, Tumkur, India
dayanandkumar.nc@gmail.com

[2]Dept. of Electronics and Communication Engineering, Siddaganga Institute of Technology, Tumkur, India
sureshkvsit@sit.ac.in

[3]Dept. of Information Science and Engineering, Jain University, Bangalore, India
dr.dineshr@gmail.com

Abstract: **Hand Gesture Recognition (HGR) systems has gained a lot of interest in research community due to its application in Human Computer Interaction (HCI), Advanced Driver Assistance Systems (ADAS) and Sign Language Recognition (SLR) for non verbal communication using various hand postures. Multi modal HGR systems with combination of RGB, depth and sensor data etc., have proved to be more efficient as compared to uni-modal systems. Also the classification decision based on the voting of different classifiers can be more accurate than single classifier. In this paper, we propose ensemble classifier of support vector machine, random forest and multi-layer perceptron classifiers for classification of hand gestures. Ensemble classifier is evaluated on HOG, LBP features with principal component analysis (PCA) and the pre-trained VGG16 model based deep features on both RGB and Depth data. Experiments are conducted on two different RGB-D dataset NTU and OUHANDS to evaluate the proposed method. Average classification accuracy of 97.50% is achieved on NTU dataset using the proposed method.**

Keywords: Hand Gesture Recognition, Depth, HOG, LBP, PCA, VGG, Ensemble learning

I. Introduction

Hand Gesture Recognition (HGR) is an important area of research in the field of machine vision and video recognition. HGR aimed at providing information on human physical activity and facilitating the realization of gesture user interfaces [1]. Due to lack of motions and their corresponding spatio-temporal dependency information, the static HGR task is perhaps more complicated. In addition to this, the varying lighting conditions and background, diversity in how people perform the gestures further increase the complexity of the recognition task [2].

Initially, wearable sensors attached to the hand were used to recognize gestures. Electronic signals can be converted from hand movements or finger flexions by these wearable sensors. Currently, vision-based hand gesture recognition systems are capable of capturing a wide range of meaningful inputs without any sensors attached [3]. In the last few decades, research related to hand gesture recognition has received a significant amount of attention [4]. In real-time applications, there are two types of hand gestures: static and dynamic. In case of dynamic hand gesture, hand posture changes over time and is captured in a video. Whereas, posture remains same in case of static hand gestures captured in a image frame. The existing approaches for static HGR are mainly device based or vision based [5]. Device-based HGR requires the subject to wear a wearable sensor device, which limits the ease with which it can be used in real time. As opposed to this, vision-based HGR approaches eliminate the need for wearable devices. As part of this approach, the camera is used in conjunction with a robust algorithm that identifies the hand gesture. It is, however, challenging to design an algorithm that is robust and accurate. Particularly when a single camera is used instead of multiple cameras [6]. Also it is challenging to develop a system that is capable of recognizing hand gestures based on vision. Hand gesture recognition systems that rely on vision-based systems typically require knowledge of machine learning, algorithm design and application-specific programming.

II. Literature Survey

This section presents a brief literature review of latest techniques for recognizing hand gestures using vision. In the study, RGB cameras and depth sensors are used to recognize hand gestures based on machine learning and deep learning algorithms.

A. *Hand Gesture Recognition Using RGB Sensor Input*

The static hand gesture recognition is a process of recognizing hand position and pose from an image. The Histogram of Oriented Gradient (HOG) features represents the distribution of intensity gradients or edge directions, which captures the contour and the edge information [7]. Local Binary Patterns (LBP), can effectively describe the texture characteristics [8] i.e. the correlation among pixels within a local area (e.g., 2 × 2 area), which mainly characterises the local information. LBP features achieves rotation and scale invariance, hence, researchers combined HOG and LBP to increase the recognition precision in various applications like gesture recognition [9], brain tumor detection [10], human activity recognition [11], etc. Using the NUS Hand Posture dataset Houssem et al. [9] marked-out human gestures based on LBP and HOG features. For the recognition of sign language, Deep features of AlexNet and VGG-16 are extracted and SVM is used as a classifier to extract Deep features [12]. Using a standard dataset, the classification accuracy found to be 70% using the leave-one-subject-out cross-validation (LOO CV) test. Richa et al. [13] used Faster R-CNN for hand detection, SIFT (scale-invariant feature transform) for hand tracking and modified back-propagation to recognize the hand gestures in the colored videos captured in a real time.

B. *Hand Gesture Recognition Using RGB-D Sensor Input*

RGB-D sensors can overcome the limitations in gesture accuracy recognition using RGB data. Using RGB-D sensors, hand gestures have been recognized in the following literature. A deep attention network was proposed by Yuan et al. [14] for joint recognition and localization of hand gestures using static RGB-D images. In the CNN framework, deep attention is achieved by using a soft attention mechanism to automatically locate hands and classify gestures using a single network.
Bin et al.[15] presented a novel technique for extracting bag of contour fragments (BCF) features from depth projection maps (DPMs). Three projection views are used to obtain the shape feature: the front view, the side view, and the top view. A final shape feature can be obtained by concatenating the above three views. According to the authors, a major contribution to hand gesture recognition is provided by the front view projection. To recognize hand gestures, Prachi et al [16] merged local binary sub-pattern distance and geometric features and number of fingers from the segmented hand. SVM classifiers were used to classify two datasets using the fusion of different features.
Inspired by the performance of CNN based frameworks, Jaya Prakash et al [17] proposed an end-to-end fine-tuning method for a pre-trained CNN model and applied score-level fusion technique for hand gesture recognition. The usage of pre-trained model results in efficient classification of hand gesture images even with minimal samples.
From the aforementioned literature review, we draw the conclusion that high inter-class similarity, human noise, and variety in backgrounds are the primary factors hindering the improvement of recognition accuracy for RGB input images. Therefore, one of the key steps is computing robust and distinguishing feature representation of the various gesture classes. As a result, in this work, we propose an HGR framework in which conventional features, such as HOG and LBP, as well as deep representation features from the VGG16 framework, are computed for both on RGB and Depth image, and evaluated using an ensemble of different classifiers, such as SVM, ANN, and Random forest. In the subsequent sections that follow, we provide a brief discussion of the proposed framework.

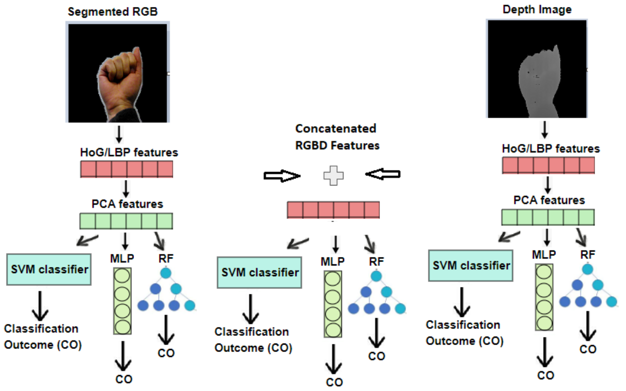

Figure. 1: The proposed static HGR framework using classifier ensembling.

III. Proposed Method

In this paper, we analyze the conventional features like HOG and LBP along with VGG16 deep cnn features, where these features are computed both on RGB and Depth image and evaluated using ensemble of various classifiers like SVM, ANN and Random forest. Here, the objective is to study the performance of RGB and depth features independently and also as a combined RGB-D feature using suitable fusion technique. After analysing the accuracy of these features with ensemble of different classifiers, we can identify the suitable combination of features and classifier suitable for static hand gesture recognition.
The main contributions of this paper are:

1. Analysis of conventional and deep features on RGB data, depth data and combined RGB-D data.
2. Identifying the suitable PCA component after verifying the accuracy on multiple values.
3. Proposed the ensemble approach for static HGR classification on RGB-D data.

A. *Depth Segmentation*

RGB and its corresponding Depth image is captured using Kinect device which forms RGB-D pair. The raw depth image d_r is a matrix of float values where each pixel represents the distance from the sensor to the object in millimeter. The nearest object to the camera sensor will have less depth value as compared to the farthest object. Hence with the assumption of hand region being closer object as compared to the background scene, we can discard the background and segment only the foreground hand region using suitable depth threshold.

Segmented depth map d_s is obtained by subtracting background from depth map d_r using Eq.1,

$$d_s = \begin{cases} d_r, & \text{if } d_r \leq T_d \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where T_d is the depth threshold computed using Eq. 2.

$$T_d = (max(d_r) - min(d_r)) * D_c + min(d_r) \tag{2}$$

D_c is the depth range constant varying between 0 to 1 which depends on the distance between object and the depth sensor. Additionally, the depth data after threshold segmentation is normalised using the min-max normalisation procedure, and converted to a gray scale depth image using the DepthNorm equation Eq.3

$$d_{norm\_s} = 255.0 * \frac{d_s - min(d_s)}{max(d_s) - min(d_s)} \tag{3}$$

where d_{norm_s} is the depth segmented image normalized to pixel range of $[0, 255]$, d_s represents background segmented depth image, $max(d_s)$ & $min(d_s)$ represents maximum and minimum depth values respectively.

Noisy depth outliers are removed using morphological open operation where dilation followed by erosion operation is performed on depth filtered image as a post processing step. Binary depth segmentation mask corresponding to filtered depth image is generated by threshold operation as in Eq. 4. To eliminate the background and create the segmented RGB image, this binary mask is logically merged with the input RGB image using the bit-wise AND operation.

$$d_{bin\_mask} = \begin{cases} 255, & \text{if } d_{norm\_seg} \geq T_b \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

where T_b is the threshold value computed as nonzero minimum of d_{norm_s}, $min(|d_{norm\_s}|) \neq 0$

B. *Feature Extraction*

Feature descriptors represents useful information of an local image patch or the global attributes of entire image. There exists a number of feature extraction methods that derives meaning full information from the given image which discriminates it from other image samples. Here the goal is to have high inter class variance which makes the given data separable from the data of different class and have less intra-class variance or higher degree of correlation between the samples from same class.

In this paper, we analyze the effectiveness of Histogram of Oriented Gradients (HOG) and Local Binary Patterns (LBP) features that are widely used in computer vision applications. These descriptors are computed individually on both RGB and depth image, since these features are of higher dimension it is transformed to eigen space using Principal Component Analysis (PCA) to obtain feature vector with less attributes but with higher variance. Further these RGB and depth features are effectively combined using suitable feature fusion method and analyzed by its classification accuracy.

For further comparative analysis, deep features are extracted using pre-trained VGG16 model on RGB and depth image which are fused to form RGB-D descriptor. These conventional and deep features are ensembled using Support Vector Machines (SVM), Neural Network (NN) and Random forest classifiers to identify the suitable feature extraction and classification methods for static hand gesture recognition using multi-modal RGB-D data.

1) Histogram of oriented gradients feature

HOG feature descriptor [18] effectively captures the shape and appearance locally within an patch of image region by computing the distribution of gradient magnitude and direction. For a given image, exhaustive scanning is performed using sliding window approach and HOG descriptor is extracted locally for all the image patches, these descriptors are finally aggregated by concatenation to form a feature vector.

Hog descriptor computation is briefly explained as following. Gradient magnitude and direction of image patch I is computed using the point derivatives G_x and G_y in x and y direction respectively as in Eq. 5. This can be implemented using convolution operation of I with gradient mask $[-101]^T$ and $[-101]$ respectively.

$$\begin{aligned} G_x(x, y) &= I(x, y + 1) - I(x, y - 1) \\ G_y(x, y) &= I(x - 1, y) - I(x + 1, y) \end{aligned} \tag{5}$$

Using G_x and G_y, gradient magnitude and angle of each pixel is calculated as in Eq. 6

$$\begin{aligned} magnitude(\mu) &= \sqrt{G_x^2 + G_y^2} \\ angle(\theta) &= |\tan^{-1}(\frac{G_y}{G_x})| \end{aligned} \tag{6}$$

After computing the gradient magnitude and angle of all the pixels in 64 x 128 detection window image patch, these pixels are grouped into a overlapping grid of pixels known as block. The amount of overlapping of these blocks B are specified by the stride value S which will be usually set to 8. These blocks are further dividend into non-overlapping cells.
For each cell C, the histogram of length equal to the number of orientation bins $b = 9$ is obtained by placing each pixel of the cell into the histogram bin based on their gradient magnitude and angle. Evenly spaced 9 orientation bins are considered over $0°$ to $180°$ range hence each bin is of $20°$ as in Eq. 7.

$$\begin{aligned} Number\_of\_bins, \quad b &= 9 \\ stepsize(\Delta\theta) &= 180°/b = 20° \end{aligned} \tag{7}$$

Histogram based feature of all the cells within the block are concatenated to obtain block feature vector BF which is normalized using L2 Norm. Block feature length

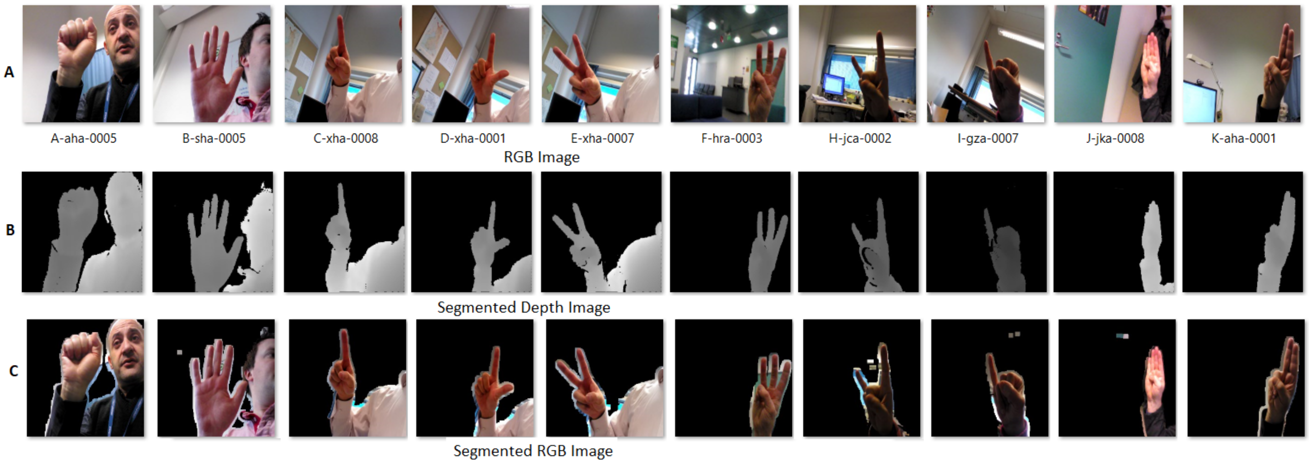

Figure. 2: A-RGB Image, B-Segmented Depth Map, C- Segmented RGB on OUHANDS dataset

L is equal to number of bins b multiplied by number of cells C within the block, $BF \in R^L$. If $C = 4$, then $L = b{*}C = 9{*}4 = 36$ which is the dimension block feature.

Let block feature $BF_i = \{bf_1, bf_2, bf_3....bf_L\}$ where $i \in (1, K)$, K is the total number of overlapping blocks within the image patch detection window. Block features are normalized using Eq. 8.

$$\begin{aligned} BF_{i\_norm} &= \frac{BF_i}{\sqrt{||BF_i^2||}} \\ v &= \sqrt{bf_1^2 + bf_2^2 + bf_3^2....bf_L^2} \\ BF_{i\_norm} &= [(\frac{bf_1}{v}), (\frac{bf_2}{v}), (\frac{bf_3}{v})....(\frac{bf_L}{v})] \end{aligned} \tag{8}$$

The normalized block feature vectors are aggregated over the image patch F to obtain its corresponding feature descriptor. $F_n \in R^{(K*L)}$ where $n \in (1, N)$, N is the total number of image patch sliding window of entire image, hence for a given image HOG descriptor $F_{hog} \in R^{(N*K*L)}$ is obtained.

2) *Local binary pattern feature*

LBP encodes the details of texture variation in the local region by comparing the pixel intensity difference between the center pixel and the surrounding neighborhood pixels within a local window and generates the binary code as shown in Figure. 3. These local features can be aggregated by concatenating all the features from a sliding window and scan the entire image to obtain pattern representation of entire image. It is represented as binary descriptor where the feature vector contains only '0' and '1', this makes the descriptor computationally efficient when compared to other floating point features. It is widely used in addressing the computer vision problems like object detection and classification due to its capability to strongly capture the texture variation and computational efficiency.

Consider a local window of 3 x 3 region with $N = 8$ neighbourhood pixels, center pixel value P_c is considered as reference and it is compared with all of its neighborhood pixels (Fig. 3 matrix A). If the pixel value P_i of i^{th} neighbourhood is greater than or equal to the center pixel P_c, then 1 is assigned to it else if it is less than P_c then 0 binary value is assigned (Fig. 3 matrix B). After obtaining the binary values for all the neighbourhood pixel cells, LBP decimal code is generated by element wise multiplication of 2^i where i varies from 0 to $N - 1$ with all the binary values according to its neighbourhood position index i and adding together as in Eq. 9 (Fig. 3 matrix E).
Pixel difference is denoted as $pd_i = p_c - p_i$ then,

$$LBP_{code} = \sum_{i=0}^{N-1} s(pd_i) * 2^i \tag{9}$$

$$s(pd) = \begin{cases} 1, & \text{if } pd \geq 0 \\ 0, & \text{if } pd < 0 \end{cases} \tag{10}$$

Eq. 10 is used to eliminate the negative values when p_i is less than p_c.

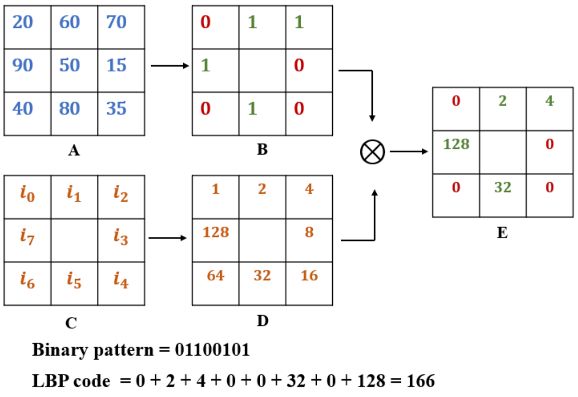

Figure. 3: LBP descriptor computation

HOG and LBP features are of higher dimension and all of its attributes many not significantly contribute in decision making of the classifiers. Hence these features can be transformed to lower dimensional feature space in which the highly correlated features are discarded and only the prominent features with more variance is retained.

3) Principal component analysis feature transform

Principal component analysis is mainly used to transform the high dimensional features to low dimensional feature space by orthogonal projection.

Let F be the feature matrix of M x N where $f_i \in R^N$ be the N dimensional feature vector in i^{th} row and M represents the number of data samples where $i \in (0, M-1)$.
Mean feature vector from the feature matrix is obtained using Eq. 11.

$$f_m = \frac{1}{M} \sum_{i=0}^{M-1} f_i \tag{11}$$

Using f_m, covariance matrix is computed as in Eq. 12

$$\Sigma = \frac{1}{M} \sum_{i=0}^{M-1} (f_i - f_m)(f_i - f_m)^T \tag{12}$$

Eigen equations of the covariance matrix Σ is solved to generate the N x p transformation matrix U, where p represents the number of principal components selected.
Now the high dimensional feature vector $f_i \in R^N$ is projected to R^p using transposed transformation matrix $U^T \in R^{(pXN)}$ to get $d_i \in R^p$ is shown in Eq. 13

$$d_i = U^T(f_i - f_m) \tag{13}$$

Hence the N dimensional feature $f_i \in R^N$ is transformed to $d_i \in R^p$ feature vector where $p < N$. To decide the optimal number of principal components p, experiments are conducted on HOG-PCA and LBP-PCA with different p values and classification accuracy is analyzed to select the p which corresponds to maximum accuracy.

For the comparative analysis along with conventional HOG and LBP features, we use pre-trained VGG-16 deep learning CNN model for feature extraction.

4) Pre-trained VGG16 CNN feature

VGG16 is a popular CNN model with 16 layers (13 convolutional layer and 3 dense layer) which is trained on ImageNet dataset and capable of classifying various objects from 1000 class. Here we have not done the transfer learning, only used the output from last convolutional layer $Conv5-3$ followed by global average pooling layer as the feature vector of length 512, $f_{VGG16} \in R^{512}$.

C. Ensemble classification

1) Support vector machine classifier

SVM classifier tries to maximize the decision boundary or the distance between hyperplane of different classes. Data points closer to the decision boundary decides the shape and position of the hyperplane where classifier tries to maximize the separation distance between these border data points, hence these points are called as support vectors. If the data is linearly separable then SVM fits the hyperplane in the given feature space. In the other case, it uses non-linear kernel function which transforms the data points and projects it into higher dimensional space where the transformed data becomes linearly separable.

2) Artificial neural network classifier

ANN tries to learn the decision boundary using set of input and output neurons through intermediate neurons. It consists of one input layer followed by one or more intermediate layers known as hidden layer and one output layer. Each layer might also additionally have varying number of neurons relying at the model architecture and data complexity, these neurons are interconnected and the strength of connection depends on the weights associated with it. ANN is trained to learn the data pattern by iteratively adjusting the weights and bias of the network. Model training is done by back propagation of error and updating the weights until the network is converged or completes the specified number of iterations.

Consider the Multi Layer Perceptron (MLP) ANN model where $X = \{x_1, x_2, x_3 .... x_n\}$ be the input feature vector of length n and $W = \{w_1, w_2, w_3 .... w_n\}$ be the corresponding weights associated with connection between input feature and the neuron with bias b and activation function f, then the output of the neuron z is given by Eq.14.

$$z = f\left(b + \sum_{i=1}^{n} x_i w_i\right) \tag{14}$$

3) Random forest classifier

It is a bootstrap meta estimator that fits several dataset subsets and uses averaging to increase predicted accuracy and reduce over-fitting. It is built on a variety of decision tree classifiers.

4) Ensemble learning

Classifiers refers to the training algorithm which generates the model or the hypothesis using the given dataset in the training phase. The trained model is used to predict the label of a given data sample during testing. It is observed that using a group of classifiers to derive the hypothesis for data prediction gives more accurate results as compared to the decision taken from the individual classifier. Hence the method of training the multiple classifier to learn multiple hypothesis to solve the same problem is known as ensemble learning.

Some of the popular ensemble methods are bagging, boosting and stacking.
Bagging or Bootstrap Aggregation uses multiple subsets of a training dataset by sampling with replacement known as bootstrap.
Boosting is a meta-algorithm where weak learners are combined to generate a strong learner.
Stacking refers to the process of combining multiple classifiers to form a meta-level classifier, by combining the outputs of the base classifiers.
In this paper, we propose to use the stacking approach by combining SVM, ANN and Random forest classifiers.

Table 1: Comparison of F1-score with different PCA values on SVM, MLP, RF and Ensemble classifiers using HOG features on NTU dataset

PCA	HOG feature											
	SVM			MLP			RF			Ensemble		
	RGB	Depth	RGB-D	RGB	Depth	RGB-D	RGB	Depth	RGB-D	RGB	Depth	RGB-D
100	0.7261	0.7400	0.7183	0.7504	0.6876	0.6868	0.7701	0.7133	0.7370	0.8000	0.7585	0.7779
200	0.8104	0.7565	0.7890	0.7672	0.7623	0.7355	0.7479	0.7664	**0.8195**	0.8301	0.7846	0.8155
300	0.8195	0.7832	0.7991	0.7688	0.7124	0.7384	**0.8087**	0.7221	0.7841	0.8338	0.7779	0.7933
400	0.8039	0.7886	0.8039	0.7718	0.7629	0.7180	0.7603	**0.7739**	0.8048	**0.8436**	0.7944	0.8123
500	0.8339	0.7835	**0.8241**	0.7782	0.7121	**0.8215**	0.7824	0.7146	0.7753	0.8174	0.7688	0.8279
600	0.8386	**0.7939**	0.8187	0.7822	0.7309	0.8126	0.7462	0.7481	0.7895	0.8240	0.7888	**0.8497**
700	**0.8441**	0.7888	0.8072	0.7829	**0.7663**	0.7836	0.7268	0.7128	0.7833	0.8239	**0.8218**	0.8351
800	0.8441	0.7888	0.8072	**0.8175**	0.7519	0.7859	0.6861	0.6696	0.7310	0.8284	0.8046	0.8273

Table 2: Comparison of F1-score with different PCA values on SVM, MLP, RF and Ensemble classifiers using LBP features on NTU dataset

PCA	LBP feature											
	SVM			MLP			RF			Ensemble		
	RGB	Depth	RGB-D	RGB	Depth	RGB-D	RGB	Depth	RGB-D	RGB	Depth	RGB-D
100	0.8212	0.8636	0.8483	0.6181	0.7154	0.6502	0.8604	0.8393	0.8708	0.8315	0.8469	0.8285
200	0.9248	0.8857	0.8999	0.8601	0.8129	0.8628	0.8852	0.8430	0.8955	0.9253	0.8834	0.9049
300	0.9594	0.9048	0.9295	0.9399	0.8692	0.9039	0.9050	0.8479	**0.9102**	0.9496	0.8843	0.9195
400	0.9494	0.9102	0.9342	0.9350	0.8598	0.9398	0.9047	0.8364	0.8999	0.9540	0.9203	0.9350
500	0.9544	0.9049	0.9493	0.9441	0.8792	0.9546	**0.9259**	0.8401	0.8994	0.9544	0.9148	0.9495
600	0.9592	0.9054	0.9394	0.9492	0.8897	0.9398	0.9003	0.8467	0.8952	0.9545	0.8994	0.9495
700	0.9746	**0.9204**	**0.9550**	**0.9599**	**0.9102**	0.9547	0.9100	**0.8707**	0.9102	**0.9750**	0.9252	0.9547
800	**0.9746**	0.9204	0.9550	0.9598	0.9101	**0.9598**	0.9207	0.8455	0.9091	0.9749	**0.9254**	**0.9596**

Table 3: Comparison of F1-score with different PCA values on SVM, MLP,RF and Ensemble classifiers using HOG features on OUHANDS dataset

PCA	HOG feature											
	SVM			MLP			RF			Ensemble		
	RGB	Depth	RGB-D	RGB	Depth	RGB-D	RGB	Depth	RGB-D	RGB	Depth	RGB-D
100	0.7994	0.8238	0.8212	0.7120	0.7615	0.7542	**0.7470**	**0.7820**	**0.8261**	0.8104	0.8414	0.8528
200	0.8318	0.8364	0.8340	0.7548	0.7756	0.8144	0.7326	0.7663	0.7476	0.8372	0.8379	0.8472
300	0.8324	**0.8487**	**0.8343**	0.7818	**0.8014**	0.8255	0.6843	0.7164	0.7428	0.8274	0.8426	0.8244
400	0.8319	0.8334	0.8263	0.7880	0.7898	**0.8341**	0.7237	0.6746	0.6921	0.8348	0.8449	0.8386
500	**0.8349**	0.8412	0.8340	**0.8186**	0.7941	0.8039	0.6878	0.6546	0.7178	0.8300	**0.8516**	**0.8618**
600	0.8265	0.8285	0.8315	0.8092	0.7970	0.8223	0.6685	0.6400	0.7025	**0.8377**	0.8370	0.8350
700	0.8296	0.8253	0.8269	0.7957	0.7977	0.8217	0.6657	0.6163	0.6798	0.8144	0.8264	0.8348
800	0.8243	0.8261	0.8292	0.8100	0.8003	0.8202	0.5978	0.6248	0.6810	0.8076	0.7930	0.8346

Table 4: Comparison of F1-score with different PCA values on SVM, MLP,RF and Ensemble classifiers using LBP features on OUHANDS dataset

PCA	LBP feature											
	SVM			MLP			RF			Ensemble		
	RGB	Depth	RGB-D	RGB	Depth	RGB-D	RGB	Depth	RGB-D	RGB	Depth	RGB-D
100	**0.4985**	**0.5339**	**0.5627**	0.3857	0.4846	0.4793	**0.5062**	**0.5808**	**0.5983**	**0.5304**	**0.5521**	**0.6280**
200	0.4113	0.5268	0.4688	0.2909	0.4267	0.3775	0.3709	0.4888	0.4957	0.4338	0.5048	0.5378
300	0.4625	0.5047	0.4590	0.3740	0.4244	0.4158	0.3988	0.4493	0.4424	0.4603	0.4872	0.5145
400	0.4825	0.5145	0.4476	0.3962	0.4601	0.4530	0.4060	0.4367	0.4195	0.4711	0.5079	0.4889
500	0.4594	0.4921	0.4637	0.4637	**0.4926**	0.4788	0.3511	0.4119	0.4234	0.4893	0.5225	0.5116
600	0.4573	0.4883	0.4532	0.4970	0.4699	0.4979	0.3598	0.3533	0.4471	0.5053	0.5033	0.4916
700	0.4595	0.4770	0.4455	0.4836	0.4654	0.5311	0.3887	0.4177	0.4045	0.4984	0.5197	0.5329
800	0.4577	0.4757	0.4510	**0.5010**	0.4685	**0.5369**	0.3540	0.3620	0.3773	0.4769	0.4948	0.4846

Table 5: Comparison of F1-score on SVM, MLP, RF and Ensemble classifiers using VGG features on NTU and OUHANDS dataset

Dataset	VGG feature											
	SVM			MLP			RF			Ensemble		
	RGB	Depth	RGB-D	RGB	Depth	RGB-D	RGB	Depth	RGB-D	RGB	Depth	RGB-D
NTU	0.6111	0.5006	0.6116	0.744	0.694	0.8246	0.8793	0.7403	0.8594	0.7985	0.7186	0.8398
OUHANDS	0.8751	0.7732	0.8982	0.8557	0.8687	0.9095	0.8587	0.8184	0.9013	0.8849	0.8797	0.9377

IV. Experimental Results

In order to appraise the proposed HGR method, we have evaluated our framework with two publicly used datasets as described below.

1) NTU Hand Digits dataset - It consists of 1000 RGB color images in cluttered backgrounds and their corresponding depth maps collected using Kinect device. Dataset consists 10 hand gestures representing decimal digits $0 - 9$, gestures are captured by 10 subjects performing 10 gestures with 10 samples per each gestures. Hence the total dataset of 1000 images are resized to 256 x 256 image resolution and split in the ratio of $80 : 20$, where 800 images is used to train the classification model and 200 images are used in evaluation.

2) OUHANDS dataset - It consists of RGB color images and its corresponding depth images captured from 23 subjects performing 10 unique gestures with varying shape, size and complex background. Train dataset of 2000 images are resized to 256 x 256 image resolution is split in the $80 : 20$ ratio, 1600 images are used train classifier model and 400 images are used in evaluation. This dataset also consists a separate testing set of 1000 images from different subjects, Since the conventional features needs at least few samples of the test set to be seen in training phase, we have not considered the test set for evaluation and bench-marking.

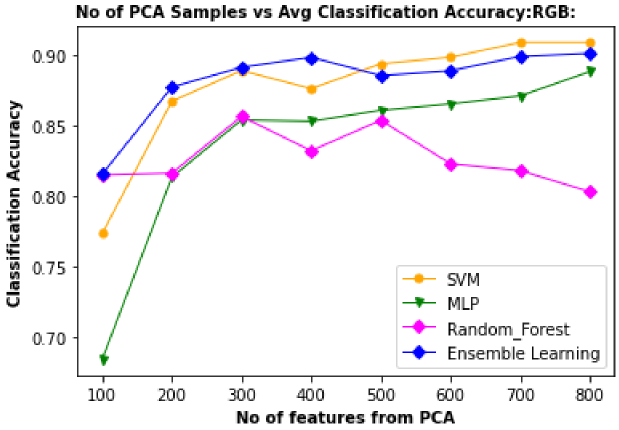

Figure. 4: The average classification accuracy of each classifier, in classifying features from RGB images of NTU dataset, with various PCA components.

The results obtained on each dataset is listed in Tables 1-5. Table 1, illustrates the experimental outcome i.e. F1 score of the proposed framework based on HOG features, considering RGB transformed to gray image, Depth and concatenated RGB-D features. The computed HOG features are passed on to PCA. Where PCA performs the dimensionality reduction and for experimental analysis, we have considered maximum of 800 principal components. The RGB, Depth and RGB-D features are fed as an input to SVM, MLP, Random forest and ensembling classifiers. Each table records the F1 score outcome by the proposed HGR framework. In case of SVM classifier, for RGB image, 700 features results in highest F1 score of 0.84%. In case of Depth image, for 600 features,

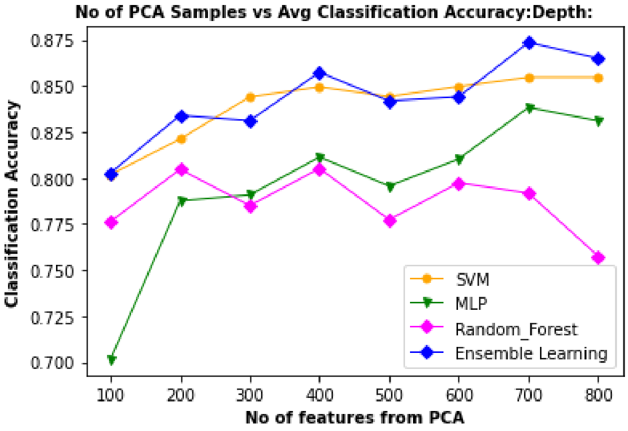

Figure. 5: The average classification accuracy of each classifier, in classifying features from Depth images of NTU dataset, with various PCA components.

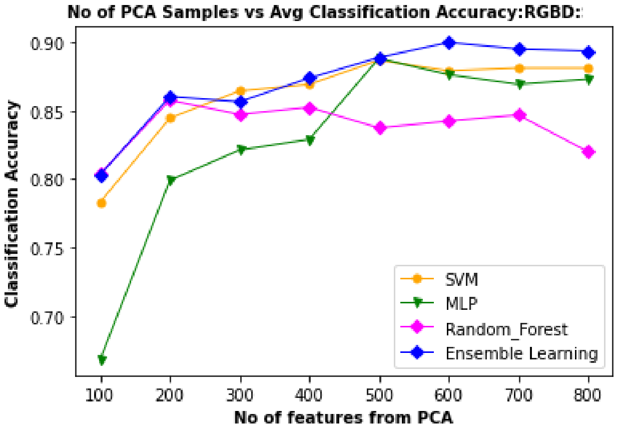

Figure. 6: The average classification accuracy of each classifier, in classifying RGB-D features of NTU dataset, with various PCA components.

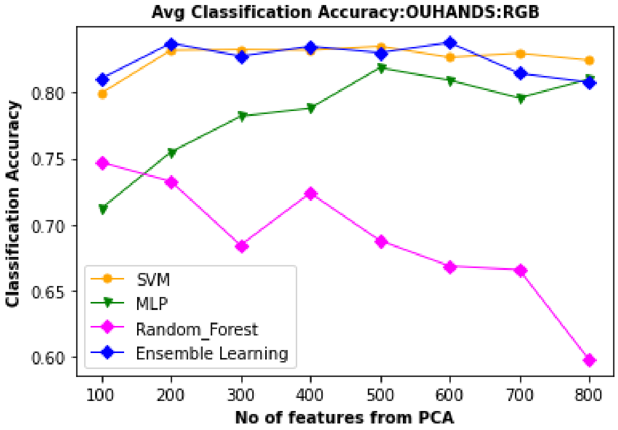

Figure. 7: The average classification accuracy of each classifier, in classifying features from RGB images of OUHANDS dataset, with various PCA components.

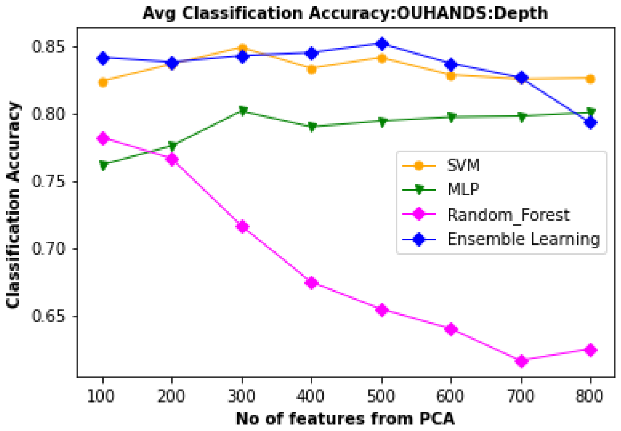

Figure. 8: The average classification accuracy of each classifier, in classifying features from Depth images of OUHANDS dataset, with various PCA components.

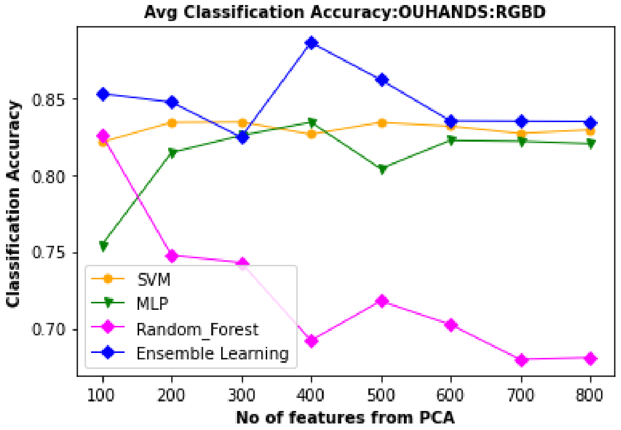

Figure. 9: The average classification accuracy of each classifier, in classifying RGB-D features of OUHANDS dataset, with various PCA components.

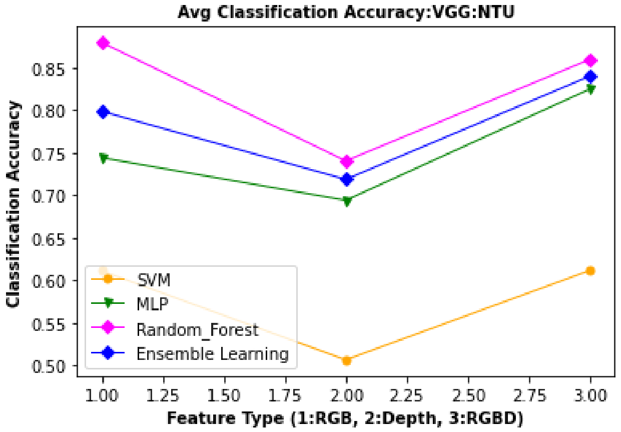

Figure. 10: Average classification accuracy of each classifier, in classifying RGB, Depth and RGB-D features of NTU dataset.

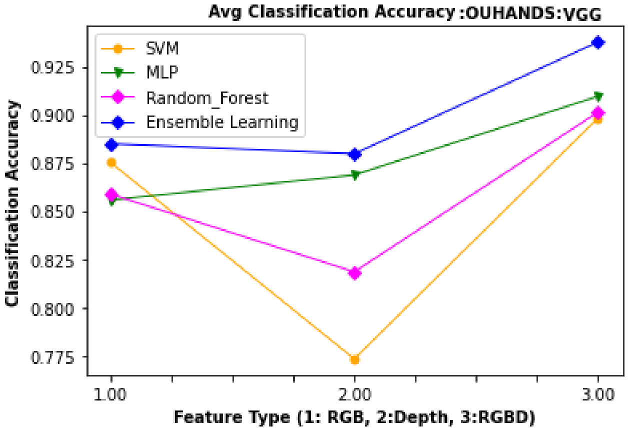

Figure. 11: Average classification accuracy of each classifier, in classifying RGB, Depth and RGB-D features of OUHANDS dataset.

Table 6: Comparison of Hand gesture recognition accuracy on NTU dataset

Approach	Accuracy
Thresholding Decomposition+FEMD [19]	90.6
Near-convex Decomposition+FEMD [19]	93.9
Hand dominant line + SVM [20]	91.1
HOG [21]	93.1
H3DF [21]	95.5
VS-LBP + SVM [22]	95.9
Shape context without bending cost [23]	92.2
Shape context with bending cost [23]	85.375
Skeleton matching [23]	90.475
Ploar + DNN [24]	94.2
SP-EMD [25] (shape only)	96.5
SP-EMD [25]	97.2
Proposed LBP-PCA ensemble	**97.50**

0.79% and in case of RGB-D for 500 features, 0.82% F1 score is the outcome by the proposed framework. The ensemble classifier results in highest F1 score compared to SVM, MLP and RF classifiers. Similar observations can be drawn from other tables. In majority of cases, the ensembling classifier outcomes highest F1 score.

The experimental outcome of the proposed framework are depicted in Fig 4 - Fig 11. Figure 4, depicts the average classification accuracy of each classifier in classifying RGB images of NTU dataset, with increased set of features from 100 to 800. Figure 4 depicts that, in case of NTU dataset, initially, the MLP outcomes a classification accuracy of 61%, as the number of features increases, SVN, MLP and Ensembling of classifiers results in close to 90% accuracy. Random forest classifier demonstrates the decreasing accuracy, as the number of features reaches close to 500. Similar observations can be drawn from figures 5 and 6. The recognition accuracy of the proposed framework in case of OUHANDS dataset is depicted in figures 7-9.

In case of OUHANDS dataset, a different trend is observed as compared to NTU dataset. As the number of features increases, the recognition accuracy decreases. The ensembling classifier starts with close to 75% accuracy and slightly decreases as the number of features increases. Figure 10 and 11 illustrates the recognition accuracy of the feature outcome from the VGG-16 model and using SVM, MLP and RF as classifier.

In both the datasets, the recognition accuracy initially decreases and shows an increasing trend as the number of feature increases. The RF, MLP classifiers outcomes higher recognition accuracy, in case of NTU and OUTHANDS dataset respectively. As depicted in Table 6, we have compared the proposed framework with similar methods which are based out of machine learning techniques. The proposed framework achieves state-of-the-art outcome of 97.50%, which confirms the efficiency of the proposed framework which is based on traditional classifiers and ensembling techniques.

V. Conclusion

The main pitfall in static hand gesture recognition is the noise in the critical segments and dealing with the image background. To overcome these challenges, in this paper, we have proposed static hand gesture recognition framework using RGB-D data. The proposed methods uses machine learning classifiers, deep learning models and ensembling techniques to classify the gesture. We have thoroughly evaluated the proposed method on two widely used datasets NTU and OUHANDS.

HOG and LBP features are extracted on gray converted RGB image and the depth image after segmenting the background. Further PCA was used to reduce the feature dimensionality. These features are used to train SVM, RF, MLP and ensemble classifier, where F1 score outcome of all the features and classifier combination is evaluated. For further comparison, the proposed method is evaluated on deep VGG-16 model features also. From the experimental results, we can infer that combined RGB-D multi-modal features with combination of ensembled classifier gives better accuracy in most of the cases as compared to individual classifier on uni-modal data. In future work, we are considering one-shot and few shot learning, in which we are trying to achieve better F1-score with lesser training samples of hand gesture images.

References

[1] M. Kawulok, J. Kawulok, J. Nalepa, and B. Smolka, "Self-adaptive algorithm for segmenting skin regions," *EURASIP Journal on Advances in Signal Processing*, vol. 2014, no. 170, pp. 1–22, 2014. [Online]. Available: http://sun.aei.polsl.pl/~mkawulok/gestures/

[2] G. Thippa Reddy, G. Srivastav, and G. Liyanage, "Hand gesture recognition based on a harris hawks optimized convolution neural network," *Computers and Electrical Engineering*, vol. 2022, no. 170, pp. 1–8, 2014. [Online]. Available: http://sun.aei.polsl.pl/~mkawulok/gestures/

[3] S. Yuying, C. Sujie, L. Ming, L. Siying, P. Yisen, and G. Xiaojun, "Flexible strain sensors for wearable hand gesture recognition: From devices to systems," *Computers and Electrical Engineering*, vol. 1002, no. 170, pp. 1–17, 2022. [Online]. Available: https://onlinelibrary.wiley.com/doi/epdf/10.1002/aisy.202100046

[4] S. Jaya Prakash, S. Suraj Prakash, A. Samit, and P. Sarat Kumar, "Rbi-2rcnn: Residual block intensity feature using a two-stage residual convolutional neural network for static hand gesture recognition," *Signal, Image and Video Processing*, vol. 1007, no. 170, pp. 1–17, 2022. [Online]. Available: https://link.springer.com/article/10.1007/s11760-022-02163-w

[5] B. Gopa, V. Monu, C. Mahesh, and V. Santosh Kumar, "Hyfinet: Hybrid feature attention network for hand gesture recognition," *Multimedia Tools and Applications*, vol. 1007, no. 170, pp. 1–17, 2022. [Online]. Available: https://link.springer.com/article/10.1007/s11042-021-11623-3

[6] M. Abavisani, H. Joze, and V. Patel, "Improving the performance of unimodal dynamic hand-gesture recognition with multimodal training." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1165—-1174, 2019. [Online]. Available: https://arxiv.org/abs/1812.06145

[7] T. Junaid, I. Amir, A. Ammar, R. Hameedur, and A. Fayadh, "Hevc's intra mode process expedited using histogram of oriented gradients," *Journal of Visual Communication and Image Representation*, vol. 88, pp. 1165—-1174, 2022. [Online]. Available: https://doi.org/10.1016/j.jvcir.2022.103594

[8] L. Qiwu, S. Jiaojiao, Y. Chunhua, S. Olli, and L. Li, "Hevc's intra mode process expedited using histogram of oriented gradients," *Journal of Visual Communication and Image Representation*, vol. 132, pp. 1165—-1174, 2022. [Online]. Available: https://doi.org/10.1016/j.patcog.2022.108901

[9] L. Houssem and N. Mahmoud, "Hand gesture recognition system based on lbp and svm for mobile devices," *International Conference on Computational Collective Intelligence*, vol. 11683, pp. 1—-13, 2019. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-28377-3_23

[10] K. Kaplan, K. Yılmaz, and K. Melih, "Brain tumor classification using modified local binary patterns (lbp) feature extraction methods," *Medical Hypotheses*, vol. 139, pp. 1—-13, 2020. [Online]. Available: https://doi.org/10.1016/j.mehy.2020.109696

[11] S. K. Alok Kumar, S. Subodh, and S. Rajeev, "Multi-view human activity recognition based on silhouette and uniform rotation invariant local binary patterns," *Multimedia Systems*, vol. 23, pp. 451—-467, 2017. [Online]. Available: https://link.springer.com/article/10.1007/s00530-016-0505-x

[12] A. Barbhuiya, R. Karsh, and R. Jain, "Cnn based feature extraction and classification for sign language," *Multimed. Tools Appl.*, vol. 80, pp. 3051—-3069, 2021.

[13] R. Golash and Y. Jain, "Low-cost design of vision-based natural user interface via dynamic hand gestures," *International Journal of Computer Information Systems and Industrial Management Applications*, vol. Volume 13, pp. 031–042, 03 2021.

[14] L. Yuan, W. Xinggang, L. Wenyu, and F. Bin, "Deep attention network for joint hand gesture localization and recognition using static rgb-d images," *Information Sciences*, vol. 441, pp. 66–78, 2018. [Online]. Available: https://doi.org/10.1016/j.ins.2018.02.024

[15] F. Bin, L. Huifen, W. Yongjiang, W. Xinggang, and L. Wenyu, "Robust hand gesture recognition based on enhanced depth projection maps (edpm)," *2016 8th International Conference on Wireless Communications & Signal Processing (WCSP)*, vol. 80, pp. 1–5, 2016.

[16] S. Prachi and A. Radhey Shyam, "Depth data and fusion of feature descriptors for static gesture recognition," *IET Image Processing*, vol. 14, pp. 1—-12, 2020. [Online]. Available: https://digital-library.theiet.org/content/journals/10.1049/iet-ipr.2019.0230

[17] S. Jaya Prakash, P. Allam Jaya, P. Pawel, and S. Saunak, "Real-time hand gesture recognition using fine-tuned convolutional neural network," *Sensors*, vol. 22, pp. 1—-14, 2022. [Online]. Available: https://digital-library.theiet.org/content/journals/10.1049/iet-ipr.2019.0230

[18] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, vol. 1. Ieee, 2005, pp. 886–893.

[19] Z. Ren, J. Yuan, and Z. Zhang, "Robust hand gesture recognition based on finger-earth mover's distance with a commodity depth camera," in *Proceedings of the 19th ACM international conference on Multimedia*, 2011, pp. 1093–1096.

[20] Y. Wang and R. Yang, "Real-time hand posture recognition based on hand dominant line using kinect," in *2013 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*. IEEE, 2013, pp. 1–4.

[21] C. Zhang, X. Yang, and Y. Tian, "Histogram of 3d facets: A characteristic descriptor for hand gesture recognition," in *2013 10th IEEE international conference and workshops on automatic face and gesture recognition (FG)*. IEEE, 2013, pp. 1–8.

[22] A. I. Maqueda, C. R. del Blanco, F. Jaureguizar, and N. García, "Human–computer interaction based on visual hand-gesture recognition using volumetric spatiograms of local binary patterns," *Computer Vision and Image Understanding*, vol. 141, pp. 126–137, 2015.

[23] Z. Ren, J. Yuan, J. Meng, and Z. Zhang, "Robust part-based hand gesture recognition using kinect sensor," *IEEE transactions on multimedia*, vol. 15, no. 5, pp. 1110–1120, 2013.

[24] H. V. Vo, T. T. Son, and Q. N. Ly, "Hybrid deep network and polar transformation features for static hand gesture recognition in depth data," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 5, 2016.

[25] C. Wang, Z. Liu, M. Zhu, J. Zhao, and S.-C. Chan, "A hand gesture recognition system based on canonical superpixel-graph," *Signal Processing: Image Communication*, vol. 58, pp. 87–98, 2017.

Author Biographies

Dayananda Kumar N.C received B.E. degree in electronics and communication engineering in the year 2010 and M.Tech in Signal Processing in the year 2014 both from Visvesvaraya Technological University, India. Currently he is pusuing Ph.D along with his work as software developer. He has 10 years of experience in software development. His research interests include image processing and computer vision.

K.V. Suresh received B.E. degree in electronics and communication engineering in the year 1990 and M.Tech in industrial electronics in the year 1993 both from the University of Mysore, India. From March 1990 to september 1991, he served as a faculty in the department of electronics and communication engineering, Kalpataru Institute of Technology, Tiptur, India. Since 1993, he is working as a faculty in the department of electronics and communication engineering, Siddaganga Institute of Technology, Tumkur, India. He completed Ph.D in the department of electrical engineering, Indian Institute of Technology, Madras in 2007. His research interests include signal processing and computer vision.

Dinesh R is Professor in the Department of Information Science and Engineering, Jain University, Bangalore, Karnataka, India. He obtained Ph.D in Computer Science and Engineering from Mysore University. He is having 15 years of experience in software development, academics and research. He is expertise in image processing and pattern recognition research field. He has several patents for his research work. He has published research papers in the international journals and in the conference proceedings.