A

Dissertation

On

Sign Language Dynamic Gestures Recognition using Depth Data

Submitted in partial fulfilment of the requirements
for the award of degree

Of

Master of Technology
in
Computer Science and Engineering

Submitted by

Mudit Goyal
Enrolment No. – 14535029

Under the guidance of

Dr. R. Balasubramanian

Department of Computer Science and Engineering,

Indian Institute of Technology, Roorkee

Roorkee – 247667, India

ABSTRACT

In this report we have proposed a framework for sign language dynamic gestures recognition from depth sequences. For feature representation two different set of features are extracted. First one is gradient local auto correlation features from the depth motion maps and to incorporate the loss of temporal information which is there in depth motion maps, the other set of features extracted is HON4D (Histogram of oriented 4D normal). A new framework for fusing the features at decision level using classifier ensemble of three 2-layer feed forward neural networks is been proposed . The proposed framework is tested on two datasets MSRGesture3D and ISL3D dataset. The ISL3D dataset is created by us having 12 dynamic Indian Sign Language gestures. The recognition accuracies achieved on the two datasets are: 96.99% on MSRGesture3D dataset and 81.38% on ISL3D dataset.

ACKNOWLEDGEMENT

This research work is greatly facilitated by my supervisor Dr.R.Balasubramanian and my wholeheartedly thanks are due firstly to him for giving me such a magnificent opportunity to join his enthusiastic research group.

I offer my special gratitude to Dr. Ankush Mittal for his guidance and motivation throughout this project.

I am also very grateful to my friend Pushkar Shukla for providing his valuable suggestions and help in carrying out this research work. Lastly I would like to thank my family members and friends for their patience and cooperation while doing this work.

Place: Roorkee Mudit Goyal

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

Sign language recognition system aims to understand the meaning of the gestures and postures in a sign language communication. Gestures involves the movement of fingers, hand, face, head etc. to convey some useful information between to people communicating with each other. Sign language is mostly used by deaf and dumb people as a mode of communication. As, sign language is learned only by the deaf and dumb people and usually it is not known to normal people, so it becomes a challenge for communication between a normal and hearing impaired person. So, sign language recognition system aims to bridge that gap and makes it possible for the normal and hearing impaired person to communicate with each other. Sign language recognition system takes as an input the gestures made by the hearing impaired person, understands and translates them into a language understood by the hearing person.

1.1Sign Linguistics

Sign language communication not just only involves hand gestures but it also involves Non-Manual signals for communication which involves facial expressions, head movements, torso movements, body postures etc. So, [1] divides the sign linguistics in the following three components:

1.1.1. Manual Features

Manual features involve the gestures made with the hands. The hand shapes and their motion together convey the meaning. Manual features in a sign language are dominant as most of the meaning is conveyed through them. According to the sign language recognition systems survey by [2] most of the work in sign language recognition system is been done mainly focusing upon the manual features only. In [3] the hand gestures are classified into dynamic and static gestures, and the dynamic gestures are further classified into sub categories. Static hand gestures are the orientation and position of hand in space without any movement with time and if movement is there it becomes a dynamic gesture. In our work we have focused only on the manual features of sign language communication. Figure 1.1 shows the 12 dynamic gestures of American Sign Language, and Figure 1.2 shows the 26 static gestures (from A to Z) of Indian Sign Language.

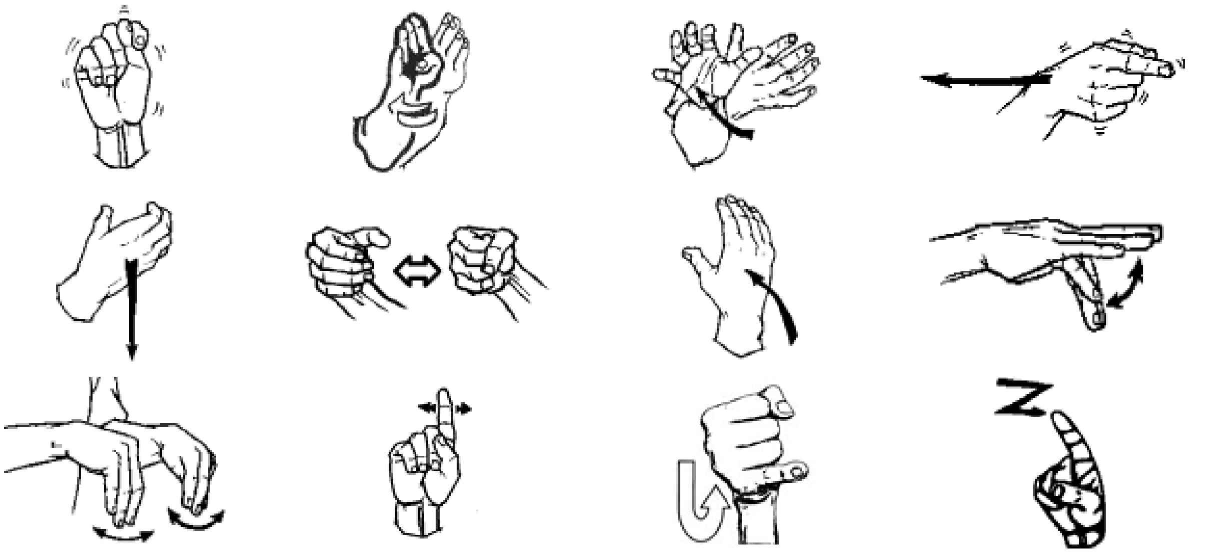


Figure 1.1 12 Dynamic gestures from American Sign Language [4]

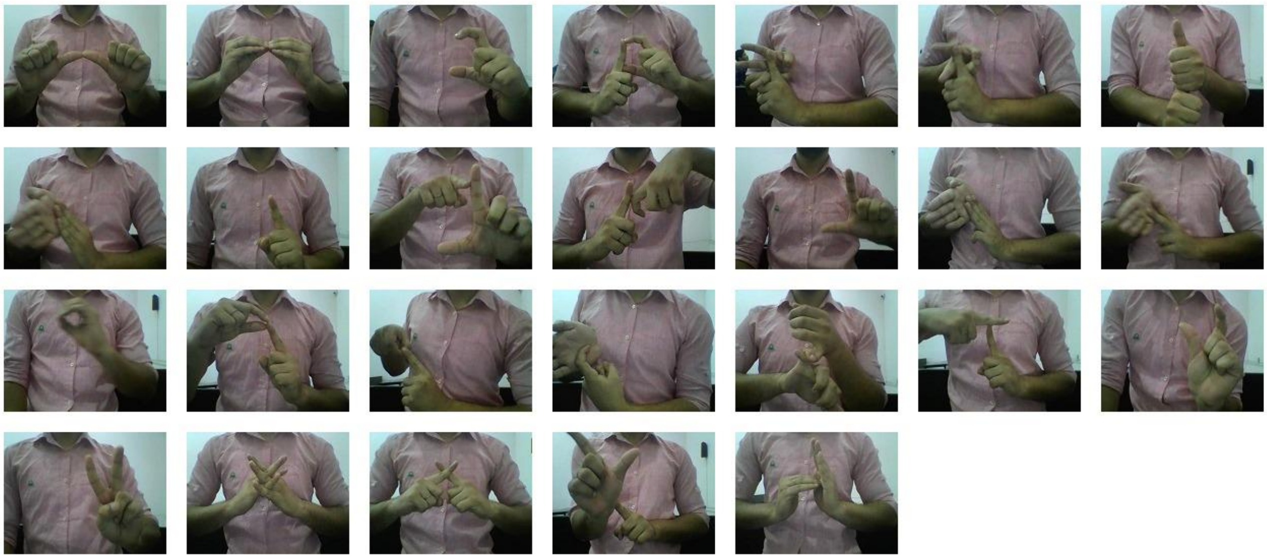


Figure 1.2 26 Static gestures (A to Z) from Indian Sign Language

1.1.2. Non-Manual Features

Non-Manual features includes facial expressions, body postures, head movements, torso movements etc. Non-Manual signs can form a part of sign or can modify the meaning of a sign, ex eyebrow position can determine the type of question. Some sign are distinguished only by the non-manual signs as they share a common manual signal. Hence, to correctly recognize the meaning of a sign recognizing non-manual sigs is also very important.

1.1.3. Finger Spelling

Finger Spelling is used when the sign is not known by the recipient or the signer. So, by the finger spelling the local spoken word for the sign is spelt explicitly. For recognizing the finger spelling the hand shapes should be dealt carefully.

1.2 Sign language gestures recognition using depth data

Most of the research in the field of gesture recognition till now was based on recognizing the gestures by the use of conventional RGB cameras. But there were many limitations with these traditional techniques. As explained in [5], the major limitations in gesture recognition using traditional RGB cameras are as follows:

1. Low level challenges which includes varying illumination conditions, shadows and cluttered backgrounds.

2. View changes, the same gesture can give different appearances from different perspectives.

3. Scale variances

So, with the traditional RGB cameras there are a lot of limitations. In the systems using RGB cameras for data acquisition needs to focus a lot on the background subtraction and still because of its limitations cannot achieve good results. With the recent emergence of the cheap depth sensors like Microsoft Kinect (Figure 1.3 shows a Microsoft Kinect Sensor) has led to their widespread use in gesture recognition. Along with the color image unlike traditional RGB cameras these devices also gives a depth image of the scene. The depth image is insensitive to the lighting conditions. Figure 1.4 shows a color image and its corresponding depth image produced by the Kinect sensor. Thus, using a depth image makes the background subtraction and hand and body segmentation processes very easy. And using depth maps we can also generate skeleton joint points which could be useful for gesture recognition tasks. So, depth data from depth sensors can help to make a robust sign language recognition system.

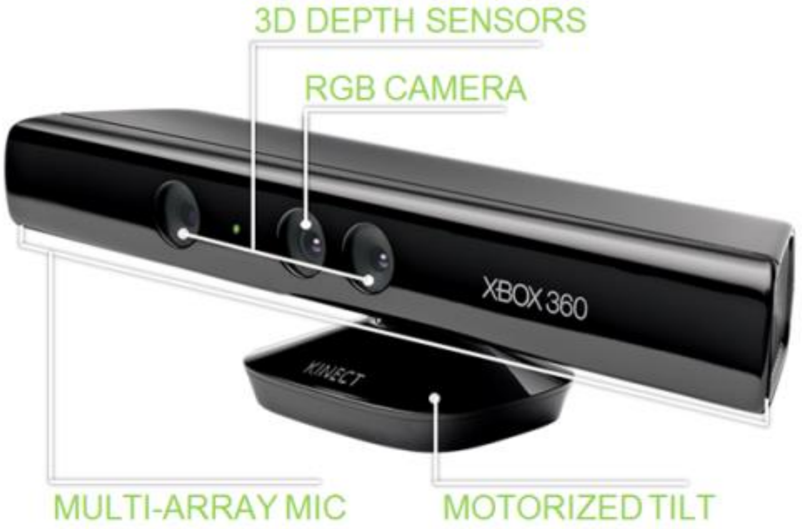
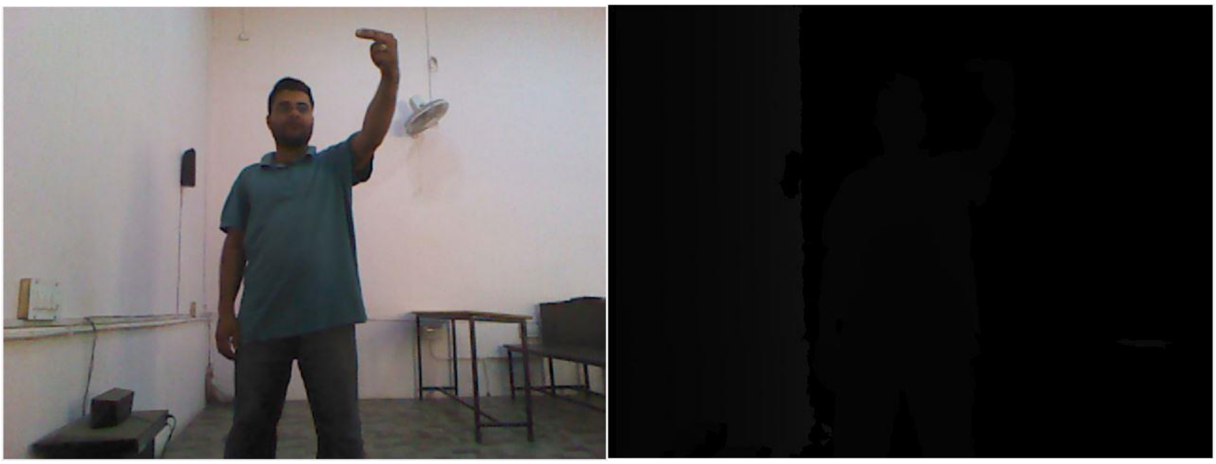


Figure 1.3 Microsoft Kinect Sensor



(a) (b)

Figure1.4 (a) Color image (b) Depth image corresponding to color image in (a)

1.3System Overview

In this work we have a designed a system to recognize dynamic gestures from American and Indian sign language using depth sequences of the gestures. First we have segmented the human body from the background. Then we created the depth motion maps and extracted gradient auto correlation features from them and also histogram of oriented 4d normal features are extracted from the depth sequences which are then fed to the neural network classifier separately and finally a weighted fusion is done using a third neural network for the final results. We tested our system on two datasets one is MSRGesture3D dataset at the other one is the ISL3D (Indian Sign Language) dataset which is generated by us.

1.4Organization of Report

The rest of the report is organized as follows: Chapter 2 consists of the Literature Review of the different methods used for Feature Extraction and classification for gesture recognition. Chapter 3 discuss the proposed system in detail, the feature extraction and the classification techniques used in the proposed system. Chapter 4 presents the datasets and the experiment results. Chapter 5 finally concludes the report.

2. LITERATURE REVIEW

In the pipeline of gesture recognition feature extraction and classification are the main steps. Research in gesture recognition using depth data has explored various feature extraction and classification techniques. The various representations used for depth sequences for gesture recognition includes silhouette's and occupancy features [4], random occupancy patterns [10], depth motion maps based approaches [11,12,13], Histogram of depth gradients[14], histogram of oriented 4D normals[8], super normal vectors[15], motion history and binary shape templates[16] . The various classification techniques used in gesture recognition includes action graph [4], support vector machine [8, 10, 11, 16], random decision forests [14], extreme learning machine [12, 13]. Here in this report we will review some of the major feature extraction and classification techniques used for gesture recognition from depth data.

In [4], a real time system for hand gesture recognition system has been proposed. They have used two different kind of features silhouette's features and occupancy features. To derive the cell occupancy feature they have divided the hand image into a uniform grid and for each cell in the grid they have calculated the occupancy or the occupied area of the hand in the cell. Then they have combined the depth value and the occupancy in a single vector. To derive the silhouette feature they have divided the image into a lot of fan like sectors and for each sector they have calculated the average distance from the hand contour in the segment to the origin. Finally they have concatenated all these distances from all the sectors into a single vector. Figure 2.1 represents the feature extraction used in [4]. For the classification they have used action graph which is a modification of HMM (Hidden Markov Models).

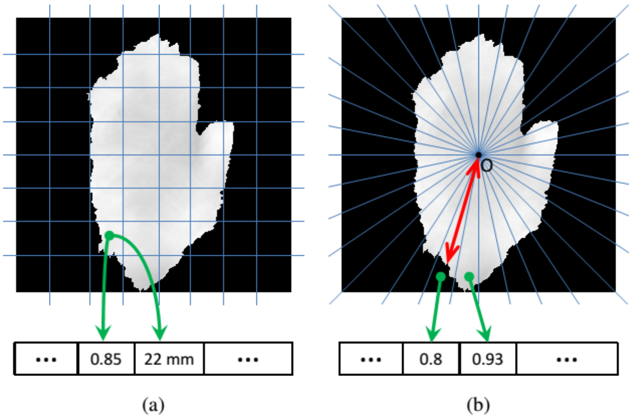


Figure 2.1 Feature Extraction (a) Cell Occupancy Feature (b) Silhouette Feature [4]

In [10], semi local features random occupancy patterns are extracted. In their proposed approach they have treated the depth sequence as a 4D volume and then have defined the value of the pixel in this 4D volume to be 0 or 1. To construct the features they have employed a four dimensional random occupancy patterns and their value is defined by the soft-threshold sum of the pixels in a sub volume defined as follows:

$$O_{xyzt} = \delta(\sum_{q \epsilon bin_{xyzt}} I_q) \tag{2.1}$$

Here, Iq =1 if the point cloud has a point in the location q and otherwise Iq =0. Figure 2.2 represents the overview of the proposed method in [10].

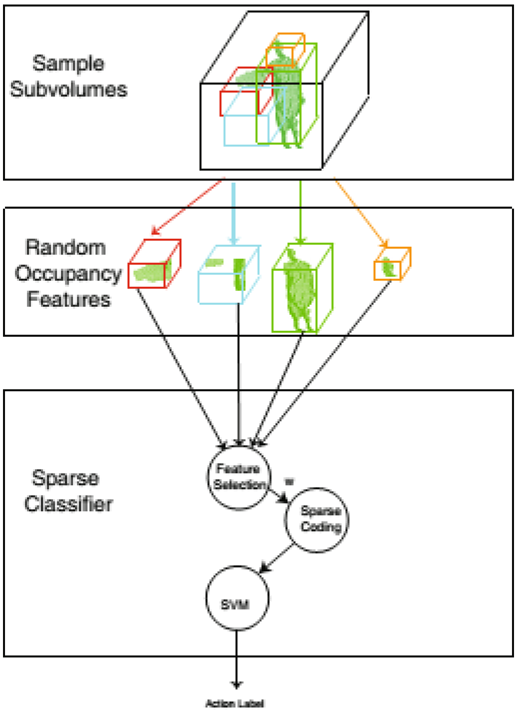


Figure 2.2 Framework of the proposed method in [10]

In [11, 12, 13] depth motion maps of the depth sequence are created. Each frame in a depth sequence is projected on to three orthogonal planes giving three projection views of a depth frame which is front, side and top. In all the three projection views depth motion map is created by taking the absolute difference of the consecutive depth frame. Thus, a 3D gesture is represented by three 2D depth motion maps. In [11] HOG (Histogram of oriented gradients) features are extracted from depth motion maps, while in [12] LBP (Local binary patterns) features are extracted from depth motion maps. Figure 2.3

represents the framework of the proposed feature extraction method used in [11].For classification Support vector machine is used in [11] while extreme learning machine is used in [12, 13].

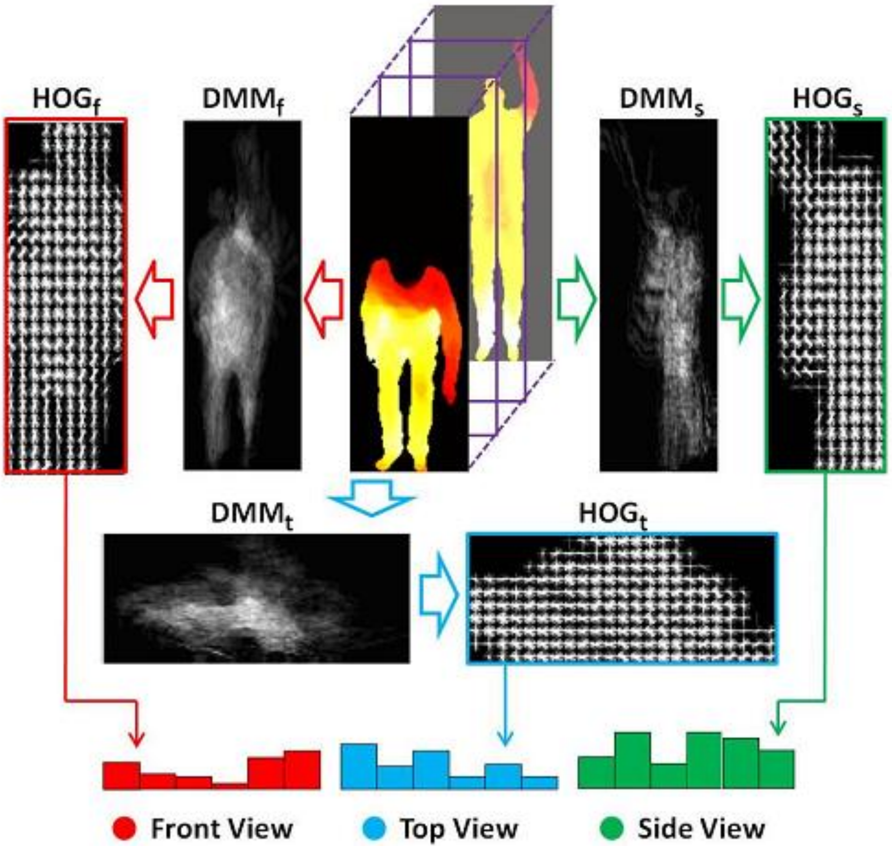


Figure 2.3 Framework of computing HOG from Depth Motion Maps in [11]

In [14], have a computed a feature vector using many independent local features. The spatio-temporal variations of depth gradients is encoded at a space time location in the gesture sequence is encoded in the feature vector. They have also computed a local 3D joint position difference histogram. They have used random decision forests to retain only the discriminative features in the feature set.

In [15], a polynormal by clustering the hypersurface normals in a depth sequence is formed to jointly characterize the shape and local motion information. An adaptive spatio-temporal pyramid is also introduced by subdividing the depth video in a set of space-time grids to globally capture the spatial and temporal orders. They have also proposed a scheme of aggregating the low level polynormals into a super normal vector (SNV) which is a simplified version of the Fisher kernel representation.

In [16], a gesture depth sequence is divided into temporally overlapping blocks. Then each block which is the set of image frames is used to generate motion history templates and binary shape templates. Each frame in the depth sequence is projected on to three orthogonal planes and then for each view Motion History Templates are made by stacking the difference of consecutive frames in the sequence in a weighted manner. From the motion history templates histograms of oriented gradients features are extracted and concatenated into a single feature vector. Finally, a Radial Kernel based SVM classifier is used for classification. Figure 2.4 represents the process flow of the proposed method in [16].

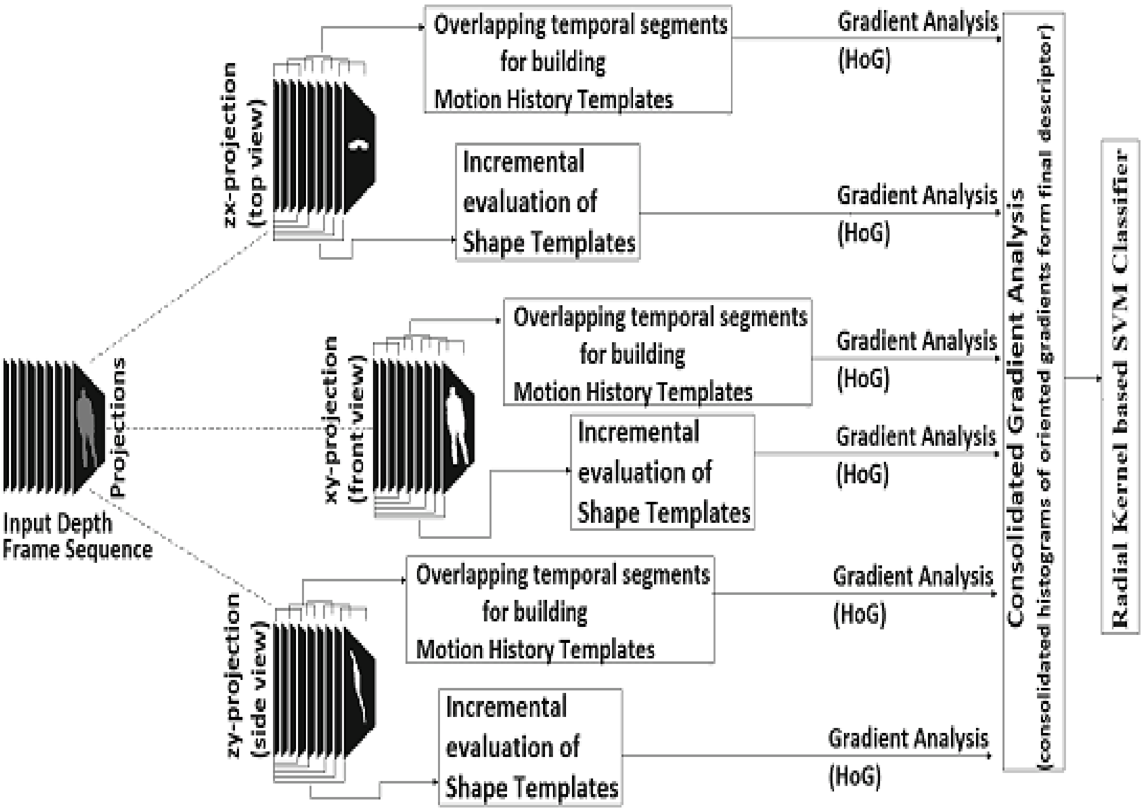

Input Depth Frame Sequence
Projections
zx-projection (top view)
xy-projection (front view)
zy-projection (side view)
Overlapping temporal segments for building Motion History Templates
Incremental evaluation of Shape Templates
Gradient Analysis (HoG)
Consolidated Gradient Analysis
(consolidated histograms of oriented gradients form final descriptor)
Radial Kernel based SVM Classifier

Figure 2.4 Process flow of the system using MHI and HOG in [16]

Table 2.1 Summary of techniques and datasets used in literature

Authors	Year of Publication	Feature Extraction Technique	Classifier	Datasets Used
Z.Zhang, Z.Liu et al. [4]	2012	Silhouette and Occupancy Features	Action Graph	MSRGesture3D
Wang, Ziang et al. [10]	2012	Random Occupancy Patterns	SVM(Support Vector Machine)	MSRAction3D MSRGesture3D
Yang, Zhang et al.[11]	2012	DMM and HOG	SVM(Support Vector Machine)	MSRAction3D MSRGesture3D
C.Chen, Jafari et al.[12]	2015	DMM and LBP	ELM(Extreme Learning Machine)	MSRAction3D MSRGesture3D
C Chen et al.[13]	2015	GLAC	ELM(Extreme Learning Machine)	MSRAction3D MSRGesture3D

Rahmani, Hossein, et al. [14]	2014	Histogram of depth gradients	Random Forests	MSRAction3D MSRGesture3D MSR Daily Activity3D
Oreifej, Omar et al.[8]	2013	HON4D	SVM	MSRAction3D MSRGesture3D MSR Daily Activity3D 3D Action Pairs
Yang, Xiaodong et al. [15]	2014	SNV (Super Normal Vector)	SVM	MSRAction3D MSRGesture3D MSR Daily Activity3D 3D Action Pairs
Jetley, Saumya et al. [16]	2014	Motion History and Binary Shape Templates	SVM	MSRAction3D MSRGesture3D 3D Action Pairs UT- Kinect

3. PROPOSED METHOD

Our proposed method consists of the following main steps:

1. Segmentation or Background Subtraction
2. Creation of depth motion maps
3. Extracting Gradient local auto correlation features from depth motion maps
4. Extracting HON4D features from depth sequences
5. Classification and Weighted Fusion of Classifier Outcomes

3.1 Segmentation or Background Subtraction

Human body who is performing the hand gesture is our region of interest in a depth frame. Thus, for each input depth frame of a gesture we have first segmented the human body from the background. We have used two datasets 1.MSR3D Gesture dataset 2. ISL3D Gesture dataset, dataset 1 which is a benchmark dataset by Microsoft have already given the segmented hand region in the depth sequences. So, for dataset 1 we did not performed this segmentation step.

In a depth frame each pixel represents the depth value. With the assumption that there is only a single person in front of the camera performing the gesture and there is no object between the Kinect camera and the person performing the gesture , the depth values of the human body would be less than the other objects in the background. This assumption is reasonable in many practical situations. Thus, the human body part is segmented using thresholding. We found the second minimum in a depth frame and added it with a certain threshold value. And finally the region lying in the range is extracted. Figure 3.1 represents an input depth image of a gesture and Figure 3.2 represents its corresponding segmented image. The input depth frame is of the dimension of 480x640 and the segmented frame is of the dimension of 480*320.

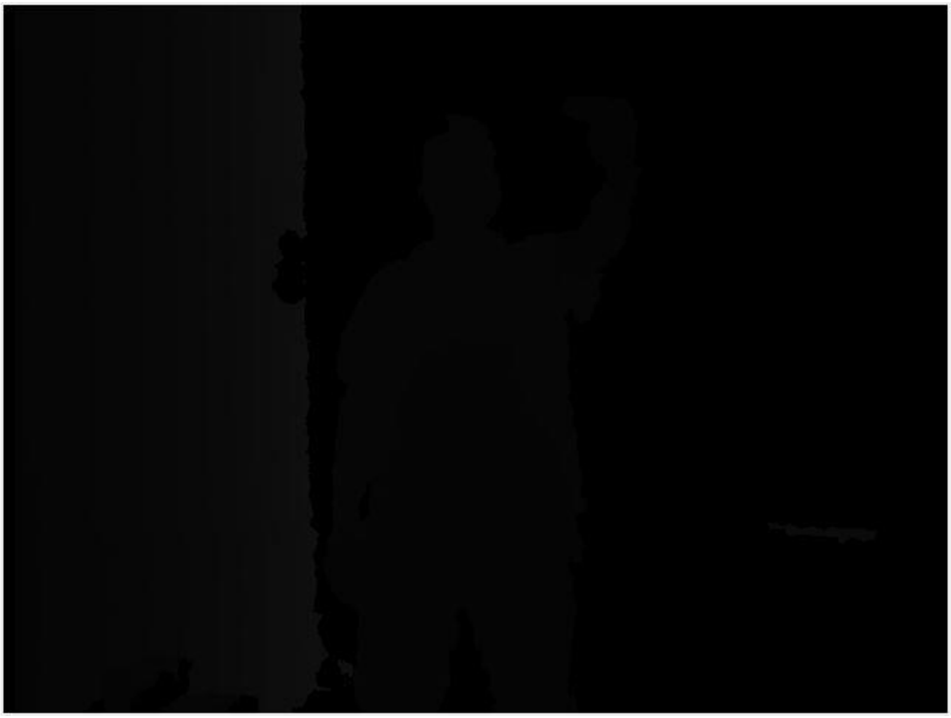


Figure 3.1 Input depth frame of a gesture

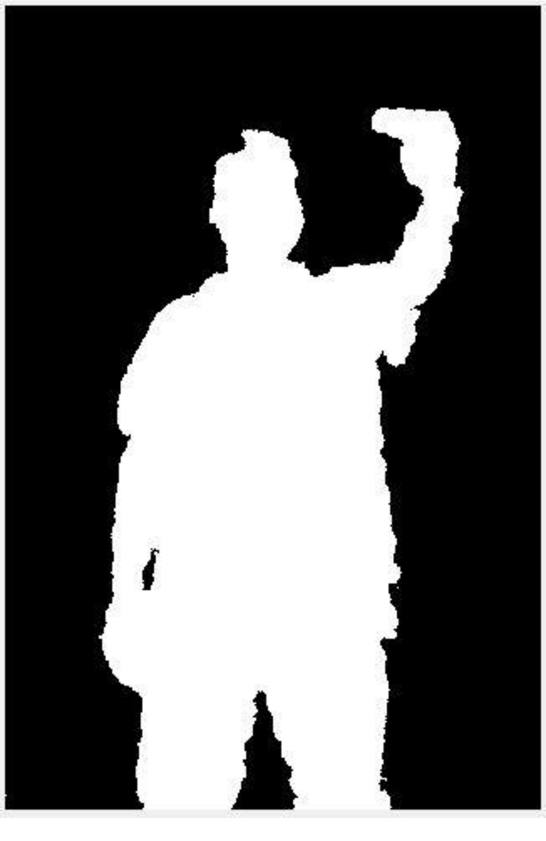


Figure 3.2 Segmented Depth Frame of frame in Figure 3.1

3.2 Feature Extraction

Feature extraction is a type of dimensionality reduction that efficiently represents interesting parts of an image as a compact feature vector. It is the most crucial step of the image processing technique. In our proposed system we have extracted two different features one is the gradient auto correlation features from the depth motion maps and the other one is the histogram of oriented 4D normal(HON4d). And finally we have done the weighted fusion of the features at the decision level.

3.2.1 Extraction of GLAC (Gradient local auto correlation features) from depth motion maps

For each input gesture sequence we have created three depth motion maps. Hence a 3D gesture is represented into three 2D depth motion maps. After the creation of depth motion maps gradient local auto correlation features are extracted from the three depth motion maps and concatenated into a single feature vector which is then used for classification.

3.2.1.1 Creation of Depth Motion Maps

In order to use the body shape and motion information from the input depth sequences, each frame of a depth sequence of a gesture is projected onto three orthogonal Cartesian planes. Three projected views of a frame gives us three kind of view of each frame which is Front, Side and Top. For each projection view motion energy is accumulated by adding the absolute difference of the consecutive frames. Hence, each 3D gesture depth sequence is reduced to three 2D depth motion maps (DMM_F, $DMM_{S,}$ $DMM_{T)}$. Say a given gesture depth sequence has N frames ,then the three depth motion maps from the three projection views front, side and top is calculated as follows:

$$DMM_{\{F,S,T\}} = \sum_{j=1}^{N-1} |map_{\{F,S,T\}}^{j+1} - map_{\{F,S,T\}}^{j}| \qquad (3.1)$$

Figure 3.3 represents three depth motion maps (DMM_F, $DMM_{S,}$ DMM_T) generated from a depth sequence of an action high throw.

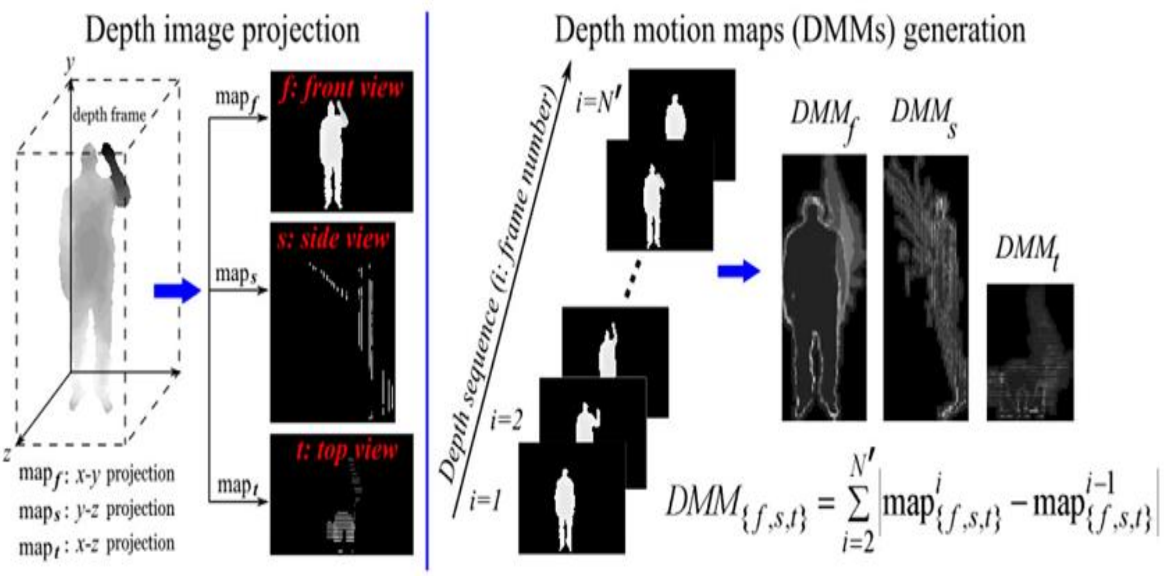

Depth image projection
Depth motion maps (DMMs) generation
y
depth frame
z
map_f
f: front view
map_s
s: side view
map_t
t: top view
map_f: x-y projection
map_s: y-z projection
map_t: x-z projection
Depth sequence (i: frame number)
i=N'
i=2
i=1
DMM_f
DMM_s
DMM_t
$DMM_{\{f,s,t\}} = \sum_{i=2}^{N'} \left| \text{map}^{i}_{\{f,s,t\}} - \text{map}^{i-1}_{\{f,s,t\}} \right|$

Figure 3.3 Three depth motion maps generated from depth sequence of an action high throw [6]

3.2.1.2 Gradient local auto correlation (GLAC) feature [7]

"GLAC: Gradient local auto correlation" extracts 2nd order statistics of gradients and thus gives more detailed and discriminative information compared to conventional histogram based methods like HOG(Histogram of gradients) and SIFT(Scale invariance feature transform). Gradient local auto correlation features are mainly based on the spatial and orientation correlation of local image gradients, their gradients are described in terms of their orientation and magnitude. HOG (Histogram of Gradients) and SIFT (Scale Invariance Feature Transform) are based on the first order statistics which is gradients while GLAC features are based on the second order statistics i.e., auto correlations. Thus, GLAC is an extension of features like HOG and SIFT.

Let I be an image region of the image we have to find the feature and let 'r' be a position vector in I defined as r = (x, y)^t . The magnitude of the image gradient at each pixel of the region is given by

$n = \sqrt{\frac{\partial I^2}{\partial x} + \frac{\partial I^2}{\partial y}}$ and its orientation angle is given by $\theta = \arctan\left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}\right)$. The orientation angle θ is then converted into D orientation bins by giving weights to their nearest bins to form a gradient orientation vector $f \epsilon\ R^D$. Using the magnitude of the image gradient 'n' and the gradient orientation vector 'f' , the Nth auto correlation of the local gradients can be calculated as follows:

$$R(d_0,..,d_n,a_1,...,a_n)=\int \omega[n(r), n(r+a_n), \dots, n(r+a_n)] f_{d0}(r) f_{d1}(r+a_1) \dots f_{dn}(r+a_n) dr \quad (3.2)$$

Here, w(.) is a weighing function , a_i are the displacement vectors from the reference points r, f_d id the d-th element of vector f. In the experiments, $N \epsilon \{0,1\}$, $a_{1x,y} \in \{\pm\Delta r, 0\}$ and $w(.) \equiv \min(.)$ were used as being suggested in [7], Δr represents the interval in both vertical and horizontal directions . For $N \epsilon \{0,1\}$, the calculation of GLAC is given by :

$$\mathbf{F}_0 : R_{N=0}(d_0) = \sum_{\mathbf{r}\in I} n(\mathbf{r}) f_{d_0}(\mathbf{r})$$

$$\mathbf{F}_1 : R_{N=1}(d_0, d_1, \mathbf{a}_1) = \sum_{\mathbf{r}\in I} \min[n(\mathbf{r}), n(\mathbf{r}+\mathbf{a}_1)] f_{d_0}(\mathbf{r}) f_{d_1}(\mathbf{r}+\mathbf{a}_1). \tag{3.3}$$

Figure 3.4 represents the spatial auto correlation patterns of (r, r+a1).

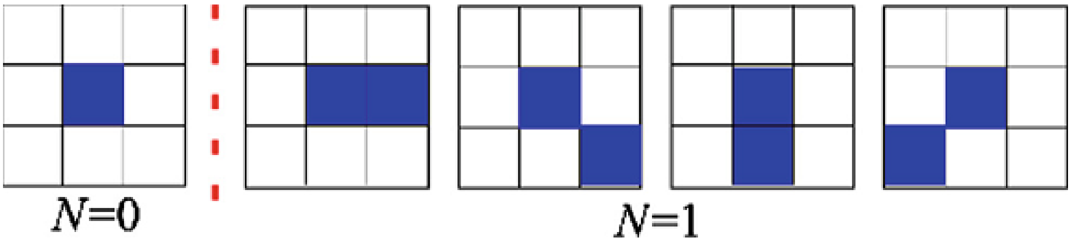


Figure 3.4 Configuration patterns of (r, r+a1)

The dimensionality of the above gradient auto correlation features F0 and F1 is given by D+4D^2. Even though the dimensions of the GLAC features are high, the computational cost is low because of the sparseness of ‘f’. The computational cost is invariant to the number of bins D, because the sparseness of ‘f’ do not depend on D.

The three depth motion maps generated in the previous step for each gesture represents only the pixel level features. To have more discriminative and compact information we extract the GLAC features defined here from the depth motion maps by dividing the depth motion map into several non-overlapping blocks. GLAC features for each block are extracted and then concatenated into a single vector. The GLAC features are extracted for all the three depth motion maps (DMM$_F$, DMM$_S$, DMM$_T$), and finally all the three GLAC feature vectors ($GLAC_F$, $GLAC_S$, $GLAC_T$) corresponding to the three depth motion maps are then concatenated to form a single feature vector. Figure 3.5 represents the flowchart of the process of extracting GLAC features from the depth motion maps.

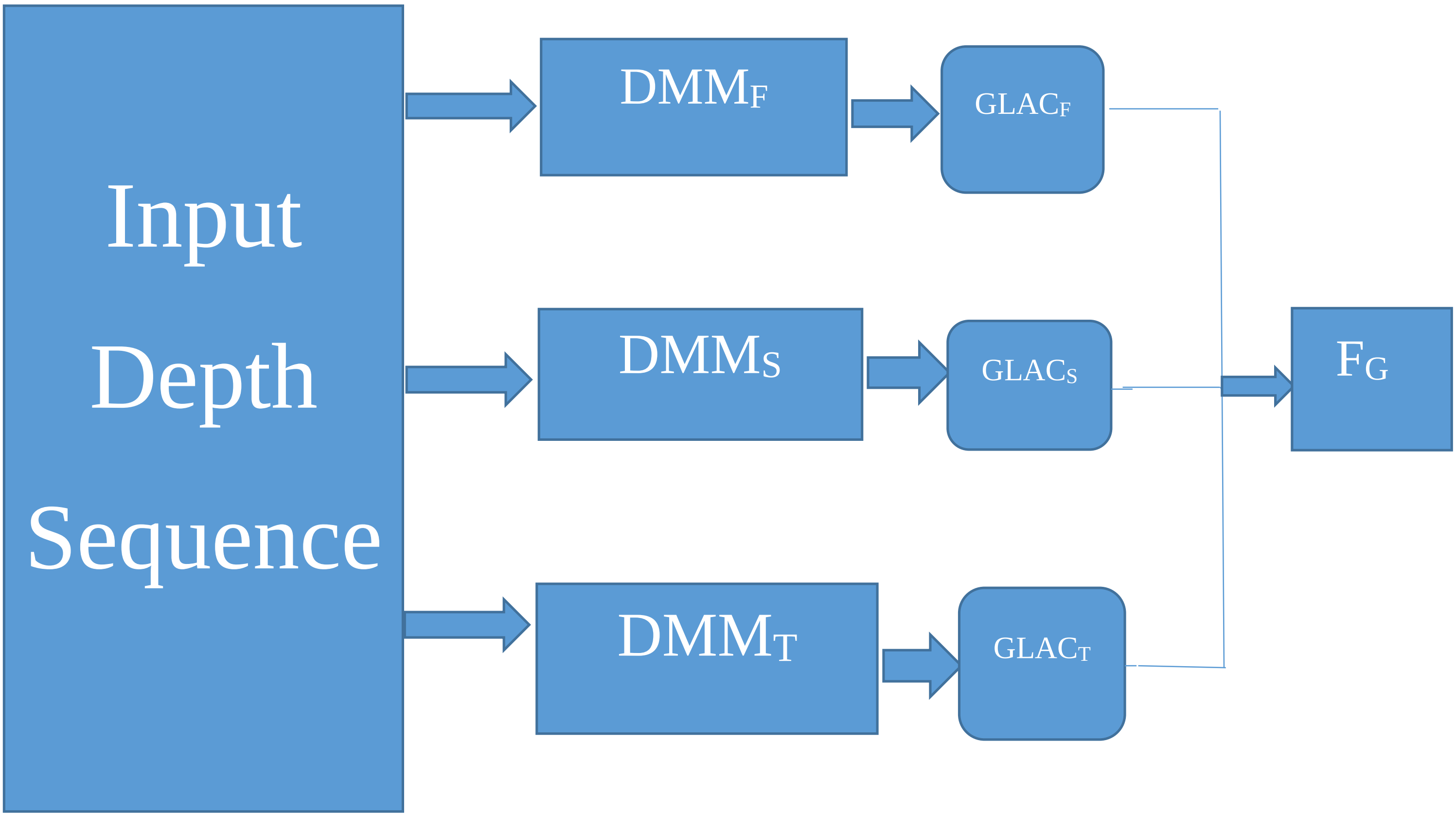


Figure 3.5 Flowchart of extracting GLAC features from depth sequence

3.2.2 Extraction of HON4D (Histogram of oriented 4D normal) features [8]

Using HON4D, depth sequence of a gesture is described using the surface normal orientation in the 4D space of time, depth and spatial coordinates. 4D projectors are created which quantize the 4D space and thus represents the possible directions of the 4D normal. The projectors are initialized using the vertices of a regular polychoron. The projectors are then refined using a discriminative density measure.

The shape cues at a particular time instance are captured by the surface normals while the motion cues is captured by the change in the surface normals over time. Figure 3.6 represents the overview of the steps involved in calculating HON4D descriptor.

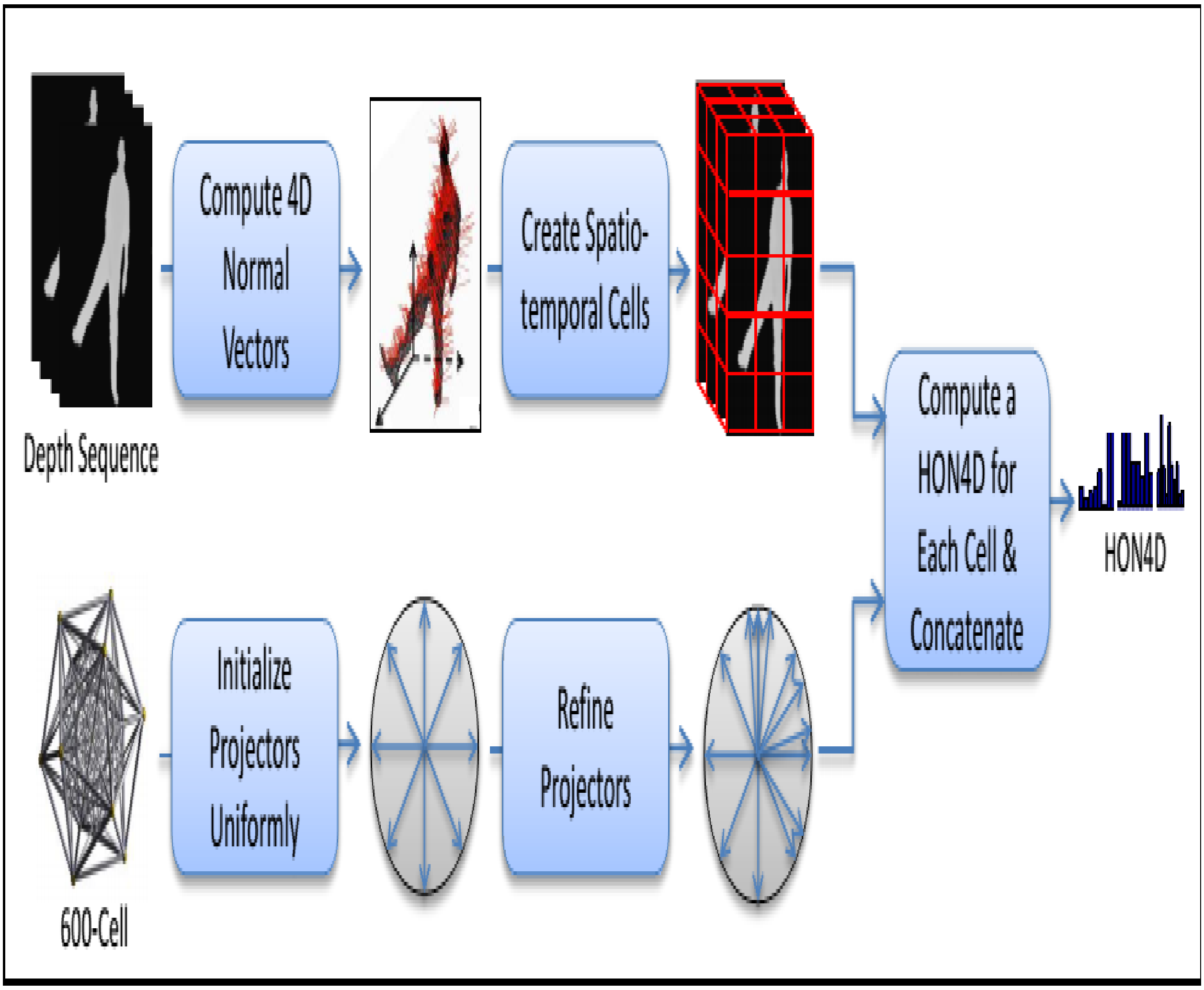

Depth Sequence
Compute 4D Normal Vectors
Create Spatio-temporal Cells
Compute a HON4D for Each Cell & Concatenate
HON4D
600-Cell
Initialize Projectors Uniformly
Refine Projectors

Figure 3.6 Steps involved in calculation of HON4D descriptor [8]

3.3 Classification using 2 layer Feed Forward Neural Network

It is an information processing system based on the idea of the working of neurons in the brain. It uses neurons (nodes) as its fundamental functional units. The neurons are connected through links and they have some associated weight. Each node receives input and the input function computes the weighted sum of the input values. Activation function transforms this sum into output value. Some of the commonly used activation functions are step, sigmoid, softmax etc. Now, multi layered network is used where hidden layer nodes do not have any communication with the outside world, it requires complex training but provides better performance in terms of classification accuracy. The network learns by updating the weight. A neural network is mainly characterized by three things 1. The way the neurons are connected with each other (the architecture of the network) 2. The algorithm used for the training, training algorithms mainly focuses on updating the weights for optimal classification 3. The activation function used.

In our proposed system we have used a 2 layer feedforward neural network. The network has one input layer, one hidden layer and one output layer. The activation function used in the hidden layer is sigmoid. The neurons in the output layer are softmax and the network is trained using scaled conjugate gradient backpropogation algorithm.

Trainscg function in matlab is used for training the neural network using scaled conjugate gradient backpropogation algorithm. Trainscg function can train a network as long as its weight, net input and transfer functions have derivative functions. Backpropogation is used to calculate derivatives of performance with respect to the weight and bias variable x. Figure 3.7 represents the architecture of the neural network used in our method. There is one input layer with 21168 neurons which is the feature vector length of a gesture, there is one hidden layer with 30 neurons and sigmoid activation function .Since there are 12 classes in our dataset, output layer has 12 softmax neurons. These output neurons gives the probability outputs in the range 0 to 1, and all the outputs add to 1.

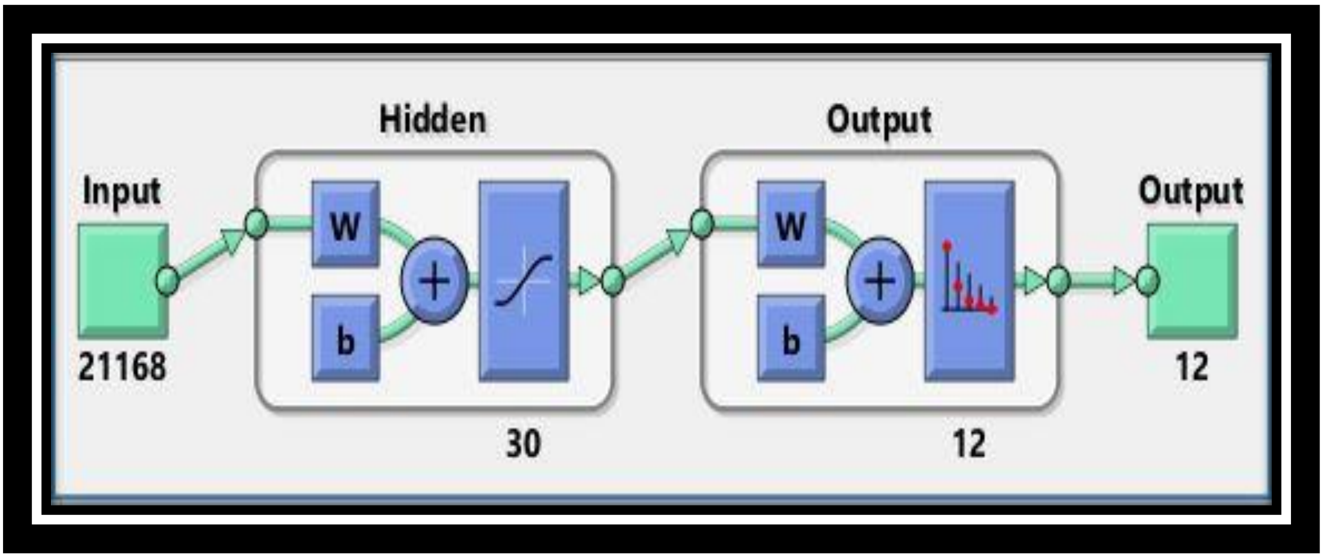


Figure 3.7 Architecture of the neural network

3.4 Weighted Fusion of Classifiers Outcomes

We have generated two different features from our given depth sequences, one is gradient local auto correlation features (GLAC) from depth motion maps (F_G) and the other one is (HON4D) histogram of 4D normal features (F_H). In our DMM based GLAC feature extraction technique (F_G) we have taken the average difference between the depth frames. This feature loses all the temporal variations information of a depth sequence and thus can give poor results when temporal variation information is of significance. To overcome this shortcoming we have extracted another feature HON4D which describes a depth sequence using the surface normal orientation on the 4D space of time, depth and spatial co-ordinates. Thus, the temporal information which may get lost using DMM based GLAC features will be retained using HON4D feature descriptors.

Finally, to get the benefits of both the features DMM based GLAC as well as HON4D, we have fused both the features at the decision level using an ensemble of three neural network classifiers. F_G is the features extracted from DMM based GLAC and F_H is the features extracted from HON4D descriptors. First, both these features are fed individually to the Neural Network and then their probability outcomes are fused to get the final result.

Say, P_G is the probability outcome of the F_G features and P_H is the probability outcome of the F_H features. We will fuse the probability outputs P_G and P_H, and the final output P_O is given by the equation as follows:

$$P_O = w1 * P_G + w2 * P_H \tag{3.4}$$

W1 and w2 are the weight vectors by which the probability outcomes are multiplied .Instead of finding these weights empirically we determine these weights by feeding the outputs P_G and P_H to the third Neural Network to get the final outcome. The weighs are optimized by the neural network backpropogation algorithm to give us the final optimized result. Figure 3.8 represents the framework of the proposed fusion method, first from the input depth sequence DMM based GLAC features and HON4D features are extracted, then these features F_G and F_H are fed to the Neural Networks separately, next the probability outcomes of both the classifiers P_G and P_H are given to the third neural network to give the final outcome P_O. Third Neural Network automatically optimizes the weights and gives us the optimized output P_O.

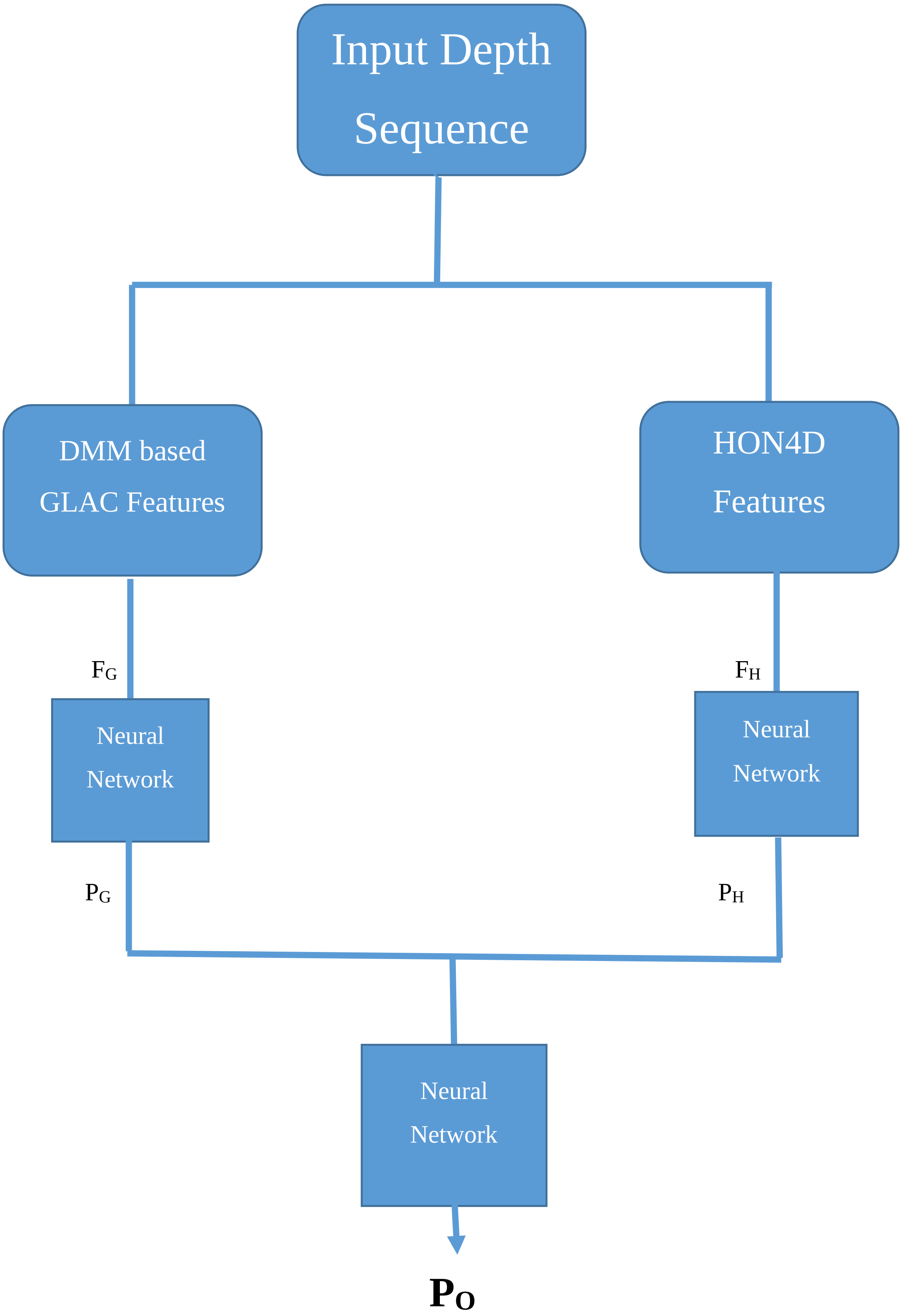


Figure 3.8 Framework of proposed fusion method

4. EXPERIMENTS AND RESULTS

We have tested our proposed method on the following two dataset:

1. **Dataset 1:** Dataset 1 is **MSRGesture3D** dataset taken from [9]. This dataset contains 12 American Sign Language dynamic gestures captured from Kinect device.
2. **Dataset 2:** Dataset 2 is **ISL3D** dataset. This dataset is created by us using a Kinect device. The dataset contains 12 dynamic gestures of Indian Sign Language.

4.1 Dataset 1: MSRGesture3D dataset

MSRGesture3D dataset is a benchmark dataset by Microsoft research. The dataset contains 12 dynamic gestures of the American Sign Language. The 12 dynamic gestures of the dataset are: z, j, where, store, pig, past, hungry, green, finish, blue, bathroom, milk. The 12 dynamic gestures are performed by 10 subjects in the dataset and each gesture is performed 2 or 3 times. The dataset is captured using a Kinect device. The dataset contains 333 files, each corresponding to a depth sequence. The hand portion above the wrist is already being segmented in the dataset. This dataset is considered challenging because of the self-occlusion issues.

Since, in the dataset the hand portion is already being segmented we did not apply the first step which is segmentation or background subtraction to the dataset. For this dataset we have directly moved to the feature extraction step. Figure 4.1 represents the 12 dynamic gestures of American Sign Language from the MSRGesture3D dataset.

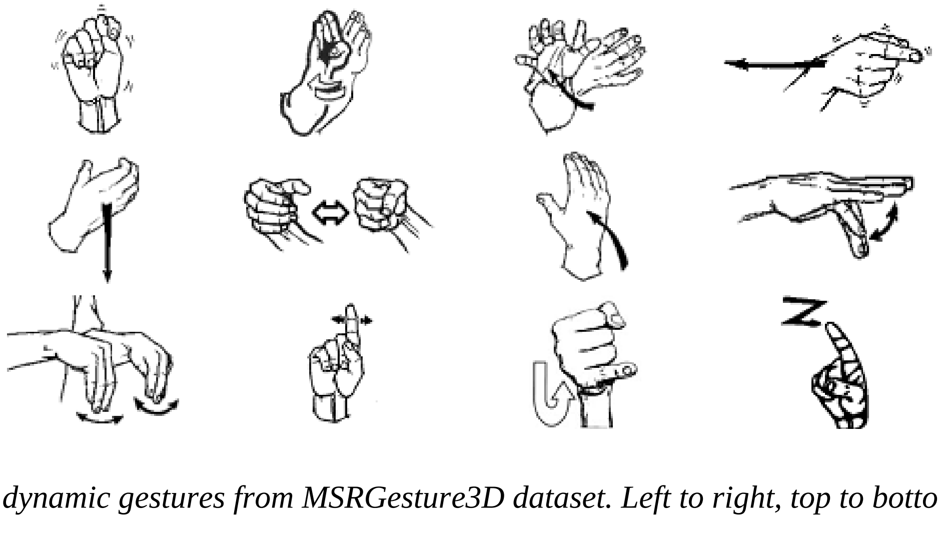
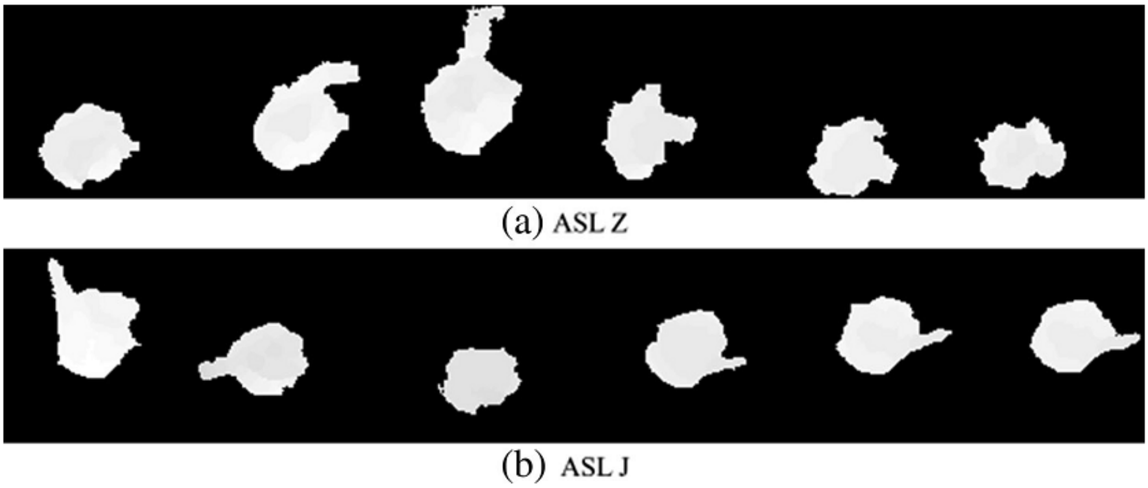


Figure 4.1 12 dynamic gestures from MSRGesture3D dataset. Left to right, top to bottom: bathroom, blue, finish, green, hungry, milk, past, pig, store, where, letter J, letter Z

Figure 4.2 represents some example depth sequences from the MSRGesture3D dataset.



(a) ASL Z

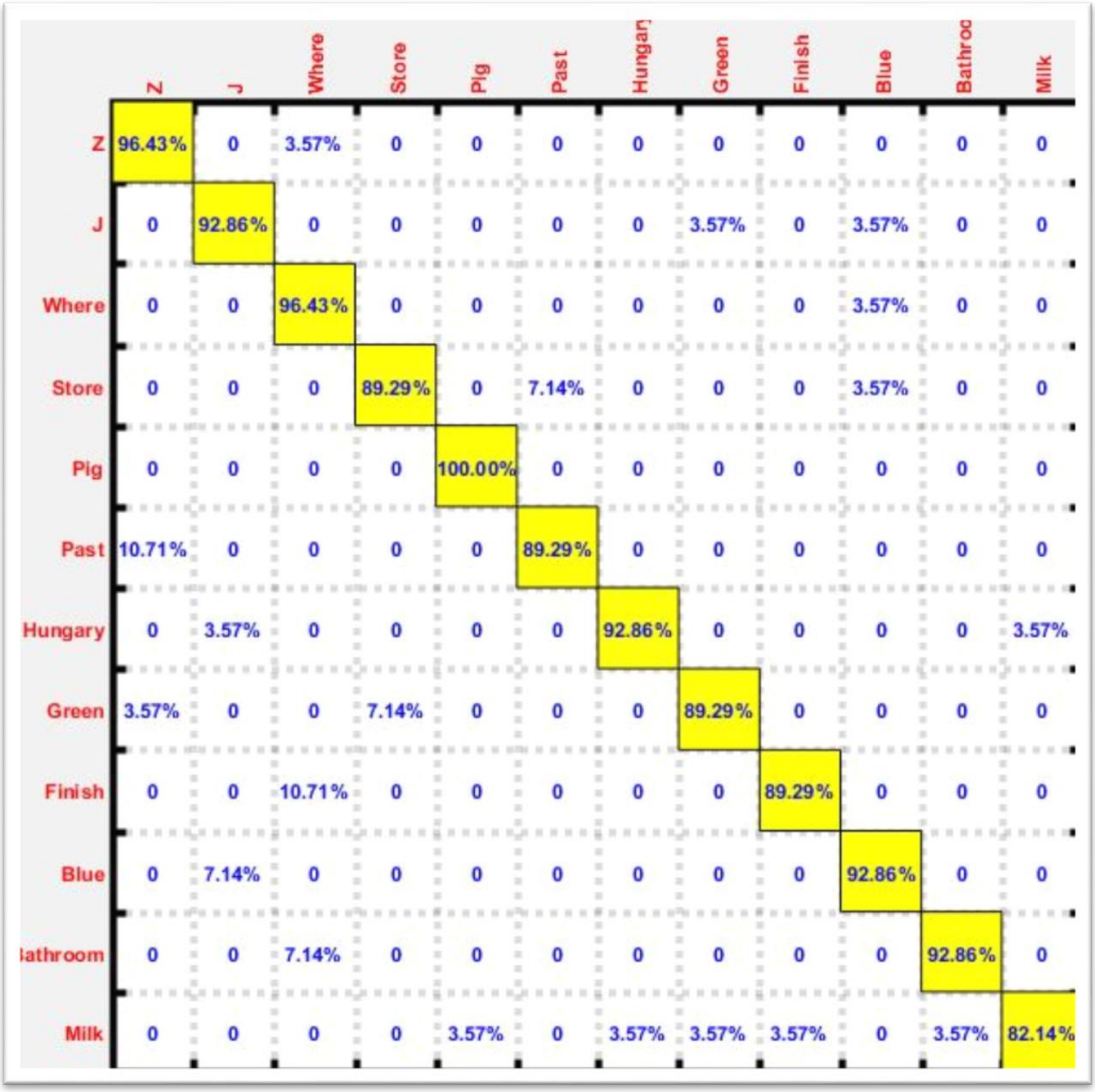
(b) ASL J

Figure 4.2 Example depth sequence from MSRGesture3D dataset (a) ASL Z (b) ASL J

4.1.1 Results using DMM-GLAC features and 2-Layer Feed Forward Neural Network

GLAC features from the depth motion maps created from the depth sequences were extracted. The length of the feature vector for each sample was 21168 x 1. Since there are 333 total sequences, the dimension of the feature matrix F_G was 21168 x 333. The classifier used is 2-Layer Feed Forward Neural Network. The number of neurons used in the hidden layer were 30. In the experimental setup the leave one subject out cross validation test is performed as in [4].

The average recognition accuracy achieved with DMM-GLAC features and 2-Layer Feed Forward Neural Network classifier is 91.89%. Figure 4.3 represents the confusion matrix.



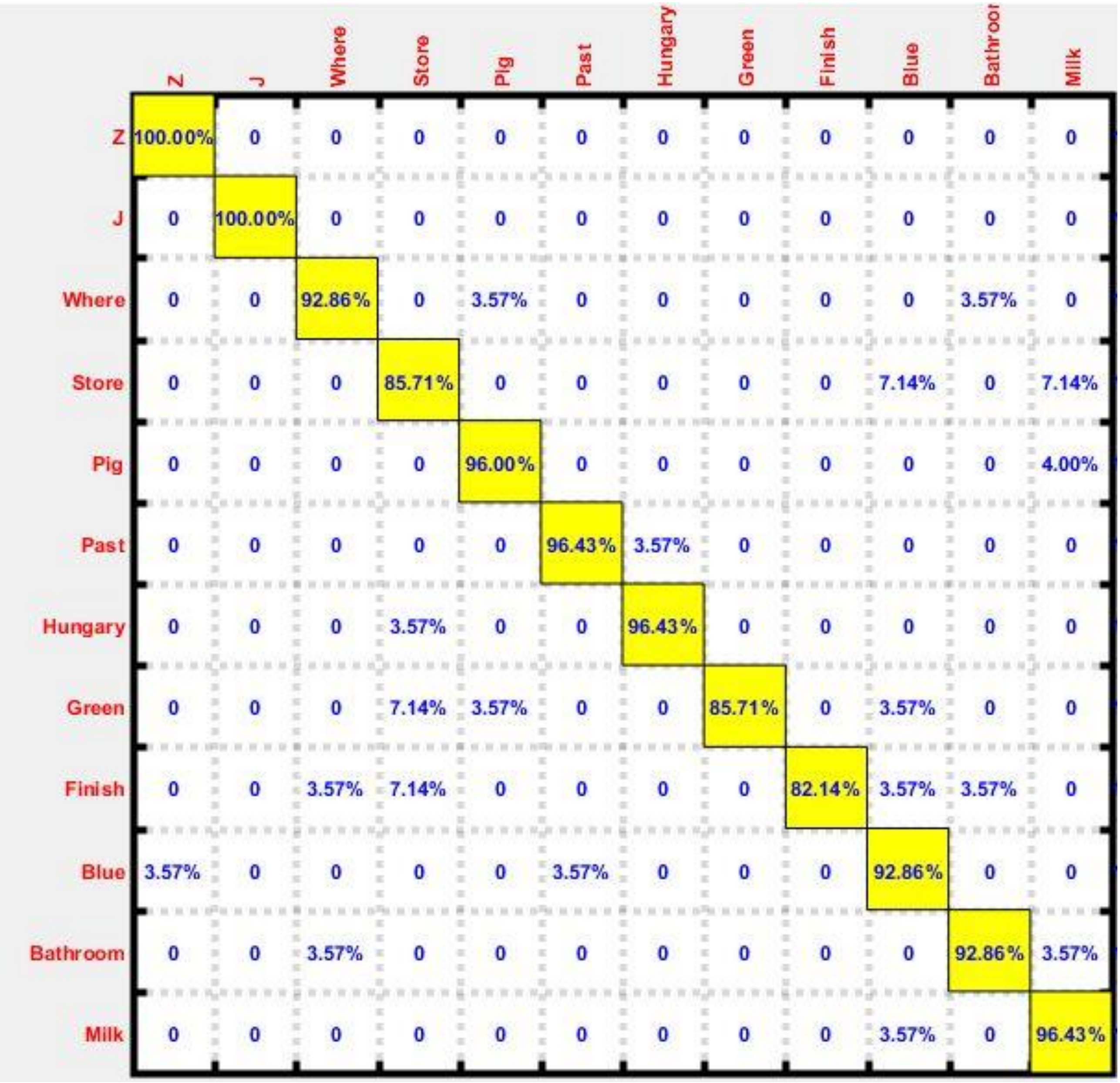
	Z	J	Where	Store	Pig	Past	Hungar	Green	Finish	Blue	Bathroo	Milk
Z	96.43%	0	3.57%	0	0	0	0	0	0	0	0	0
J	0	92.86%	0	0	0	0	0	3.57%	0	3.57%	0	0
Where	0	0	96.43%	0	0	0	0	0	0	3.57%	0	0
Store	0	0	0	89.29%	0	7.14%	0	0	0	3.57%	0	0
Pig	0	0	0	0	100.00%	0	0	0	0	0	0	0
Past	10.71%	0	0	0	0	89.29%	0	0	0	0	0	0
Hungary	0	3.57%	0	0	0	0	92.86%	0	0	0	0	3.57%
Green	3.57%	0	0	7.14%	0	0	0	89.29%	0	0	0	0
Finish	0	0	10.71%	0	0	0	0	0	89.29%	0	0	0
Blue	0	7.14%	0	0	0	0	0	0	0	92.86%	0	0
athroom	0	0	7.14%	0	0	0	0	0	0	0	92.86%	0
Milk	0	0	0	0	3.57%	0	3.57%	3.57%	3.57%	0	3.57%	82.14%

Figure 4.3 Confusion matrix, DMM-GLAC and 2-Layer Feed Forward Neural Network

Accuracy = 91.89 %

4.1.2 Results using HON4D features and 2-Layer Feed Forward Neural Network

The HON4D descriptors directly from the depth sequences were extracted. The length of the feature for each sequence was 34176 x 1. Since there were 333 depth sequences. The dimension of the feature matrix F_H was 34176 x 333. The leave one subject out cross validation test is performed as in [4].The average recognition accuracy achieved with HON4D features and 2-Layer Feed Forward Neural Network classifier is 93.09%. Figure 4.4 represents the confusion matrix.



	Z	J	Where	Store	Pig	Past	Hungary	Green	Finish	Blue	Bathroo	Milk
Z	100.00%	0	0	0	0	0	0	0	0	0	0	0
J	0	100.00%	0	0	0	0	0	0	0	0	0	0
Where	0	0	92.86%	0	3.57%	0	0	0	0	0	3.57%	0
Store	0	0	0	85.71%	0	0	0	0	0	7.14%	0	7.14%
Pig	0	0	0	0	96.00%	0	0	0	0	0	0	4.00%
Past	0	0	0	0	0	96.43%	3.57%	0	0	0	0	0
Hungary	0	0	0	3.57%	0	0	96.43%	0	0	0	0	0
Green	0	0	0	7.14%	3.57%	0	0	85.71%	0	3.57%	0	0
Finish	0	0	3.57%	7.14%	0	0	0	0	82.14%	3.57%	3.57%	0
Blue	3.57%	0	0	0	0	3.57%	0	0	0	92.86%	0	0
Bathroom	0	0	3.57%	0	0	0	0	0	0	0	92.86%	3.57%
Milk	0	0	0	0	0	0	0	0	0	3.57%	0	96.43%

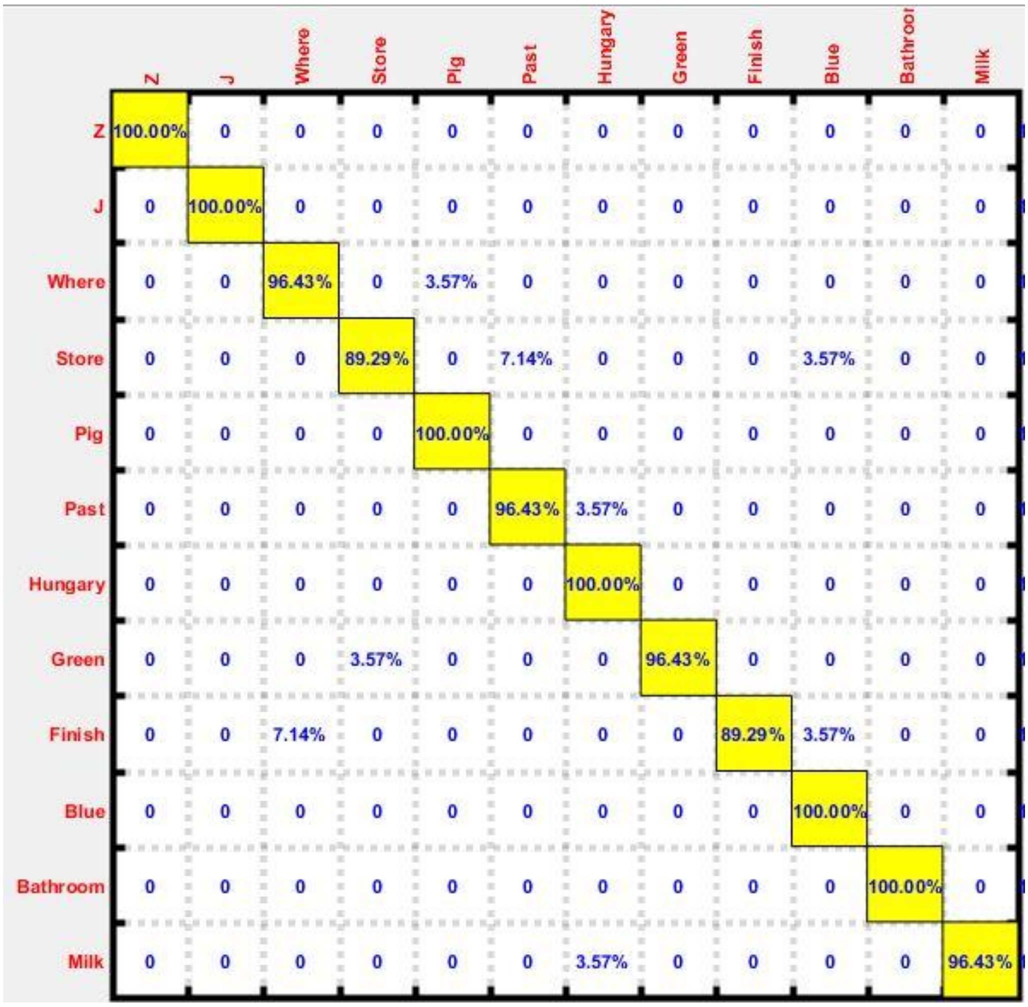
Figure 4.4 Confusion matrix, HON4D and 2-Layer Feed Forward Neural Network

Accuracy = 93.09 %

4.1.3 Results by weighted fusion of classifier outcomes using a Neural Network

We have fused the probability outcomes P_G and P_H from DMM-GLAC features and HON4D features respectively using the method as discussed in Section 3.4. For the experiment we have further divided the training data into training and validation set. The outputs on the validation set from the first two neural networks were combined to form the training data for the third neural network, and the outputs on the test set from the first two neural networks were combined to form the test data for the third neural network. The third neural network gave us the final output. 20 neurons are used in the hidden layer of the neural network.

The average recognition accuracy achieved by fusing the probability outputs using a neural network is 96.99%. Figure 4.5 represents the confusion matrix.



	Z	J	Where	Store	Pig	Past	Hungary	Green	Finish	Blue	Bathroo	Milk
Z	100.00%	0	0	0	0	0	0	0	0	0	0	0
J	0	100.00%	0	0	0	0	0	0	0	0	0	0
Where	0	0	96.43%	0	3.57%	0	0	0	0	0	0	0
Store	0	0	0	89.29%	0	7.14%	0	0	0	3.57%	0	0
Pig	0	0	0	0	100.00%	0	0	0	0	0	0	0
Past	0	0	0	0	0	96.43%	3.57%	0	0	0	0	0
Hungary	0	0	0	0	0	0	100.00%	0	0	0	0	0
Green	0	0	0	3.57%	0	0	0	96.43%	0	0	0	0
Finish	0	0	7.14%	0	0	0	0	0	89.29%	3.57%	0	0
Blue	0	0	0	0	0	0	0	0	0	100.00%	0	0
Bathroom	0	0	0	0	0	0	0	0	0	0	100.00%	0
Milk	0	0	0	0	0	0	3.57%	0	0	0	0	96.43%

Figure 4.5Confusion matrix, weighted fusion of outcomes using neural network

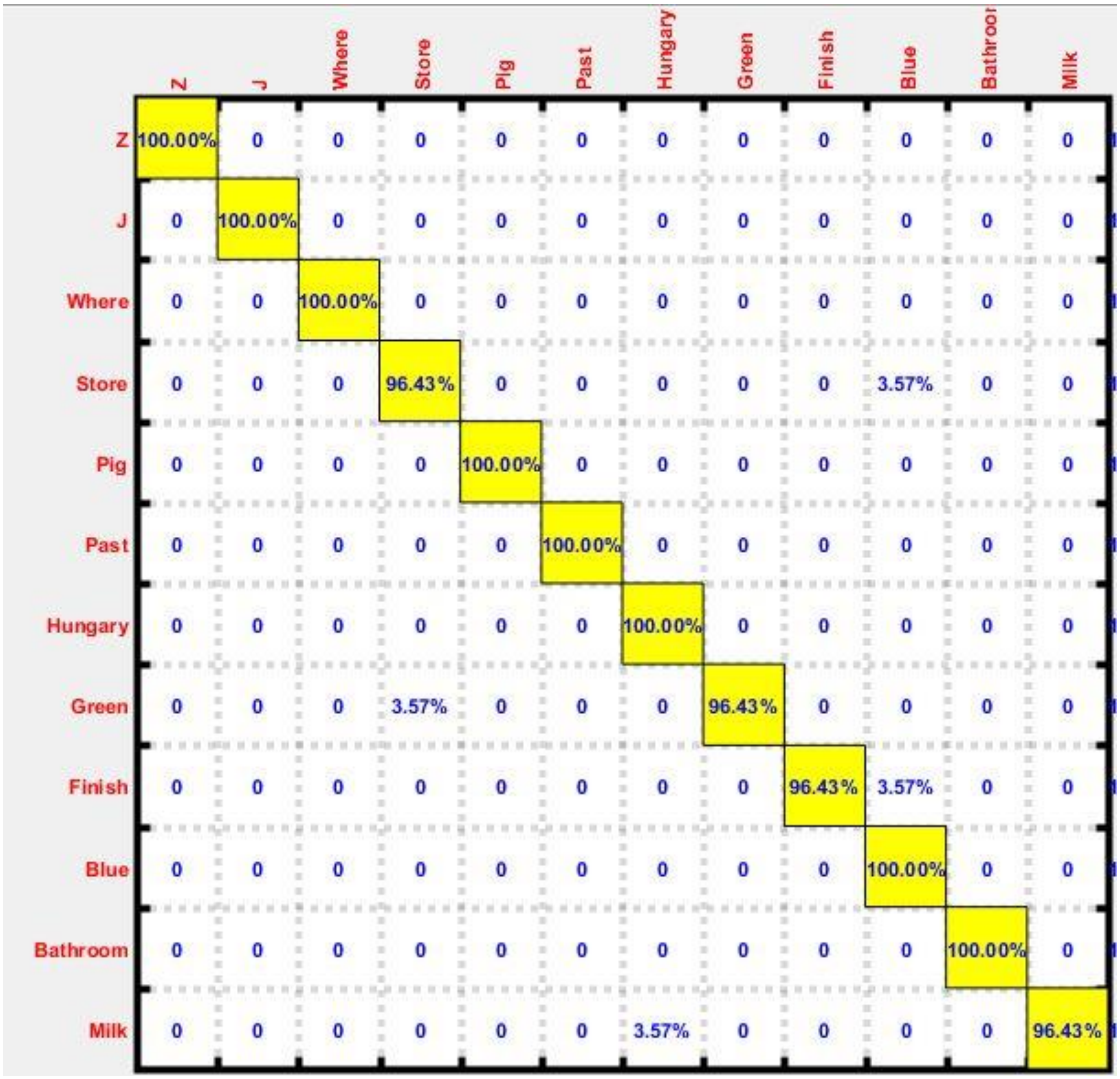
Accuracy = 96.99%

4.1.4 Results by weighted fusion of classifier outcomes using weighing factor alpha

We have fused the probability outcomes P_G and P_H from DMM-GLAC features and HON4D features respectively using the equation:

$$P_O = \alpha * P_G + (1-\alpha) * P_H \tag{4.1}$$

The value of the weighing factor α is defined empirically. The average recognition accuracy achieved by this method is 98.8 %.Figure 4.6 represents the confusion matrix.



	Z	J	Where	Store	Pig	Past	Hungary	Green	Finish	Blue	Bathroor	Milk
Z	100.00%	0	0	0	0	0	0	0	0	0	0	0
J	0	100.00%	0	0	0	0	0	0	0	0	0	0
Where	0	0	100.00%	0	0	0	0	0	0	0	0	0
Store	0	0	0	96.43%	0	0	0	0	0	3.57%	0	0
Pig	0	0	0	0	100.00%	0	0	0	0	0	0	0
Past	0	0	0	0	0	100.00%	0	0	0	0	0	0
Hungary	0	0	0	0	0	0	100.00%	0	0	0	0	0
Green	0	0	0	3.57%	0	0	0	96.43%	0	0	0	0
Finish	0	0	0	0	0	0	0	0	96.43%	3.57%	0	0
Blue	0	0	0	0	0	0	0	0	0	100.00%	0	0
Bathroom	0	0	0	0	0	0	0	0	0	0	100.00%	0
Milk	0	0	0	0	0	0	3.57%	0	0	0	0	96.43%

Figure 4.6 Confusion matrix, weighted fusion using weighing factor alpha

Accuracy = 98.8%

4.1.5 Comparison of results with other state of the art approaches

We have compared our results with the other state of the art approaches. Table 4.1 gives a comparison chart of the techniques used their accuracies on MSRGesture3D Dataset.

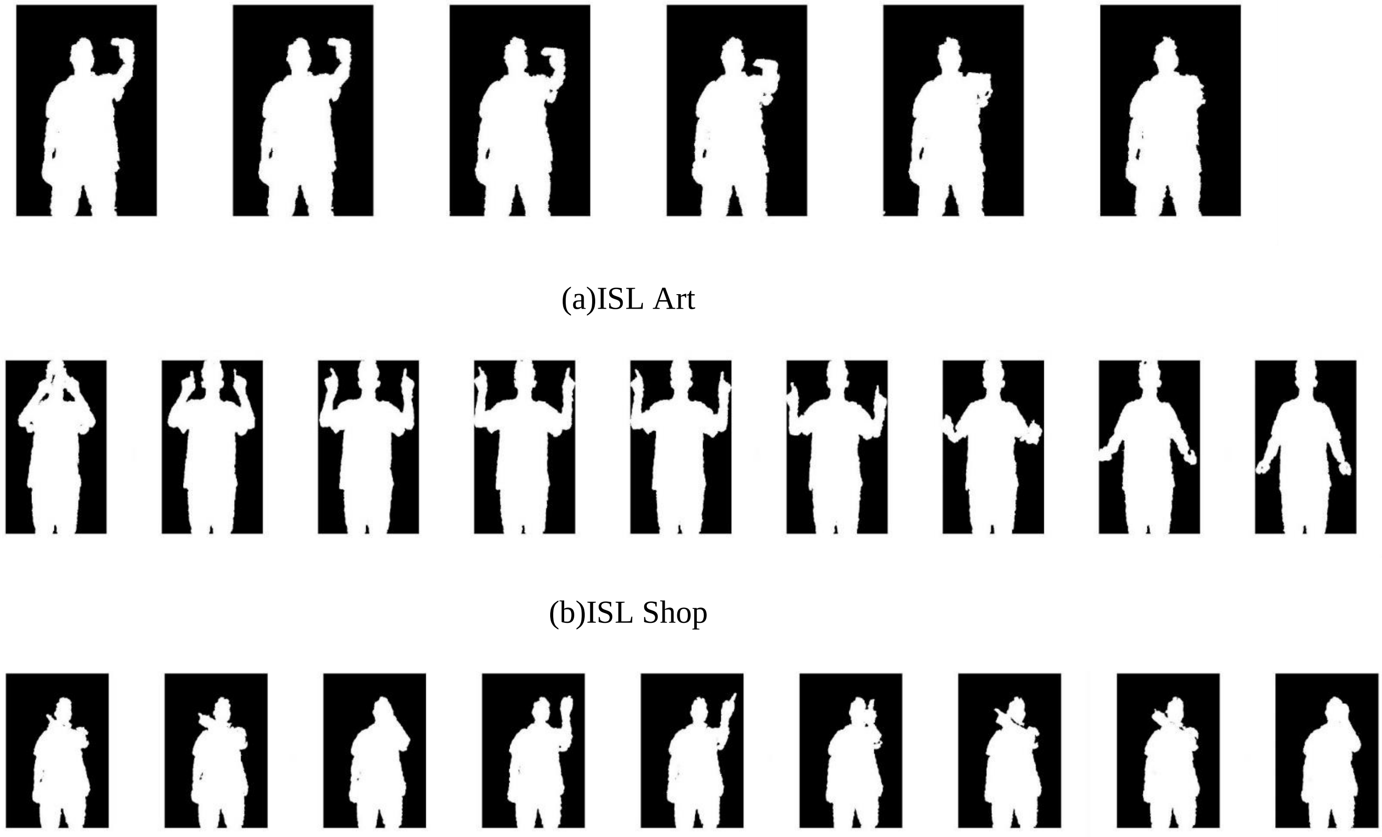
Table 4.1 Comparison of Recognition Accuracies on MSRGesture3D dataset

Method	**Accuracy**
SVM on Raw Features	67%
Action Graph on Silhouettes [4]	87.70%
Random Occupancy Patterns [10]	88.50%
DMM-HOG-SVM [11]	89.20%
HON4D [8]	92.45%
Histogram of Depth Gradients [14]	92.76%
DMM-LBP-KELM [12]	94.60%
SNV [15]	94.74%
GLAC-ELM [13]	95.50%
Motion History-Binary shape templates [16]	96.60%
2D-3D features [6]	98.5 %
Proposed Method (Fusion using neural network)	**96.99%**
Proposed Method (Fusion using weighing factor alpha)	**98.8 %**

4.2 Dataset 2: ISL3D Dataset

ISL3D is an Indian Sign Language dynamic gestures dataset created by us. The dataset contains 12 dynamic gestures of the Indian Sign Language. The 12 dynamic gestures of the dataset are: **art, eat, swallow, tell, circle, warning, red, blue, cold, work, shop, and fan**. The 12 dynamic gestures are performed by 10 subjects in the dataset and each gesture is performed 3 times. The dataset is captured using a Kinect device. The dataset contains 360 files, each corresponding to a depth sequence. This dataset is considered challenging because of the variations in the same gesture performed by different subjects. Segmentation step was performed on the dataset before feature extraction.

Figure 4.7 represents some example depth sequence from ISL3D dataset.



(a)ISL Art

(b)ISL Shop

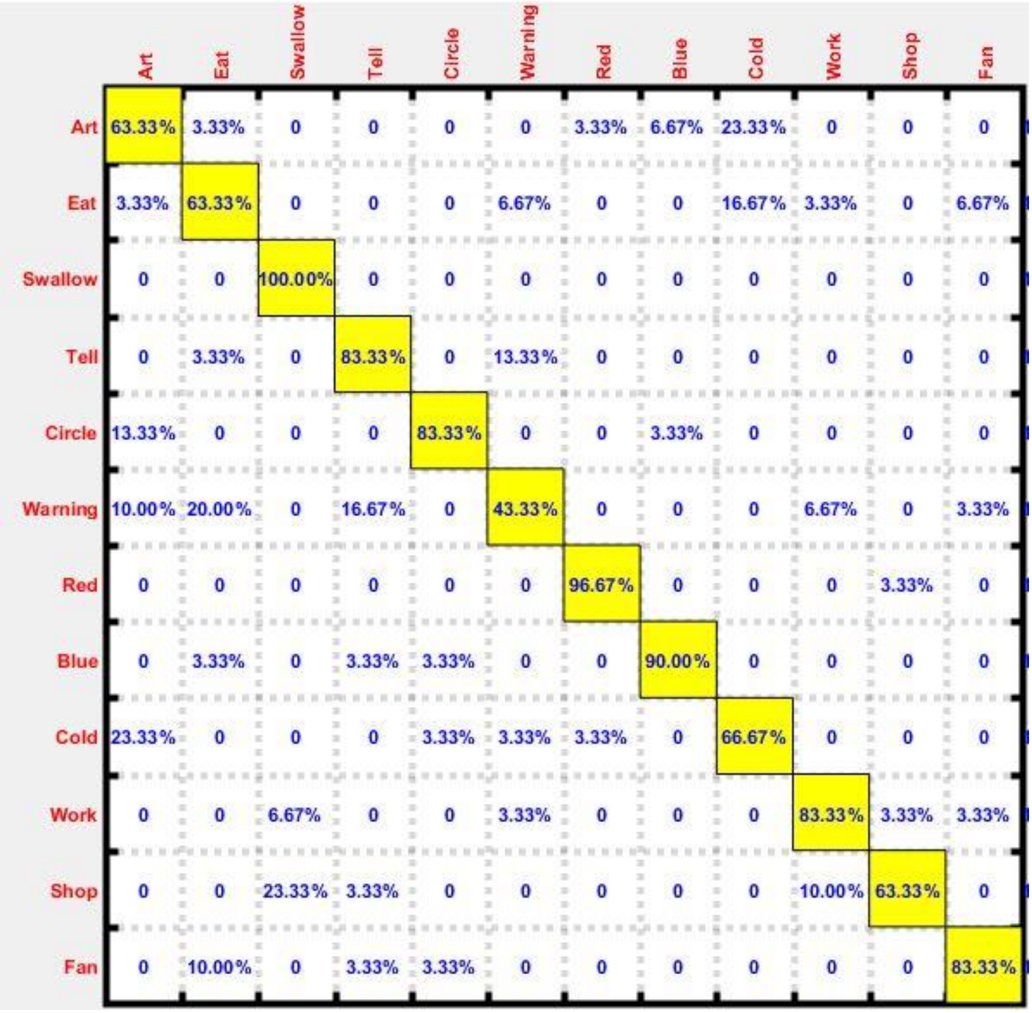
(c)ISL Fan

Figure 4.7 Example depth sequences from ISL dataset (a) Art (b) Shop (c) Fan

4.2.1 Results using DMM-GLAC features and 2-Layer Feed Forward Neural Network

GLAC features from the depth motion maps created from the depth sequences were extracted. The length of the feature vector for each sample was 11988 x 1. Since there are 360 total sequences, the dimension of the feature matrix F_G was 11988 x 360. The classifier used is 2-Layer Feed Forward Neural Network. The number of neurons used in the hidden layer were 30. In the experimental setup the leave one subject out cross validation test is performed as in [4].

The average recognition accuracy achieved with DMM-GLAC features and 2-Layer Feed Forward Neural Network classifier is 76.67%. Figure 4.8 represents the confusion matrix.



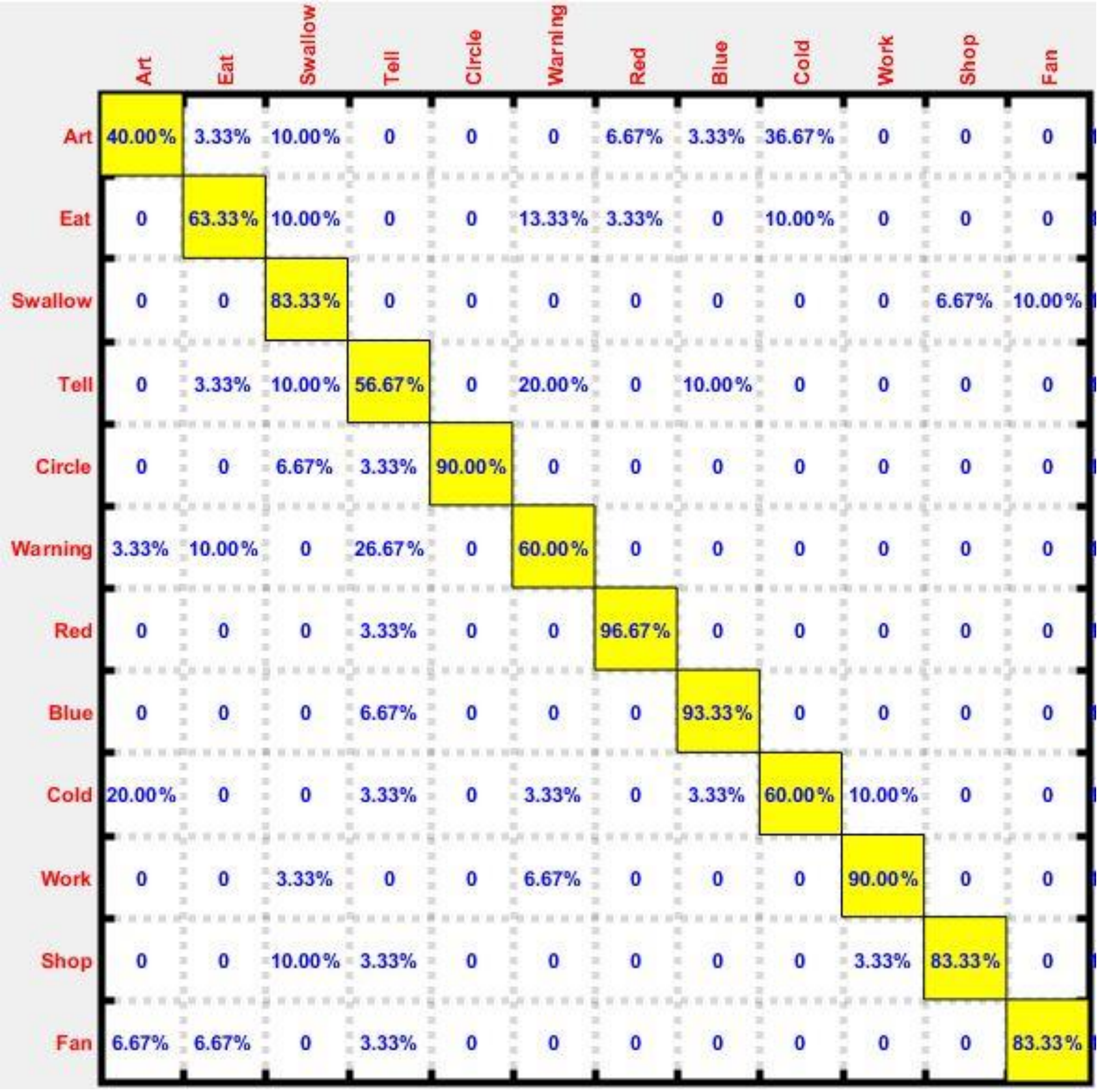
	Art	Eat	Swallow	Tell	Circle	Warning	Red	Blue	Cold	Work	Shop	Fan
Art	63.33%	3.33%	0	0	0	0	3.33%	6.67%	23.33%	0	0	0
Eat	3.33%	63.33%	0	0	0	6.67%	0	0	16.67%	3.33%	0	6.67%
Swallow	0	0	100.00%	0	0	0	0	0	0	0	0	0
Tell	0	3.33%	0	83.33%	0	13.33%	0	0	0	0	0	0
Circle	13.33%	0	0	0	83.33%	0	0	3.33%	0	0	0	0
Warning	10.00%	20.00%	0	16.67%	0	43.33%	0	0	0	6.67%	0	3.33%
Red	0	0	0	0	0	0	96.67%	0	0	0	3.33%	0
Blue	0	3.33%	0	3.33%	3.33%	0	0	90.00%	0	0	0	0
Cold	23.33%	0	0	0	3.33%	3.33%	3.33%	0	66.67%	0	0	0
Work	0	0	6.67%	0	0	3.33%	0	0	0	83.33%	3.33%	3.33%
Shop	0	0	23.33%	3.33%	0	0	0	0	0	10.00%	63.33%	0
Fan	0	10.00%	0	3.33%	3.33%	0	0	0	0	0	0	83.33%

Figure 4.8 Confusion matrix, DMM-GLAC and 2-Layer Feed Forward Neural Network

Accuracy = 76.67 %

4.2.2 Results using HON4D features and 2-Layer Feed Forward Neural Network

The HON4D descriptors directly from the segmented depth sequences were extracted. The length of the feature for each sequence was 10323 x 1. Since there were 360 depth sequences. The dimension of the feature matrix F_H was 10323 x 360. The leave one subject out cross validation test is performed as in [4].The average recognition accuracy achieved with HON4D features and 2-Layer Feed Forward Neural Network classifier is 75%. Figure 4.9 represents the confusion matrix.



	Art	Eat	Swallow	Tell	Circle	Warning	Red	Blue	Cold	Work	Shop	Fan
Art	40.00%	3.33%	10.00%	0	0	0	6.67%	3.33%	36.67%	0	0	0
Eat	0	63.33%	10.00%	0	0	13.33%	3.33%	0	10.00%	0	0	0
Swallow	0	0	83.33%	0	0	0	0	0	0	0	6.67%	10.00%
Tell	0	3.33%	10.00%	56.67%	0	20.00%	0	10.00%	0	0	0	0
Circle	0	0	6.67%	3.33%	90.00%	0	0	0	0	0	0	0
Warning	3.33%	10.00%	0	26.67%	0	60.00%	0	0	0	0	0	0
Red	0	0	0	3.33%	0	0	96.67%	0	0	0	0	0
Blue	0	0	0	6.67%	0	0	0	93.33%	0	0	0	0
Cold	20.00%	0	0	3.33%	0	3.33%	0	3.33%	60.00%	10.00%	0	0
Work	0	0	3.33%	0	0	6.67%	0	0	0	90.00%	0	0
Shop	0	0	10.00%	3.33%	0	0	0	0	0	3.33%	83.33%	0
Fan	6.67%	6.67%	0	3.33%	0	0	0	0	0	0	0	83.33%

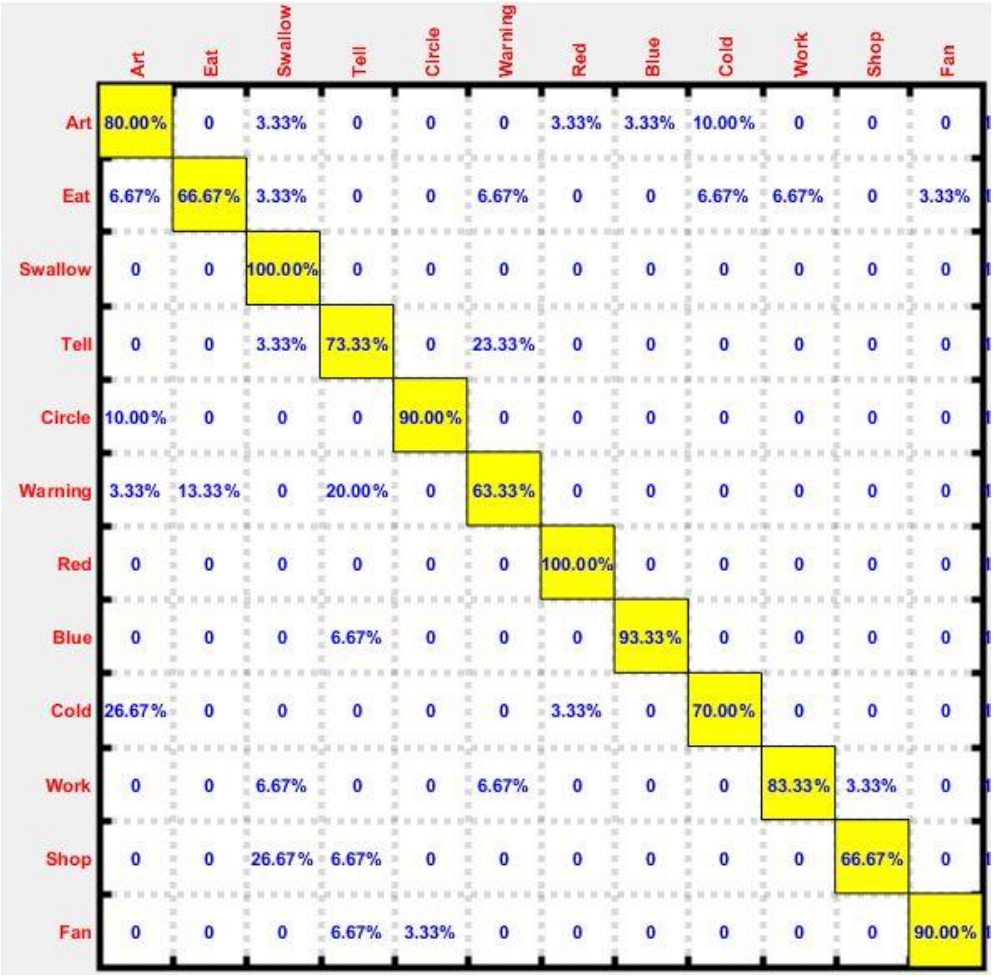
Figure 4.9 Confusion matrix, HON4D and 2-Layer Feed Forward Neural Network

Accuracy = 75 %

4.2.3 Results by weighted fusion of classifier outcomes using a Neural Network

We have fused the probability outcomes P_G and P_H from DMM-GLAC features and HON4D features respectively using the method as discussed in Section 3.4. For the experiment we have further divided the training data into training and validation set. The outputs on the validation set from the first two neural networks were combined to form the training data for the third neural network, and the outputs on the test set from the first two neural networks were combined to form the test data for the third neural network. The third neural network gave us the final output. 20 neurons are used in the hidden layer of the neural network.

The average recognition accuracy achieved by fusing the probability outputs using a neural network is 81.38 %. Figure 4.10 represents the confusion matrix.



	Art	Eat	Swallow	Tell	Circle	Warning	Red	Blue	Cold	Work	Shop	Fan
Art	80.00%	0	3.33%	0	0	0	3.33%	3.33%	10.00%	0	0	0
Eat	6.67%	66.67%	3.33%	0	0	6.67%	0	0	6.67%	6.67%	0	3.33%
Swallow	0	0	100.00%	0	0	0	0	0	0	0	0	0
Tell	0	0	3.33%	73.33%	0	23.33%	0	0	0	0	0	0
Circle	10.00%	0	0	0	90.00%	0	0	0	0	0	0	0
Warning	3.33%	13.33%	0	20.00%	0	63.33%	0	0	0	0	0	0
Red	0	0	0	0	0	0	100.00%	0	0	0	0	0
Blue	0	0	0	6.67%	0	0	0	93.33%	0	0	0	0
Cold	26.67%	0	0	0	0	0	3.33%	0	70.00%	0	0	0
Work	0	0	6.67%	0	0	6.67%	0	0	0	83.33%	3.33%	0
Shop	0	0	26.67%	6.67%	0	0	0	0	0	0	66.67%	0
Fan	0	0	0	6.67%	3.33%	0	0	0	0	0	0	90.00%

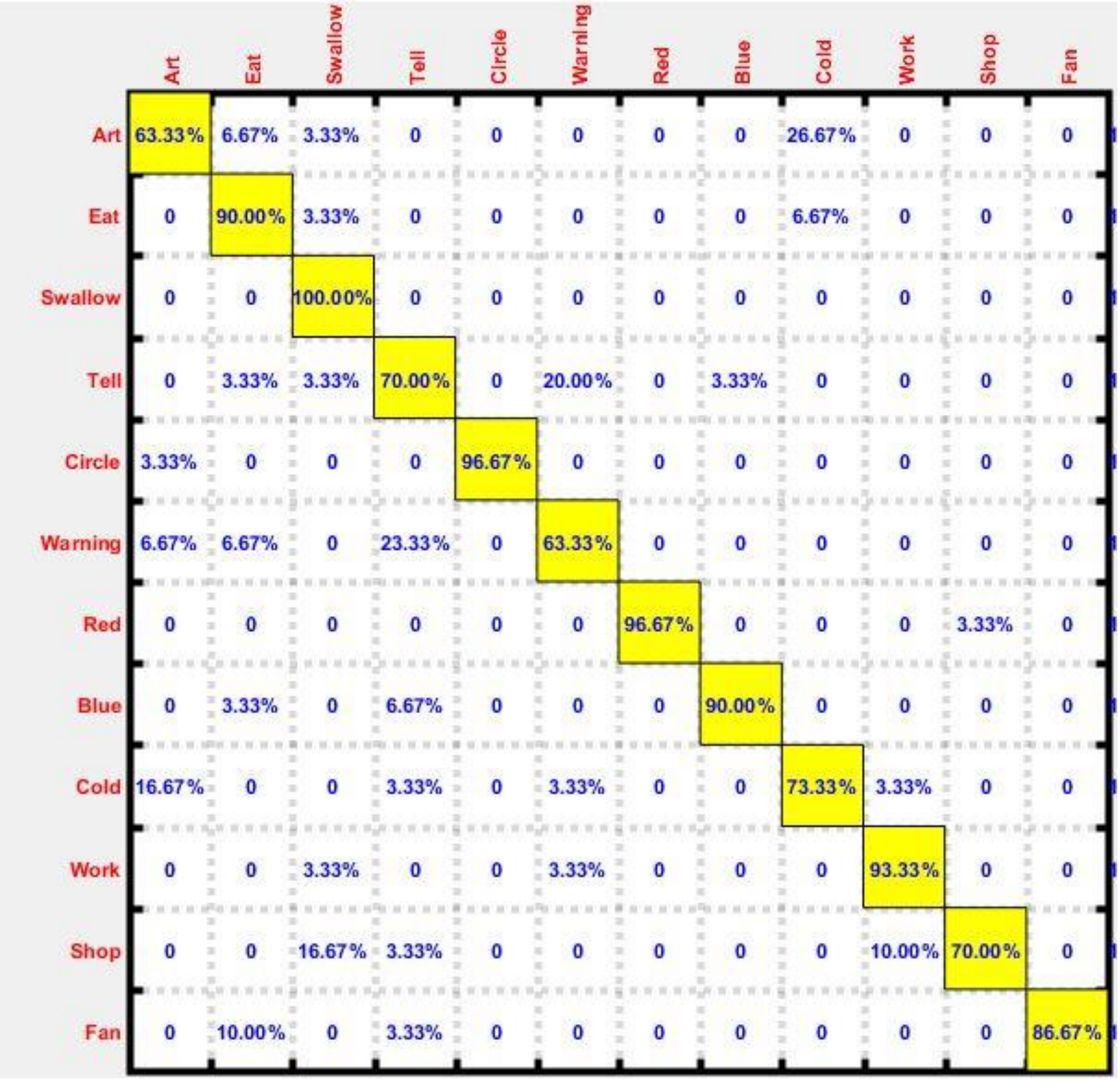
Figure 4.10 Confusion matrix, weighted fusion of outcomes using neural network

Accuracy = 81.38%

4.2.4 Results by weighted fusion of classifier outcomes using weighing factor alpha

We have fused the probability outcomes P_G and P_H from DMM-GLAC features and HON4D features respectively using the equation as discussed in Section 4.1.4.

The value of the weighing factor α is defined empirically. The average recognition accuracy achieved by this method is 82.77%.Figure 4.11 represents the confusion matrix.



	Art	Eat	Swallow	Tell	Circle	Warning	Red	Blue	Cold	Work	Shop	Fan
Art	63.33%	6.67%	3.33%	0	0	0	0	0	26.67%	0	0	0
Eat	0	90.00%	3.33%	0	0	0	0	0	6.67%	0	0	0
Swallow	0	0	100.00%	0	0	0	0	0	0	0	0	0
Tell	0	3.33%	3.33%	70.00%	0	20.00%	0	3.33%	0	0	0	0
Circle	3.33%	0	0	0	96.67%	0	0	0	0	0	0	0
Warning	6.67%	6.67%	0	23.33%	0	63.33%	0	0	0	0	0	0
Red	0	0	0	0	0	0	96.67%	0	0	0	3.33%	0
Blue	0	3.33%	0	6.67%	0	0	0	90.00%	0	0	0	0
Cold	16.67%	0	0	3.33%	0	3.33%	0	0	73.33%	3.33%	0	0
Work	0	0	3.33%	0	0	3.33%	0	0	0	93.33%	0	0
Shop	0	0	16.67%	3.33%	0	0	0	0	0	10.00%	70.00%	0
Fan	0	10.00%	0	3.33%	0	0	0	0	0	0	0	86.67%

Figure 4.11 Confusion matrix, weighted fusion using weighing factor alpha

Accuracy = 82.77%

Hence, by our proposed method of fusion using a classifier ensemble of three neural networks we achieved an average recognition accuracy of 81.38% which is better than the accuracies achieved by the two set of features individually. Even though the empirical method is giving slightly better results in this case, this empirical method of determining weights is not robust and will not yield the best results every time. In the empirical method we have to define a value of alpha where the results are optimized, since the value of alpha (α) ranges from 0 to 1 the major challenge with this method is to define the step size while calculating the value of alpha because we do not know beforehand at which value of alpha we will get the optimized results.

While our proposed method of fusion is an automatic way of calculating the weights. The reason it is giving lesser results than empirical method is because the no of classes and samples in the dataset are less and the neural network do not perform so well when less amount of data is given for training. The proposed system may outperform the empirical methods using large training data.

Table 4.2 summarizes the results on ISL3D dataset.

Table 4.2 Summary of results on ISL3D dataset

Method	**Accuracy**
DMM-GLAC and 2-layer feed forward neural network	76.67%
HON4D and 2-layer feed forward neural network	75%
Weighted Fusion (GLAC and HON4D) using neural network	81.38%
Weighted Fusion (GLAC and HON4D) using weighing factor alpha (empirical method)	82.77%

5. CONCLUSION AND FUTURE WORK

In this report a framework for sign language dynamic gesture recognition from depth sequences was introduced. For effective representation two sets of features from the depth sequences were extracted. First GLAC (Gradient local auto correlation) features from depth motion maps were extracted. Second, HON4D (Histogram of oriented 4D normal) were extracted to incorporate the loss of temporal information which is there in depth motion maps. A weighted fusion technique using a classifier ensemble of three neural networks was proposed. The proposed fusion method was compared with the empirical method of fusing outputs. The experimental results on the two datasets MSRGeature3D and ISL3D demonstrated improvement in the recognition accuracies over other state of the art approaches.

The future work may include adding more number of classes to the dataset, recognizing sentences rather than just individual words and making a real time system for sign language recognition which is the ultimate aim of this problem.

6. REFERENCES

[1] Helen Cooper, Brian Holt, and Richard Bowden, ”Sign language recognition, chapter 27”, Springer-Verlag London Limited , Visual Analysis of Humans ,pp. 539-562, 2011

[2] Ong, S.C.W., Ranganath, S.: “Automatic sign language analysis: A survey and the future beyond lexical meaning” IEEE Trans. Pattern Anal. Mach. Intell.27 (6), 873–891, 2005

[3] Siddharth, S.Rautaray, Anupam Agrawal,” Vision based hand gesture recognition for human computer interaction: a survey”, Springer, Artificial Intelligence Review, Vol. 43, No.1, pp. 1-54, 2015

[4] Alexey Kurakin, Zhengyou Zhang, Zicheng Liu, A Real-Time System for Dynamic Hand Gesture Recognition with a Depth Sensor, EUSIPCO, 2012

[5] Aggarwal JK, Lu X (2014) Human activity recognition from 3d data: a review. Pattern Recogn Lett 48:70–80

[6] Chen, Chen, et al. "Action recognition from depth sequences using weighted fusion of 2D and 3D auto-correlation of gradients features." Multimedia Tools and Applications (2016): 1-19.

[7] Kobayashi, Takumi, and Nobuyuki Otsu. "Image feature extraction using gradient local auto-correlations." Computer Vision–ECCV 2008. Springer Berlin Heidelberg, 2008. 346-358.

[8] Oreifej, Omar, and Zicheng Liu. "Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2013.

[9] http://research.microsoft.com/en-us/um/people/zliu/ActionRecoRsrc/

[10] Wang, Jiang, et al. "Robust 3d action recognition with random occupancy patterns." *Computer vision–ECCV 2012*. Springer Berlin Heidelberg, 2012. 872-885.

[11] Yang, Xiaodong, Chenyang Zhang, and YingLi Tian. "Recognizing actions using depth motion maps-based histograms of oriented gradients."*Proceedings of the 20th ACM international conference on Multimedia*. ACM, 2012.

[12] Chen, Chen, Roozbeh Jafari, and Nasser Kehtarnavaz. "Action recognition from depth sequences using depth motion maps-based local binary patterns." *Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on*. IEEE, 2015.

[13] Chen, Chen, et al. "Gradient Local Auto-Correlations and Extreme Learning Machine for Depth-Based Activity Recognition." *Advances in Visual Computing*. Springer International Publishing, 2015. 613-623.

[14] Rahmani, Hossein, et al. "Real time action recognition using histograms of depth gradients and random decision forests." *Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on*. IEEE, 2014.

[15] Yang, Xiaodong, and YingLi Tian. "Super normal vector for activity recognition using depth sequences." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014.

[16] Jetley, Saumya, and Fabio Cuzzolin. "3D Activity Recognition Using Motion History and Binary Shape Templates." *Computer Vision-ACCV 2014 Workshops*. Springer International Publishing, 2014.