ISLAMIC UNIVERSITY OF TECHNOLOGY

Depth-aware hand gesture recognition for human-computer interaction

By

Hasan Mahmud (134701)

A thesis submitted in partial fulfilment of the requirements
for the degree of Ph.D. in Computer Science and Engineering

Academic Year: 2013-2014

Department of Computer Science and Engineering (CSE)
Islamic University of Technology (IUT).
A Subsidiary Organ of the Organization of Islamic Cooperation (OIC).
Dhaka, Bangladesh.

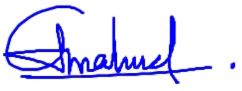
November 2021

Declaration of Authorship

I, Hasan Mahmud, declare that this thesis, titled, 'Depth-aware hand gesture recognition for human-computer interaction' and the work presented in it is my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Any part of this thesis has not been submitted for any other degree or qualification at this University or any other institution.
- Where I have consulted the published work of others, this is always clearly attributed.

Submitted By:



Hasan Mahmud - 134701
November 2021

Depth-aware hand gesture recognition for human-computer interaction

Approved By:

Md. Kamrul Hasan

Dr. Md. Kamrul Hasan
Thesis Supervisor,
Professor, Department of Computer Science and Engineering (CSE)
Islamic University of Technology (IUT), OIC.

Dr. Md. Abu Raihan Mostofa Kamal
Head of the Department,
Professor, Department of Computer Science and Engineering (CSE),
Islamic University of Technology (IUT), OIC.

Dr. Md. Hasanul Kabir
Professor, Department of Computer Science and Engineering (CSE),
Islamic University of Technology (IUT), OIC.

Prof. Dr. Chowdhury Mofizur Rahman
Vice-Chancellor,
United International University (UIU), Dhaka, Bangladesh.

Dr. Mehdi Elahi
Associate Professor, Department of Information Science and Media Studies,
University of Bergen, Norway.

Abstract

Hand gestures can be defined as the movement of the hands and fingers in particular orientations to convey some meaningful information. Recently, inexpensive depth cameras have opened ample research opportunities to work with depth-based features in parallel to image-based features. Existing computer vision-based approaches have limitations in capturing depth variations present in the fine-grained gestures and also in the coarse-grained. Hence, we got a scope to exploit depth information and use them in the machine learning models to distinguish those hand gestures correctly. In this thesis, we propose a unique depth quantization technique that can effectively distinguish different hand gestures. Using the technique first, we generate contrast varying depth images that can help to extract salient features from gestural images of static gestures. Second, we use depth values to capture hand finger movement information in the Z-direction to discriminate on-air writing tasks of English Capital Alphabets (ECAs). We have used depth-based features, like raw depth values, quantized depth values, and non-depth features like finger joint points in 2D, fingertip coordinates, other derived features from them, then merge these features to generate a unique dataset for testing the significance of depth features in terms of recognition accuracy. Experiments on both static and dynamic hand gestures showed that the proposed approach gives higher recognition accuracies. Third, to test our proposed method in deep learning settings, we design a depth-aware CNN-LSTM-based deep-learning model to recognize 14 and 28 dynamic hand gestures. The model takes gray-scale varying depth images and 2D hand skeleton joint points as multimodal input. We achieve better recognition accuracies by performing feature-level and score-level fusion techniques in the benchmark dataset.

Keyword - Gesture Recognition; Depth Information; Static Hand Gesture, Dynamic Hand Gesture, Depth Quantization, Machine Learning;

Acknowledgements

First, I would like to express my gratitude to almighty Allah for the uncountable blessings on me. I am really thankful to my supervisor Prof. Dr. Md. Kamrul Hasan for putting his trust on me and inspiring me all the way towards the degree. His immense knowledge, plentiful experience, and continuous support helped me to overcome all the obstacles arisen during the journey. Besides my supervisor, I would like to thank honorable thesis committee members for their support. I would like to thank all the users who took part voluntarily in data collection phase, help to prepare the dataset used in this thesis. Last but not the least, I would like to thank my father, mother, brother, my lovely wife and my children for their sacrifices and support in this journey. Without their tremendous understanding and encouragement over the past few years, it would be impossible for me to complete my study.
This work is supported by

Systems and Software Lab (SSL)
Department of Computer Science and Engineering (CSE)
Islamic University of Technology (IUT), OIC

Contents

List of Figures

List of Tables

I dedicate this thesis to my parents, my lovely wife, and my three wonderful daughters. I also dedicate this thesis to one of my respected teachers, late Prof. Dr. M. A. Mottalib Sir - for his fatherly affection and guidance in my personal and academic life...

Chapter 1

Introduction

Hand gesture can be defined as the movement of hands and fingers in a particular orientation to convey some meaningful information [23]. For example, pointing to some object through index fingers, expressing 'Victory Sign' or 'OK' sign, waving hands, etc. Symbolic hand gestures represent some specific symbols like OK sign or gesture that represents numeric symbol 1 (raising the index finger and bending all other fingers). In most of the cases, these gestural movements conveys single meaning in each culture having very specific and prescribed interpretations. More importantly, symbolic gestures are alternative to verbal discourse structure, different from everyday body movement which is consciously perceived. These gestures are observed in the spatial domain and are called static hand gestures characterized by the position of fingers (finger joint angle, orientation, and finger bending information). Unlike static hand gestures, dynamic gestures are considered in the spatio-temporal domain, presenting gesture as a sequence of hand shapes which includes starting through ending hand pose (e.g. hand waving, grabbing an object, pinch gesture, writing in the air, etc.). Hand gestures provide a complementary modality to speech for expressing one's ideas. Information associated with hand gestures in a conversation conveys information in degree, discourse structure, spatial and temporal structure.

There are two approaches to recognize these gestures, vision-based and sensor-based. Both of them have some advantages and disadvantages with their own research challenges. In sensor-based approaches, user needs to wear sensor enabled gloves that capture accelerometer, gyroscope, and other forms of inertial sensors data containing hand and finger movements correctly. However, this approach limits the naturalness in interactions with computers as well as noise incurred in

reading sensor values. In vision-based approaches, the problem of object segmentation from the occluded background containing various illumination conditions can reduce the recognition rate [24]. Moreover, this approach imposes restrictions on the gesturing environment, such as special lighting conditions, uncluttered background, difference in viewpoints, temporal variations, etc.

The recent advancements in stereo vision camera that utilizes depth perception from smaller to larger distances have opened a huge scope for the researchers to work with depth information [25]. Traditional web cameras do not provide the depth values (the distance of the gesturing hand from the camera). Depth information can help eliminating occlusion problems easily and can quicken the segmentation process with less error. In an occluded background, using depth information it is possible to extract the gesturing hand movement information including other important features (e.g., finger bending information). While performing gestures, (static or dynamic) gesturing hand along with different orientation of hand fingers can give motion-oriented movement information in z-direction in addition to x and y directions. Those can be represented as the most important features by effectively utilizing them in any gesture recognition system.

In the existing system, gestures, that are very close, contain overlapped fingers, joined finger parts, temporal variations, distance variations are not effectively recognized. Depth information can help generating contrast-varying depth images from low-contrast depth images. Due to the low resolution, camera provided gray-scale images do not contain enough contrast variations. So, two different gestures that vary slightly cannot be recognized. If we can quantize depth values and generate grey-scale images with variations in contrast then it can help extracting salient features. The fingers and palm of the hand occupy relatively similar depth values. We believe this apparent lack of contrast hides some meaningful information, which may be useful in gesture recognition models. For example when the fingers overlap against palm, this is not visible in the corresponding depth images. So, by quantizing depth values into specific depth levels, the contrast between fingers and palms is increased and we gain additional information. Depth values can be used to capture hand finger movement information in z-direction to discriminate gestures. Quantized depth values can be used for training to learn important information. Moreover, if we have depth-based features, like raw depth values, quantized depth values, and non-depth features like, finger joint points in 2D, finger tip coordinates, other derived features from them, then we can merge these features to generate a unique dataset for testing significance of depth features in terms of recognition accuracy.

1.1 Thesis Contributions

Considering all these observations, this thesis address the problem of constructing hand gesture recognition systems for both static and dynamic hand gestures that can utilize depth information effectively. Hence we investigate the first part by introducing depth quantization technique to generate contrast-varying depth images for static or symbolic gestures, study the impact of depth information in terms of gesture recognition accuracy. In the second part, we extend the study in dynamic hand gesture recognition by introducing depth-based features to be considered in the classification of on-air writing of English Capital Alphabets (ECA). In the last part, we consider depth-based multimodal gestural input in deep learning setup. The main contributions of this thesis are summarized below:

- **Recognition of static hand gestures using contrast-varying depth information**: We introduce a depth quantization process with the help of depth information provided by Microsoft Kinect depth camera. We generate contrast varying grey-scale depth images according to the depth map to utilize local shape information of the gesturing hand fingers. We have applied Scale-Invariant Feature Transform (SIFT) algorithm to achieve image invariant properties. The algorithm takes the generated depth silhouettes as input and produces robust feature descriptors as output. These features (after converting into unified dimensional feature vectors) are fed into a multiclass Support Vector Machine (SVM) classifier to measure the accuracy. We have tested our results with a standard dataset containing 10 symbolic gestures representing 10 numeric symbols (0-9). After that we have verified and compared our results among depth images, binary images, and images consisting of the hand-finger edge information generated from the same benchmark dataset. Our results show higher recognition accuracy while applying SIFT features on depth varying images.

- **A depth-aware dynamic hand gesture recognition system that captures the motion-oriented movement information in the process of on-air writing**: We have captured hand finger motion information using a depth camera and represented them as depth images for each ECA. We represented the hand trajectories, that is, the hand movement sequence as a series of data points (x_t, y_t, d_t), where (x_t, y_t) is the position of the hand and d_t is the depth value at time sequence t. We extend our study of depth information utilization technique to this dynamic hand gesture recognition

system by applying depth quantization process in the air-writing sequences. All the data points were converted into the time-series representation of a particular alphabet suitable for extracting important features. We merge the depth features and the non-depth features and generated 12 datasets to understand the significance of depth information from different perspective like considering all the features, taking only the depth features, taking the re-sampled features, and taking features after correlation analysis. Thus we use integrated features consisting of quantized depth values, camera provided depth values, and non-depth features represented as DTW (Dynamic Time Warping)-based distance features, fed them into a machine learning model. We found that features with depth information contribute more to recognition accuracy for all the feature selection techniques we applied.

- **A multimodal two-stream deep-learning-based dynamic hand gesture recognition system using depth information:** We have introduced multi-modal input consisting of depth-varying grey-scale images (quantized depth images) and 2D hand skeleton joint points from benchmark DHG14/28 dataset. Spatio-temporal information of dynamic gestures was captured in CNN-LSTM-based architecture consisting of two sub-networks. CNNs are designed to detect spatial patterns related to the position of the skeleton joints in 3D space and the LSTM is used to capture the spatio-temporal patterns related to the time evolution of the 2D coordinates of the finger joints. We perform a feature-level and score-level fusion-based CNN-LSTM deep-learning model consisting of multimodal input that achieved better recognition accuracy.

1.2 Thesis Outline

The rest of the thesis has been organized as follows: In chapter 2, we describe the background study and literature review on both static and dynamic hand gesture recognition technologies and related research works, then in chapter 3 we give detail description of our proposed approach in static hand gesture recognition, chapter 4 presents the recognition of on-air writing activity as a dynamic gestural event using depth information, then we elaborate how our proposed method has been extended to experiment dynamic hand gesture recognition task in deep learning-based approach in chapter 5, at last, we give concluding remarks of the thesis work in chapter 6.

Chapter 2

Literature review

Nowadays, hand gesture recognition in gesture-based interactions is a prominent research area which has a huge impact in the design and development of many HCI applications. Human hand gestures, after processing, can be considered as alternate input modality while interacting with the computers. In computer vision, understanding and recognition of the hand gestures accurately has attracted the attention of many researchers. Moreover, recently introduced low-cost depth cameras has opened ample research opportunities in the area of computer vision, machine learning, and HCI.

In this chapter, we discuss on types of hand gesture, hand gesture recognition approaches, and their applications. Then we discuss on depth information acquisition, depth cameras, depth-based benchmark datasets of hand gesture recognition, and features used in hand gesture recognition. Finally, we review the state-of-the-art approaches of static and dynamic hand gesture recognition based on depth information including relevant research challenges.

2.1 Hand Gesture Recognition

Hand gesture is the translation of gestural language in to meaningful information that is produced by the hand finger movements, orientations, shapes constructed by the hand. In HCI applications, these hand gestures should be recognized correctly so that those can be used to map gestural movements into corresponding computer interactions like giving instructions or commands to the computer applications, manipulating virtual objects, exploration of virtual world, doing natural conversations through gestures and so on. Interpretation of the hand movements

need to be understood correctly by the gesture recognition system. The gestures produced in the real-world 3D space need to be understood by the system in an efficient way. There are basically two types of hand gesture as mentioned in the introduction, static and dynamic hand gestures. No matter whether it is static or dynamic hand gestures, the task of a recognition system is to map the observed input to a particular hand configuration (e.g. Numeric symbol '7' of static hand gesture or writing capital English alphabet 'C' in the air of dynamic hand gesture). In computer vision, the gestural input could be in different forms like 2D RGB image, 3D depth image, hand skeleton joint points in 3D provided by depth camera. Those inputs are presented to the recognition system as distinctive features extracted using various relevant feature selection techniques. Hand gesture recognition consists of detection and analysis of hand gestures from the information captured from RGB cameras, depth cameras, different sensors or wearable inertial sensor. In general, the task of traditional vision-based hand gesture recognition system consists of the following steps:

1. Gestural image input: Taking input of the hand gesture from one or more input sources. The input could be images from 2D/3D cameras or input could be hand skeleton joint point provided by depth camera (e.g. MS Kinect, Intel RealSense).

2. Preprocessing: Calibration of the RGB and Depth images, segmenting hand region of interest (ROI), filtering noises, representing depth images based on depth map information are few important steps in depth-based hand gesture recognition system. Data normalization, augmentation, transformation are also part of the preprocessing step.

3. Feature selection and extraction: This step includes suitable feature selection and extraction techniques to generate the training set. The input data (e.g. the RGB image or hand skeleton joint points from depth camera) are transformed into a representation suitable for feature extraction. After extracting suitable or appropriate features the training datasets are constructed.

4. Classification: A machine learning model or a combination of machine learning models is used to recognize unknown gestures into an expected gestural class label. Cross-validations are performed to report the accuracies.

Steps of a typical hand gesture recognition system is shown in Figure 2.1.

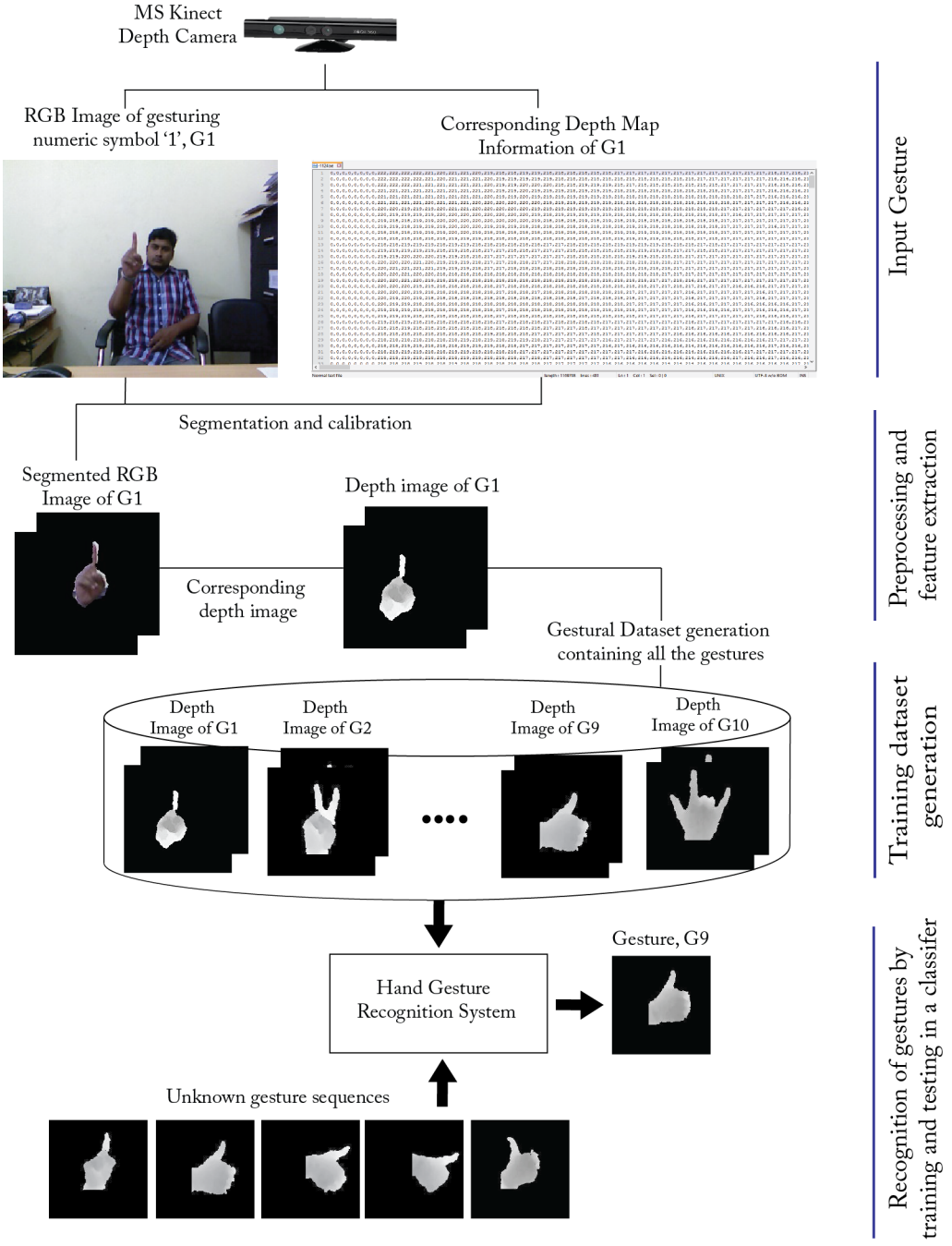


FIGURE 2.1: A typical hand gesture recognition system

In this thesis, we focus on computer vision-based hand gesture recognition that are performed in natural environment and the gestural hand contains x, y image coordinates as well as depth values in z dimension.

2.2 Depth Information acquisition and gesture representation

There are different approaches to capture hand gesture inputs. Computer vision-based approach imposes restrictions on the gesturing environment, such as special lighting conditions, simple and uncluttered background, and occlusions (the gesturing hand is occluded by other parts of the body) [23]. Traditional web cameras do not provide the depth values (the distance of the gesturing hand from the camera) so getting the full 3D information of the observed scene where gesturing events need to understand is a major challenge in computer vision. Observing the same gestural scene from two different viewpoints using 2D camera and then calculating disparity information to estimate depth information requires a lot of computational effort. However, the recent emergence of depth sensors has given an opportunity for the researchers to utilize the depth information in order to overcome those challenges. The depth data streams are provided by different depth sensors, like Microsoft Kinect [26], Intel Real Sense [27], ASUS Xtion Pro [28], etc. Figure 2.2 shows few of the depth sensors released recently. The depth information corresponding to the hand gesture images has given new dimensions to conduct research in hand segmentation process, finger identification techniques, finger joint detection, and finger tracking. Depth data can help eliminating occlusion problems easily and can quicken the segmentation process with less error.

These depth sensors provides depth-map information for each pixel from which the 3D information of the scene can be estimated. Depth map information actually stores a distance value (Z-value) for each pixel (X, Y) in an image. This depth value are represented by 8-bit values between 0-255. 0 value represents the most distant possible depth value and 255 is the closest possible depth value in the image. There are different techniques and methods to calculate depth by different depth cameras.

Microsoft Kinect version 1.0 uses Light coding technique where light-encoded infrared transmitter emits a 'stereo code' with three-dimensional depth (Figure 2.3).

(a) Microsoft Kinect Version 2.0

(b) Intel RealSense D435i

FIGURE 2.2: Example of RGB-D camera for depth information acquisition (a) Microsoft Kinect V2 (b) Intel RealSense D435i

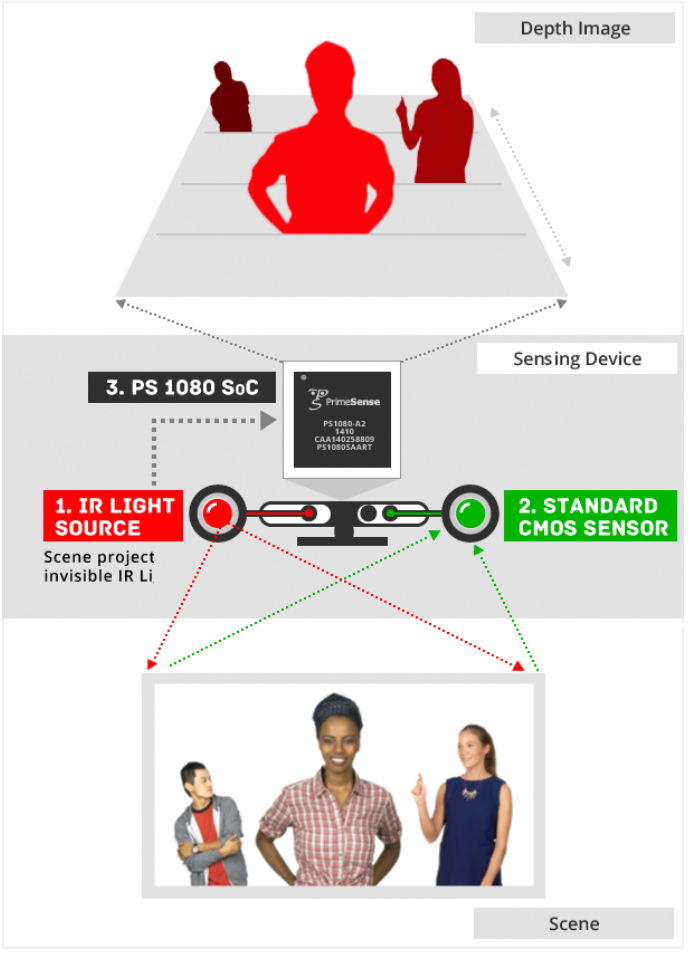


FIGURE 2.3: Depth perception uisng Structured light technique (e.g. MS Kinect V1)[3]

Kinect version 2.0 uses time-of-flight technique to estimate depth. A light signal is emitted to the scene, a receiver computes the distance of the object based on the elapsed time between the light emission and reception (Fig. 2.4). A recent survey on hand gesture recognition devices and their different applications can be found in [23, 25, 29, 30]. They have given the comparison of the devices from a survey based on different criteria like, type and number of objects could be detected, tracked, sensor implementation technologies, availability of documentations, and advantages-disadvantages of the depth sensors, etc. [29].

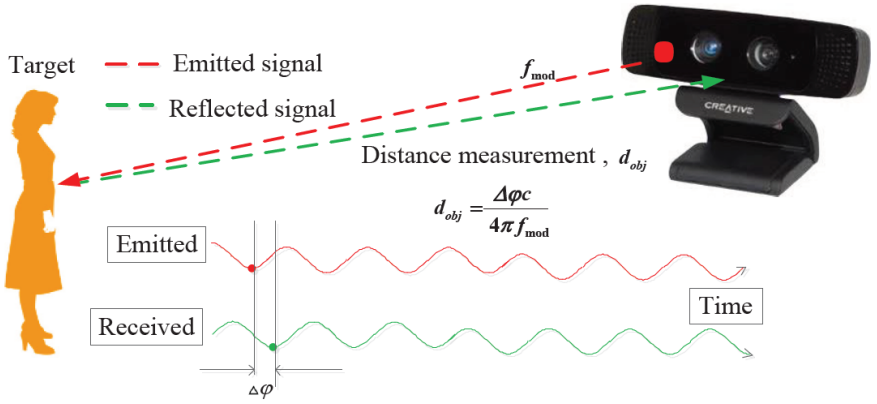


FIGURE 2.4: Depth perception uisng Time-of-Flight (ToF) technique (e.g. MS Kinect V2, Intel RealSense)[4]

Computer vision-based hand gestures can be represented in to two ways, one is 3D model-based method and the other one is appearance-based method [23]. Hand skeleton model provided by the depth camera is a 3D model-based representation that contains 3D spatial information of human-hand-joint points along with temporal information [25]. In appearance-based methods, gestures are represented using color-based model, silhouette geometry model, etc. The geometric properties of hand skin color, silhouette properties such as perimeter, convexity, bounding box, centroid, etc. are used to represent hand gestures. These representations of hand gestures help to extract discriminating properties to learn by different machine learning algorithms.

2.3 Application areas of depth-based hand gesture recognition system

A lot of HCI applications have already been emerged based on depth estimation or depth-map information. They have been employed in object recognition, tracking,

3D reconstruction, Augmented Reality (AR) based games, human activity recognition, sign language recognition, and so on [31, 32]. In the literature, gesture recognition has been used in varieties of applications where depth data become crucial to recognize hand gestures effectively [33]. A real-time low-cost character animation system was introduced in [31] that reduces manual post-processing tasks using Kinect as the depth-data acquisition device. A quick and accurate human pose recognition system has been proposed in [32] using single depth image acquired by Kinect. Figure 2.5 shows few common HCI applications that requires hand gestures to interact with the systems.

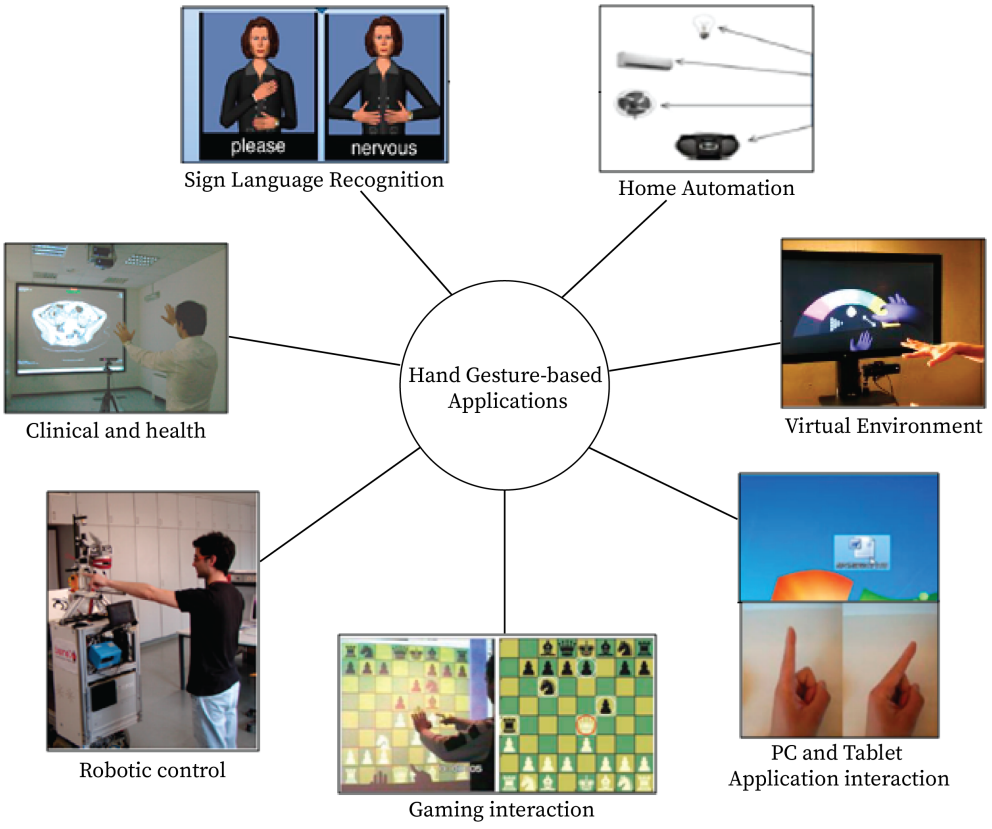


FIGURE 2.5: Few application areas of hand gesture-based interactive systems (image adapted from [5, 6, 7, 8, 9, 10, 11])

2.4 Datasets for hand gesture recognition

The depth sensors can capture and track the full body motion, body skeleton joint points to provide important information directly from the software operated devices. However, hand gesture recognition may not require to track the full body information compared to human activity recognition and analysis [34]. The datasets

required for hand gesture recognition may focus on the specific hand region of the gesturing hand. Recently, the depth-based hand gesture datasets captured by different depth sensors are becoming publicly available to analyze both static and dynamic hand gestures which gave the researchers' an ample opportunity to work with depth information.

As the main focus of the thesis is to cover depth-based hand gesture recognition so, depth sensor-based static and dynamic hand gesture recognition datasets will be reviewed here. Most of the datasets are referred from the recent publicly available datasets. In the literature a large number of datasets have been constructed with the help of MS Kinect or Intel RealSense. Unfortunately, the precise hand-shape-context information are not that much reliable due to its low resolution structure. The recent depth camera-based datasets provides depth-map information including RGB image and hand skeleton joint points in 2D or 3D [34].

A depth camera-based static hand gesture dataset named as 'NTU hand digit dataset' was created by Ren et. al. in [2], is a benchmark dataset in static hand gesture recognition. The dataset was collected using Kinect depth camera from 10 subjects. Each subject has performed 10 static or symbolic gestures 10 times. So, the dataset contains total of 1000 instances. Each gesturing instance contains a color image and the corresponding depth map. The dataset was prepared in a very challenging real-life environment containing the situations like the cluttered background and pose variations in terms of rotation, scale, orientation, articulation, changing illumination, etc. They have utilized depth-map information to improve the hand segmentation process and represented each gestural hand shape image as time-series contour curve. An overview of their time-series data generation from the gestural dataset is shown in Figure 2.7. They applied distance-based matching algorithm named as 'Finger Earth Movers Distance (FEMD)' to recognize gestures.

Depth sensor-based static hand gesture datasets to recognize American Sign language (ASL) have been used by the researchers in [15, 35, 36]. The 'HUST-ASL' dataset used in [35] and [15] consists of 34 hand gestures generating numeric digits from 0 to 9 and 24 English characters with 16 different poses with the help of 10 participants. The dataset contains 5440 RGB images and their corresponding 5440 depth-map which were superimposed to generate RGB-D images for classification. The fusion of color and depth images are fed directly in the classification model to predict the signs. The description of the ASL-FS dataset can be found in [36]. A static hand-gesture dataset for human-robot interaction introduced in

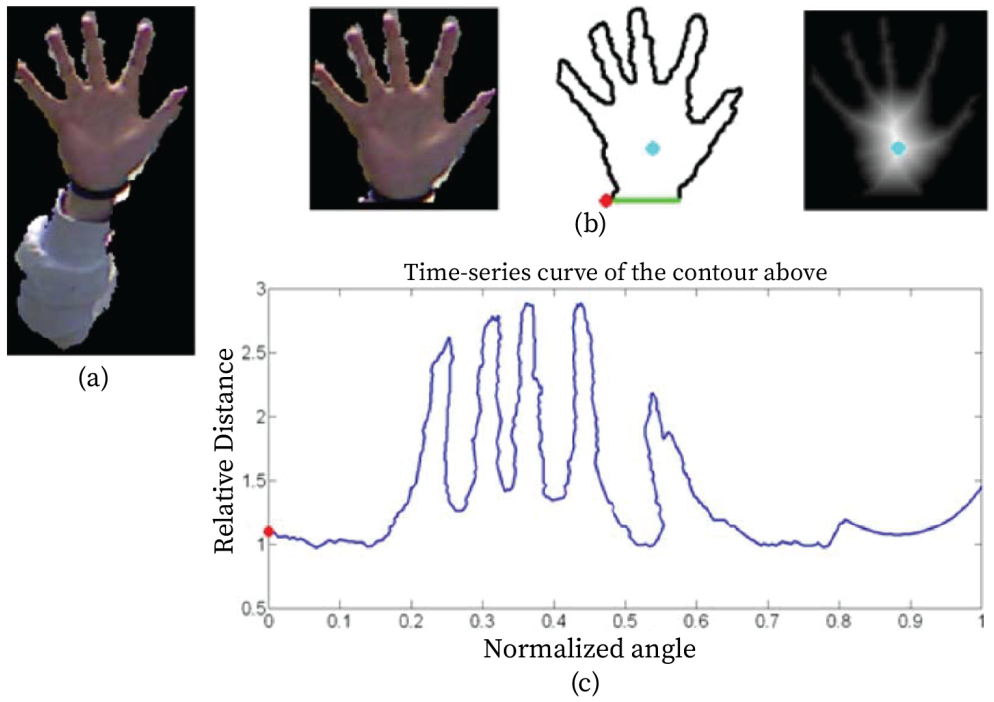


FIGURE 2.6: Time-series data generation of the gestural image (a) depth-thresholding-based hand segmentation (b) A more accurate hand detected with black belt (the green line), the initial point (the red line) and the center point (the cyan point); (c)Its time-series curve representation). Image reproduced from [2]

[12] contains 15 static gestures as show in Figure . The dataset contains spatially and temporally adjusted RGB images, depth images, and annotated bounding box coordinates in separate files.

Along with the static hand gesture datasets, recently, there are number of depth-based dynamic hand gesture datasets came into research focus. Researchers are working on these dynamic hand gesture datasets mostly by applying deep-learning-based methodologies to achieve higher recognition accuracies. However, to deal with research issues like self-occluded small hand articulated parts, low resolution depth images, capturing gestural motion information researchers are trying hard to consider these challenges in their dataset preparation. Many of the recent dynamic hand gesture datasets have been reported in [34, 37].

Different indoor and outdoor related activities in the form of hand gestures were captured from 50 subjects in a multimodal large-scale dataset named as EgoGesture [38] dataset. There were 2081 RGB-D videos, 24161 samples, and 2953224 number of frames containing 83 gestural categories captured using Intel RealSense SR300 camera. It is a publicly available gesture dataset that covers different daily activities, actions, and interaction with objects and other human. The NVDIA

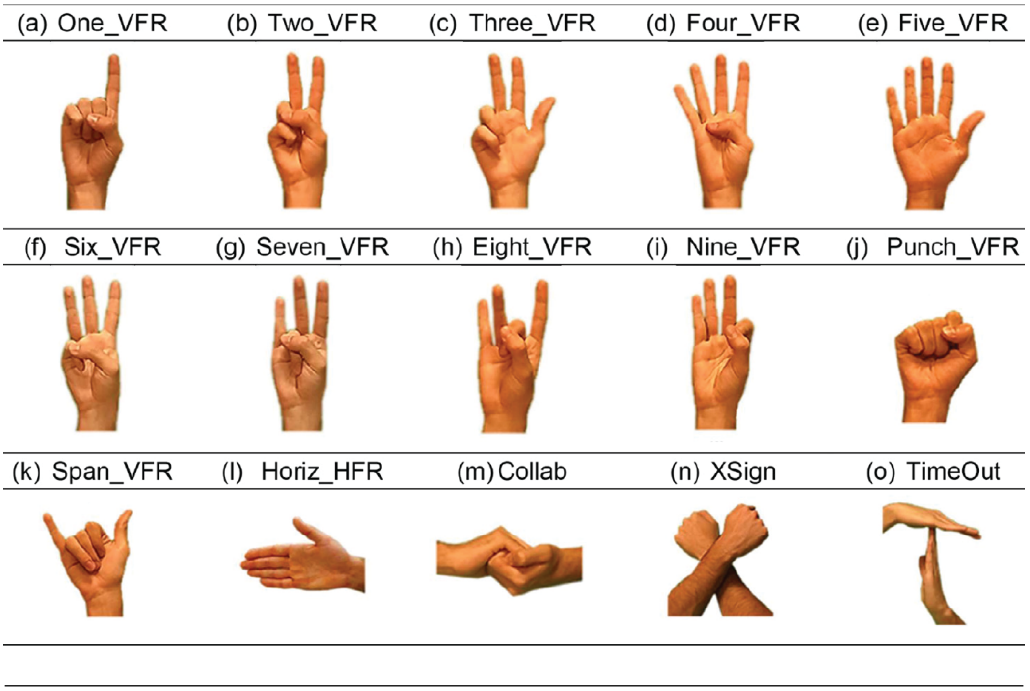


FIGURE 2.7: 15 static front right-hand gestures. Image reproduced from [12]

dataset presented in [13], contains 25 dynamic hand gestures from simulated card driving scene. It contains 5 input modalities consists of RGB image, optical flow image, depth image, IR image, and IR disparity image. They captured RGB and depth videos using SoftKinetic DS325 sensor and a top-mounted DUO 3D sensor for the IR streams. The optical flow images were generated from RGB image and IR disparity image from IR-stereo pair as shown in Figure 2.8. However, hand skeleton joint points were not considered in their dataset.

Another hand-skeleton-based depth dataset presented in [34], contains 14/28 dynamic gestures divided into fine-grained and coarse-grained gestures captured using Intel RealSense camera. The list of gestures are given in Table 5.2. This dataset has came into research focus after commencing at 2016. In this dynamic gesture event, there are also few datasets that could be found in air-writing domain as used in [39, 40]. However, these air-writing datasets are not depth camera-based datasets. A depth-camera captured air-writing dataset can be found in [18].

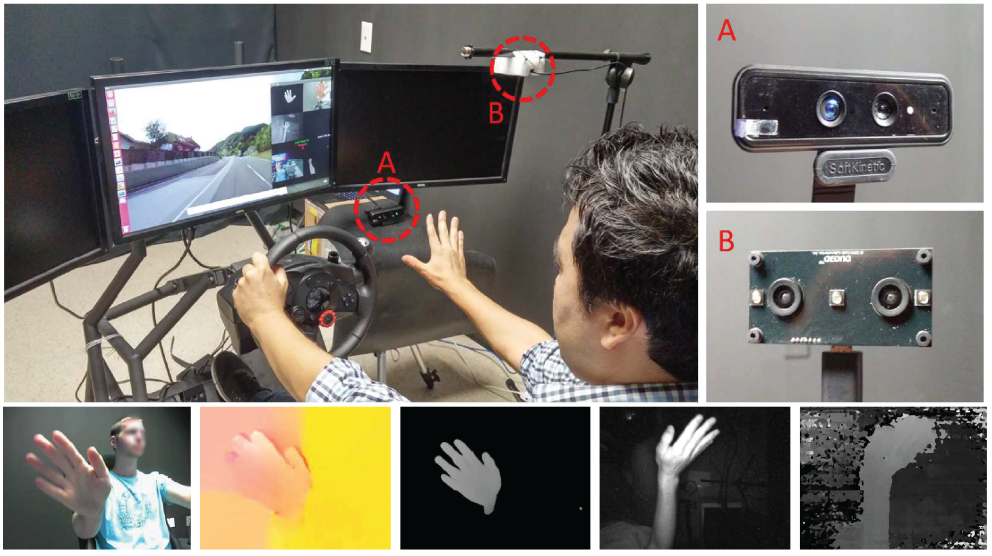


FIGURE 2.8: Environment for data collection. (Top) Driving simulator with main monitor displaying simulated driving scenes and a user interface for prompting gestures, (A) a SoftKinetic depth camera (DS325) recording depth and RGB frames, and (B) a DUO 3D camera capturing stereo IR. Both sensors capture 320×240 pixels at 30 frames per second. (Bottom) Examples of each modality, from left: RGB, optical flow, depth, IR-left, and IR-disparity. Image reproduced from [13]

TABLE 2.1: Dynamic hand gesture list in in DHG 14/28 Dataset

Class	Gesture	Grain
0	Grab	Fine
1	Tap	Coarse
2	Expand	Fine
3	Pinch	Fine
4	Rotation Clockwise	Fine
5	Rotation Counter-clock	Fine
6	Swipe Right	Coarse
7	Swipe Left	Coarse
8	Swipe Up	Coarse
9	Swipe Down	Coarse
10	Swipe X	Coarse
11	Swipe V	Coarse
12	Swipe +	Coarse
13	Shake	Coarse

2.5 Related work on static and dynamic hand gesture recognition

Human hand is a highly articulated model, prominent in making deft poses. To recognize those hand poses many research works have utilized RGB cameras and

applied either template-based approaches or model-based approaches on RGB images. Conventional RGB image-based gesture recognition techniques need to consider many research challenges, such as light sensitivity, cluttered background, and occlusions. However, the recent emergence of depth sensors has given an opportunity for the researchers to utilize the depth information in order to overcome those challenges. From the depth sensors, the most common features used in static hand gesture or posture recognitions [41] are skeleton joint positions, hand geometry, hand-finger shape, area, distance features, depth pixel values, etc. Generally, these features can be categorized as local features or global features. The major challenges of these feature descriptors are variations of gesturing hands while articulating an emblem or symbolic gesture. In case of static hand gesture, a gesture may slightly differ in terms of hand shape and size, variations in translation, or rotation of the fingers for the same gesture.

A robust hand gesture recognition system should be invariant to the scale, speed, and the orientation of the gesture performed. From the depth-image-based dataset of static hand gesture recognition described in section 2.4, did not consider depth information as important features in gesture recognition. They have considered depth-map information for effective and robust segmentation process.

2.5.1 Hand Segmentation and localization

Hand segmentation and localization is one of the fundamental steps of both static and dynamic hand gesture recognition system. Almost all the depth information-based hand gesture recognition approaches as described the pre-processing step in section 2.1, contains hand segmentation and localization based on depth values provided by the depth camera. To extract hand ROI exactly and quickly depth-thresholding-based techniques are recently applied among the state-of-the-art gestural hand segmentation techniques: Skin-color-based, background subtraction, and depth-based segmentation [2, 15, 34, 42, 43]. A typical depth-based hand segmentation process is shown in Figure 2.9.

An attention network-based hand localization method has been introduced in [14] to bypass the segmentation process. Input of the network consists the fusion of RGB image and Depth image. The network gradually focuses on hand ROI from the image which is later optimized for the classification tasks. Example hand localization procedure shown in Figure 2.10.

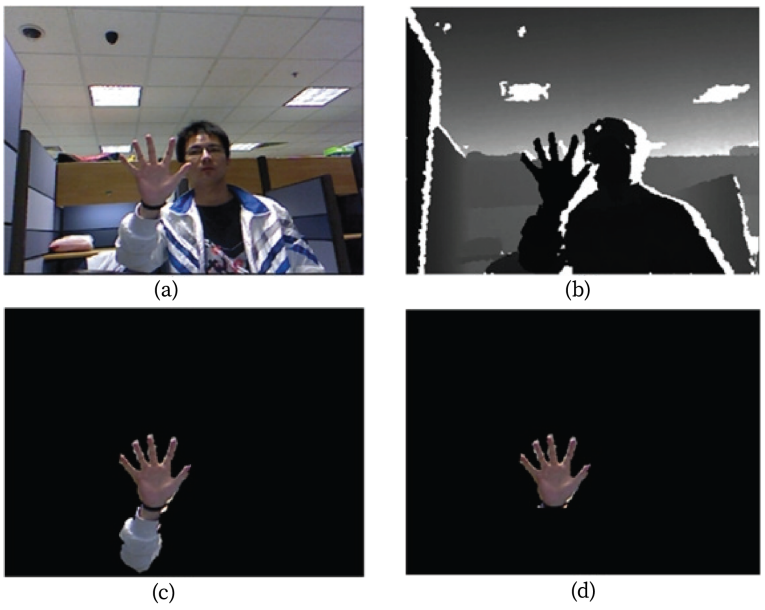


FIGURE 2.9: Hand Detection process. (a). The RGB color image captured by Kinect Sensor; (b). The depth map captured by Kinect Sensor; (c). The area segmented using depth information; (d). The hand shape segmented using RGB information. Image reproduced from [2]

FIGURE 2.10: Examples of localization results. There are five columns in total, which represent five different hand gestures we randomly chose in one subject from HUST-ASL. Each line represents the results at different iterations, which are 20 0, 40 0, 20 0 0, and 40 0 0, respectively. Green/red rectangles indicate the highest weighted proposals computed by our attention network. Green represents good localization results, while red represents unsatisfactory results. Image reproduced from [14]

An improved segmentation process described in [44] combines depth and color information using a hierarchical scan-based method and then local neighbor method. This method is suitable to give segmentation result up to two meters. A depth range-based hand segmentation were used after Otsu's global thresholding on color image as reported in [45]. A binary classification-based approach using Random Decision Forest (RDF) is applied to classify each pixel of depth information provided by the camera either contains hand or the background. However, this task is not suitable for real-time applications [46].

2.5.2 Representing gestural features to machines

Natural and contactless communication between human and machines require computer vision-based techniques. However these techniques impose lot of challenges due to illumination variations, background changes, occluded and complex scenes, processing time, frame rate, resolution, skin-color confusions, and so on. Depending on the recognition approaches like, color-based recognition, skeleton-based, motion-based, depth-based, 3D model-based, deep-learning-based, the features extraction and representation process also varies [45]. Features that are extracted manually called hand-crafted features required novel and complex functions which requires a great computational capacity. Features could also be extracted automatically due to the recent availability of computational resources using deep learning algorithms. Here we mostly describe hand-crafted features used in gesture recognition rather than automated features. However, the input modality for automated feature extraction plays significant role in gesture recognition.

2.5.2.1 Spatial features

Gestural features that contains spatial information to represent gestures can be considered as spatial features. For example hand shape and size descriptor, color descriptor, hand contour representation, hand finger positions, hand finger joint point positions, and so on [47].

Color-based features like color marker, skin color angle, non-skin color angle from the hand shape region, etc can be found in the literature [48, 49]. However, they suffer from occlusion, presence of cluttered or complex background. A free-main chain code based hand shape features were used in [48] to recognize Indonesian Sign

Language. Histogram of oriented gradients (HOG) is considered as one of the basic features in computer vision. Haar-like feature was used in real-time vision-based hand gesture recognition in [50]. It actually describe the hand posture with the computation of 'integral image'. Finger-emphasized multiscale descriptor (FMD)–Dynamic time warping (DTW)-based multiscale feature descriptor has been proposed in [51], found to be robust in RGB-D based static hand gesture recognition. The FMD describe the features in 3-scale representation of time-series data. A recent review of different features to recognize static hand gestures or hand posture is reported in [25], that mentioned techniques to extract certain features like, Wavelet Transform, Fourier Coefficients of Shape, Zernic Moment, Gabor filter, Vector Quantization, Edge Codes, Hu Moment, and so on. Static hand gestures actually contains spatial patterns to learn by the machine learning algorithms. These spatial feature are extracted in different methodologies as found in another recent literature survey in [47]. Hand finger shape-based feature extraction proposed in [52], combines number of finger tips, angle between fingertips and hand gravity center, and Scale Invariant Feature Transform (SIFT) features.

In [2], the author used depth camera provided depth values to segment the hand region of interest. They have generated hand shape contour image in the form of time-series curves. The process of generating time-series curve is shown in Figure 2.7 and their proposed approach is shown in Figure 2.11.

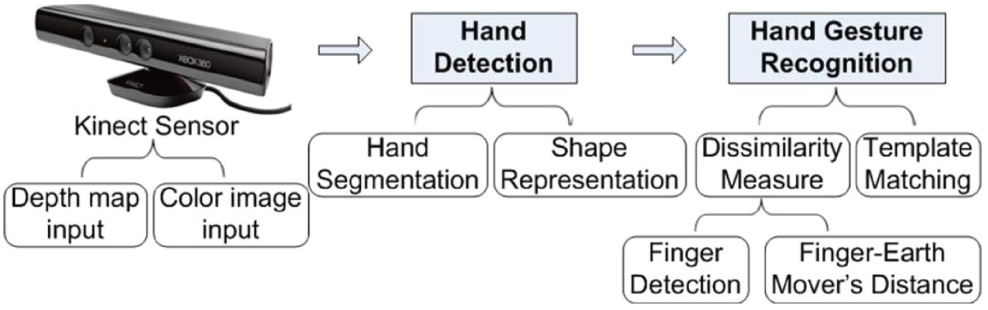


FIGURE 2.11: Proposed framework of part-based static hand gesture recognition. Image reproduced from [2]

The extracted curves of gesturing finger parts were matched using a distance-based template matching algorithm called FEMD. They represented the shape of hand fingers as a global feature (the finger cluster) by analyzing time-series curve generated from binary image. In the curve, the Euclidean distance between each contour point and the center point is considered in one dimension and the angle of these contour points made with the initial point relative to the center point is considered as another dimension. The time-series curve of the topological

hand shape considered as finger parts and matches those fingers only, not the whole hand shape. Features only from opening finger parts may not give good recognition results. Rather features including bending finger parts as local features will play a significant role to improve the recognition accuracy.

Researchers in [36], used depth images to recognize static hand gestures using parallel convolution neural network (CNN) architecture for human-robot interaction. Their architecture takes input from two channels, one from RGB images and the other one from the corresponding depth images. They considered images from different backgrounds and different illumination conditions. However, to achieve image invariant properties like scale, rotation, translation they relied on automated feature extraction using CNN from low contrast 100×100 RGB and depth images. To recognize close or fine-grained gestures where they varies with respect to small changes in the finger local shape information, we can not rely fully on automated feature learning. Rather, some sort of pre-processing in the image depending on the available image information (e.g. depth) could significantly improve the recognition accuracy.

Another work in [15], applied Depth Projection Map (DPM) and Bag-of-Contour-Fragments (BCF) named as DPM-BCF method to classify different datasets in [2, 53]. They have not considered robust key-point-based descriptors for training and testing. The depth-map is projected into three orthogonal planes to generate front, side, and top view of the same gestural image in to three binary image. The process diagram is shown in Figure 2.12.

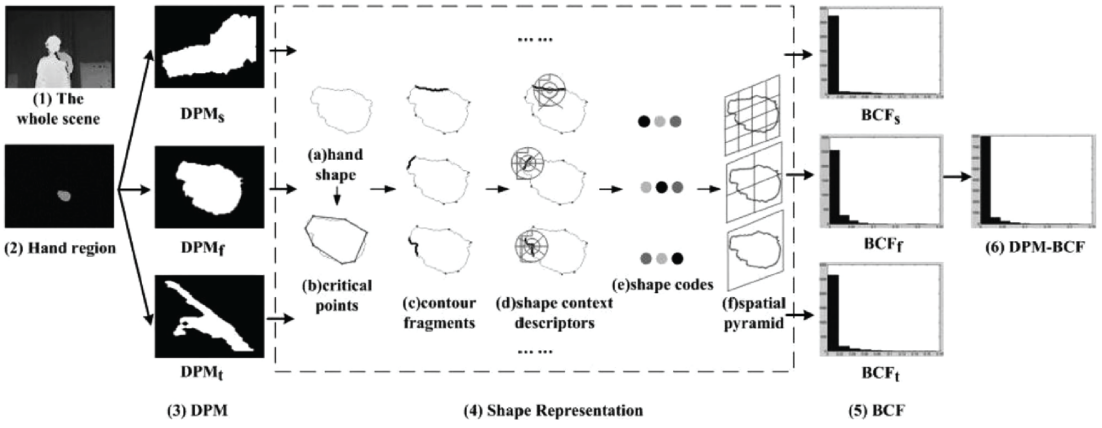


FIGURE 2.12: Pipeline of building shape representation in DPM-BCF, which is extracted from the depth map of each projection view. The middle box illustrates the procedure of building shape representation: (a) contour of the hand; (b) critical points detected using DCE; (c) some contour fragments in thick black color; (d) using shape context to describe each contour fragment; (e) shape codes; and (f) using 1×1, 2×2, and 4×4 spatial pyramid for max-pooling. Image reproduced from [15]

They achieved highest accuracy with faster computation preserving shape and topological information. However, the depth-map projections are the spatial information which are difficult to generalize and consider for temporal event (e.g. dynamic hand gesture).

2.5.2.2 Temporal features

Compared to the static hand gestural features that contain hand description from a single image, dynamic hand gestural features contain time-driven hand motion information. These temporal motions need to be extracted from a sequence of hand shapes rather than a single image. Dynamic hand gestures contains spatial as well as temporal information in the sequence of images that are presented as spatio-temporal features. However, image sequences consisting of spatial information can also be fed in to a temporal classifier.

Several feature representation techniques were utilized in case of dynamic hand gesture recognition like 3D depth information from the hand region, point clouds [54], localized hand finger joint positions [55], spatio-temporal HOG2 descriptor [56], histogram of 3D facets, N-gram model, dynamic programming on depth maps [57], and so on as described in [17]. Researchers in [58] represented dynamic hand gestures as global rotation and global translation features by wrist joint, palm joint, and metacarpophalangeal (MCP) joints. They also concatenated these global motion features with finger motion features as final feature set. To learn automated spatio-temporal features, a 3DCNN-based feature extractor was applied in Arabic Sign Language (ArSL) gestures. Motion Fused Frames (MFFs), a technique to append optical flow frame and color frame to generate Fused spatio-temporal frames representing gestures as shown in Figure 2.13.

To represent a motion trajectory of dynamic gesture the researchers in [59], extracted hand position, velocity, and angle consisting of 4-dimensional feature vector for each standard gesture. For gestural sequence modeling, recurrent neural networks (e.g. RNN, LSTM) are highly used to capture temporal relationships in the frames. Previously, the handcrafted features were extracted containing gestural sequences and classifiers such as Hidden Markov Models (HMMs), Dynamic Time Warping (DTW) were used to recognize spatio-temporal gestures. Each gestural sequence can be considered as a finite number of probabilistic states. The state transitions represent the hand positional changes in a gesture sequence [60]. Researchers in [17], worked with various hand gesture features derived from two

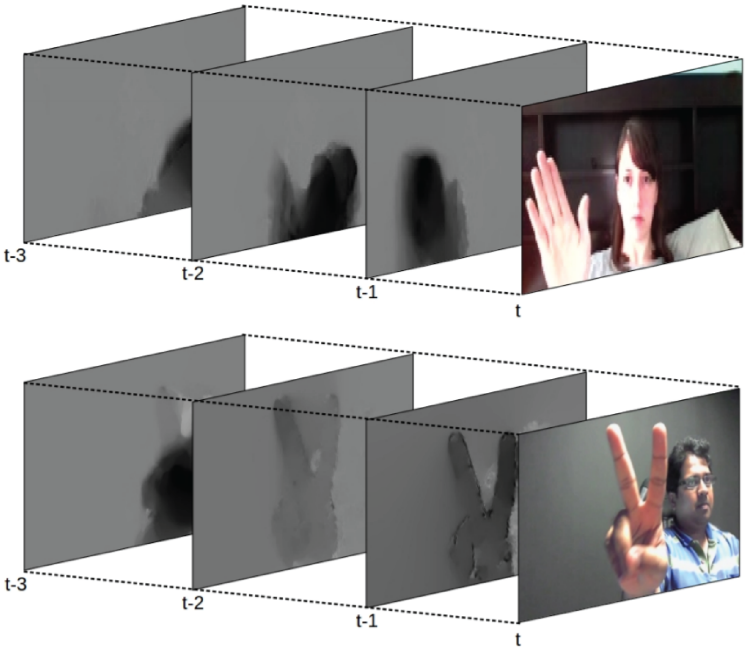


FIGURE 2.13: Motion Fused Frames (MFFs): Data level fusion of optical flow and color modalities. Appending optical flow frames to static images makes spatial content aware of which part of the image is in motion and how the motion is performed. Top: Swipe-right gesture. Bottom: Showing two fingers gesture.. Image reproduced from [16]

modalities with depth and skeleton data. Figure 2.14 shows the overview of the features of dynamic hand gestures. They have extracted five types of features to exploit robust features in dynamic hand gesture recognition. The features are motion features, skeleton shape features, normalized hand skeleton features as handcrafted features and joint point-cloud features, hand depth shape features are automated features. They claim their contribution in using the pairwise joint distance instead of using the Shape of Connected Joints (SoCJ) features as proposed in [61]. To record the temporal information while air-writing, the researchers in [18], used geometric shape features from the writing trajectory determined by collecting the finger tip points into the frame sequences. To represent the feature vector they use the slope sign variation at points on the trajectory and based on the sign they define the points as critical points. An example is shown in Figure 2.15.

2.5.3 Multimodality in hand gesture recognition

Multiple input modalities of the same gesture can be considered as multimodal input from various channels like RGB images, depth maps, and hand skeleton joint

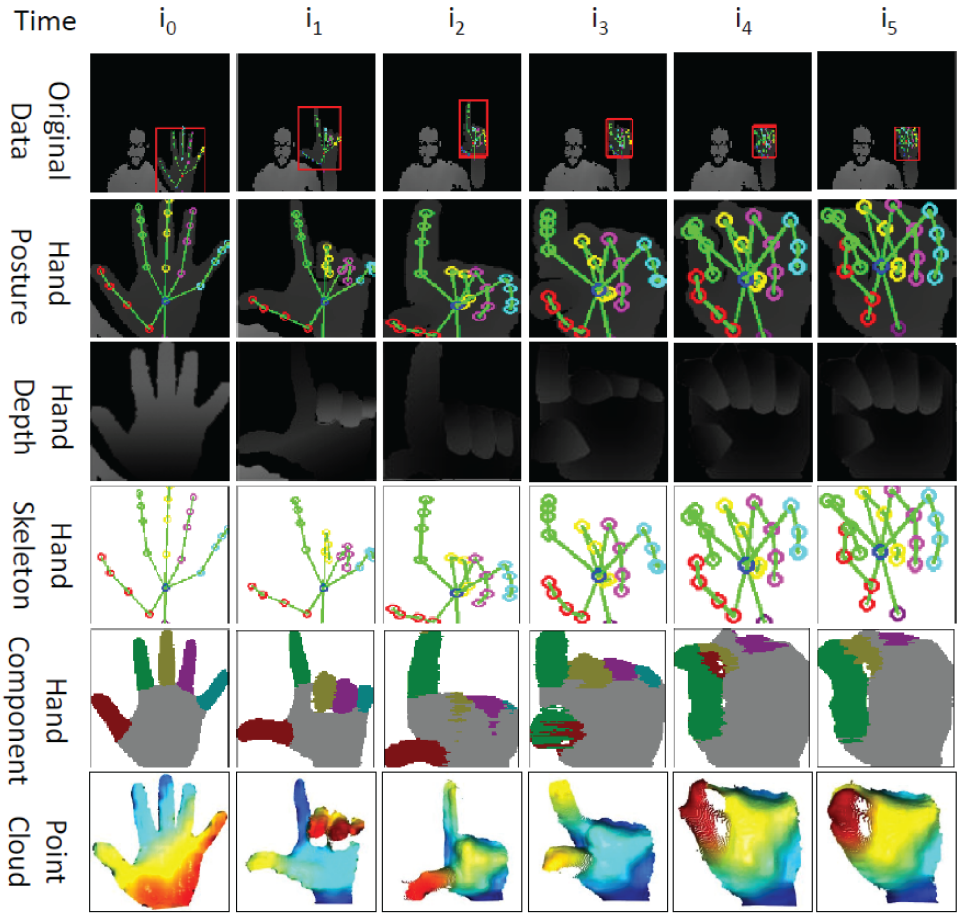


FIGURE 2.14: Overview of the features of a dynamic hand gesture. Left to right shows the time axis of the gesture, and top to bottom shows the types of hand data features, consisting of the original data, hand posture, hand depth, hand skeleton, hand component, and hand point-cloud. Image reproduced from [17]

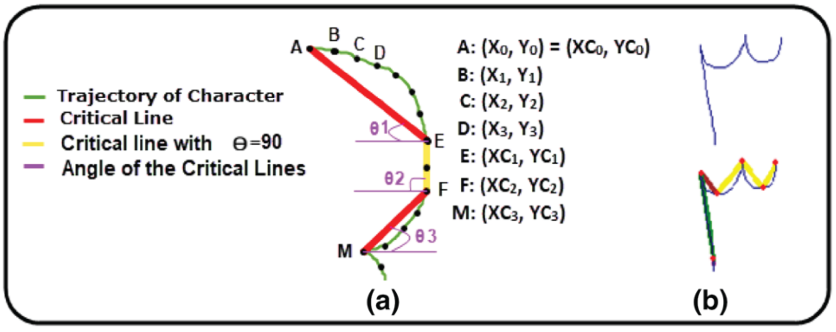


FIGURE 2.15: (a) Critical points on the trajectory, (b) critical points and lines in the Persian digit trajectory 3. Image reproduced from [18]

points belonging to the same gesture. Most of the work in hand gesture recognition consists of collecting RGB images, depth information, or hand gesture signal from hand-gloves. Each of the approaches has its advantages and disadvantages like, RGB images of hand gestures are rich in texture and color but does not contain depth-related information. The sensor-based approaches work well in occluded or different illumination conditions, but the noise and interference may reduce the recognition accuracy. Recently, researchers are using multiple input modalities to be fused for the same gesture collecting gestural images from either multiple input devices or same device with the capability of capturing multiple input streams (e.g. Depth-map, RGB, ultrasound). In [16], optical flow graph and color map information were combined to recognize dynamic hand gesture based on Jester [62], ChaLearn [63], and NVIDIA [13] datasets. A combination of RGB image, optical flow and depth information were used in [64] to recognize dynamic gesture of human upper bodies. Cues from RGB and depth modalities were proposed in [65] to extract spatio-temporal features such as HOG3D, motion boundary, dense trajectories, and gradient-based features. They have recognized in car dynamic gestures using an SVM classifier.

According to the different stages of fusion, multimodal data fusion can be divided into four stages, data level fusion [16], feature level fusion [64], score level [66], and decision level fusion [66, 67]. A sample data-level fusion is shown in Figure 2.16.

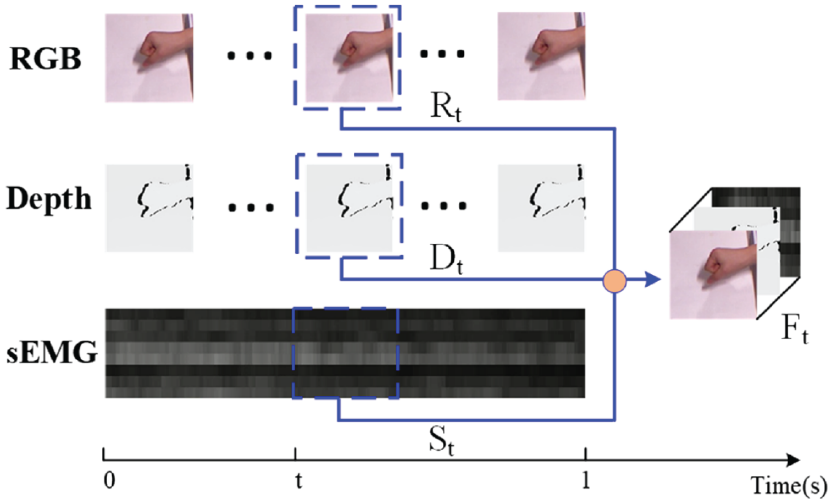


FIGURE 2.16: Data fusion process consisting of RGB, Depth, and surface EMG sensor data for Human-Robot Ineraction. Image reproduced from [19]

As mentioned in [19], among the different fusion techniques, data level fusion can achieve maximum fusion efficiency. Various information on the same hand gesture

can be combined or fused and the recognition accuracy of the hand gesture can be improved.

2.5.4 Classification models

Machine learning algorithms for classification tasks learn from data by different types of learning strategies like supervised, semi-supervised, reinforcement learning. In supervised learning, samples with labels are used for training and unknown samples are used for testing. The unsupervised learning tries to group or cluster the samples into desired number of clusters whereas, the semi-supervised learning is the mix-form of labeled and un-labeled data. Actually, depending on the hand gesture representation suitable machine learning algorithms or models are selected for classification tasks. In general, we can divide the classification models in to two parts, conventional models and deep-learning-based models.

2.5.4.1 Conventional models

Some commonly used machine learning models to recognize hand gestures are Support-Vector Machine (SVM), K-Nearest Neighbor (K-NN), Hidden Markov Model (HMM), Dynamic Time Warping (DTW), Fine State Machines (FSM), K-Means, Distance-based models and so on. These conventional models can be applied in both static and dynamic gesture recognition. However, the choice and applicability of these models depends on the feature representations and complexities, number of separable classes, types of gestures, application domain, and so on. In the literature, description of large varieties of classifiers can be found in [23, 25, 29, 45, 68]. Here, we will discuss briefly few of machine learning models used in gesture recognition.

Support Vector Machine (SVM) [69] is the non-linear supervised machine learning algorithm that uses a kernel function to map the lower dimensional space to higher dimensional space. The main idea is to identify the optimal separating hyperplane which maximizes the margin of the training data. It finds the optimal hyperplane with the help of support vectors that has the maximum margin. SVM is the most used classification model for human hand gesture and action recognition tasks [70, 71, 72, 73].

K-nearest neighbors (K-NN) is the distance-based non-parametric learning algorithm that measures the proximity distance within K-value. Several distance

measurement techniques are used like Euclidean distance, Manhattan distance, Minkowski distance etc. It computes the distances between the new instance to the training samples and then sorts the distances to find the k-nearest samples K-NN is an instance-based learning model. To recognize hand posture in different applications K-NN is used as reported in [74, 75, 76, 77].

K-means [78] is the unsupervised and well-known clustering algorithm in machine learning domain. K represents the number of clusters or cluster centroids usually chosen by the user. However, there are different ways to initialize the value of K. It is a distance-based technique that iteratively converges to a local minimum by updating the distances and group assignments of the numerical points in to a desired number of clusters. It minimizes within-cluster point scatter.

Hidden Markov Model (HMM) [79] is the probabilistic model that predicts the unobserved states from the observed sequential states. The HMM model is parameterized by a transition matrix, emission matrix, and an initial state probability distribution. HMM has been applied in varieties of recognition applications like speech recognition, optical character recognition, dynamic gesture recognition and so on. In case of dynamic hand gesture recognition, each frame in a sequence is the possible hand positions and transitions of the states mean the probability of a certain hand configuration moves to the next hand configuration [60, 80].

Dynamic Time Warping (DTW) [81] is the distance-based similarity matching algorithm that distance of two varying length time-series signals and was originally introduced for speech recognition. The DTW algorithm determines the warping path and DTW distances following few conditions like boundary condition, monotonicity condition, Continuity condition, Warping window condition, and slope condition. In [82], the authors use DTW algorithm to recognize static and dynamic hand gestures from time-series representation of hand finger contour information. There are other works on DTW-based classification tasks of hand gesture recognition reported in [83, 84]

In the literaturec we found that, most of the classification models used for static hand gesture recognition are SVM, K-NN, DTW and many of the research works have reported better recognition accuracy using the SVM model with different kernels. Whereas, most of the conventional dynamic gesture classification models are based on HMM and DTW due to the ability to predict time-series data and compute the likelihood of similarity.

2.5.4.2 Deep learning-based approach

There are various recognition approaches to recognize gestures intonated by hand movements including conventional machine learning-based models and deep learning-based methods. For both the static and dynamic hand gesture recognition tasks, deep learning-based approaches shown good recognition accuracies and robustness. Rather than handcrafted feature learning, automated feature learning through deep learning-based approaches has the ability to learn more relevant spatial and/or temporal features, has been studied extensively in the recent years [68].

CNN [85] is the convolutional neural network that takes input images of the hand gestures and works in three steps to extract features: convolution, activation, and pooling in a layer-by-layer architecture. CNN may use multiple layers making a deep layered model for feature extraction. Authors in [86] used CNN-based feature extraction module that performs image scaling to 32×32 and used ReLU as activation function. They used 2×2 max-polling layers and finally the classification module has 4×4×128 size as input to recognize Kinect sensor-based American Sign Language (ASL). Conventional 2D CNN can only extract two-dimensional spatial information. However, to extract temporal information in dynamic gestures which contains gestural information in a sequence of image frames or in a sequence of hand skeleton joint points, three approaches are used like 3D-CNN, two-stream networks, and Recurrent Neural Network (RNN)-based networks [37].

Compared to 2DCNN, 3DCNN works with multiple feature maps and are called deep 3D convolution network (3D ConvNets) [87]. Convolutional 3D (C3D) introduced in [87], is the first spatio-temporal feature extractor adopted in many dynamic hand gesture recognition tasks as reported in [37].

A two-stream CNN network was first introduced in [20] consist of RGB image frame and corresponding optical flow frames as two input stream of spatial and temporal information respectively. Figure 2.17 shows an example two-stream network architecture for spatio-temporal information classification.

Recurrent Neural Network (RNN) is a type of artificial neural network that are able to recognize and predicts sequence of data containing ordered information such as dynamic hand gestural movement in a video sequence, genome sequences, handwriting, spoken words or numerical time-series data. This deep neural network structure can hold memory data in hidden layers and can work on a sequence of arbitrary length.

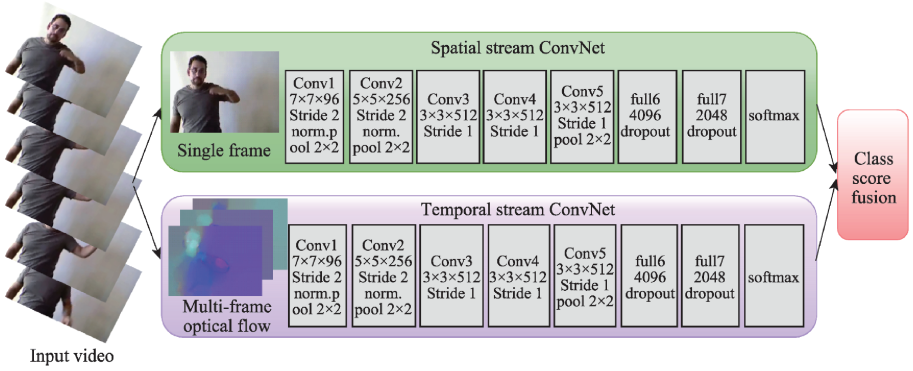


FIGURE 2.17: Two-Stream architecture for video classification. Image reproduced from [20]

A recurrent 3D convolutional neural network (R3DCNN)-based dynamic hand gesture recognition system proposed in [21] contains two-layered architecture. From the gestural frame sequences they have extracted local spatio-temporal features through a deep 3D-CNN layer, performed global temporal modeling using RNN-based layer, and finally calculated class-conditional probabilities using a softmax layer. Figure 2.18 shows their proposed architecture.

Long-Short Term Memory (LSTM) network is a special type of RNN capable of learning long-term dependencies in a gestural sequence. It handles the gradient disappearance and gradient explosion problems during RNN training [37]. A dynamic gesture recognition system based on CNN and LSM network is presented in [88] based on Leap Motion Controller (LMC) device. A combination of CNN followed by a LSTM-based network architecture was proposed in [22] to detect spatial patterns related to the hand finger skeleton joint points in 3D using CNN and spatiotemporal patterns using LSTM. Their architecture is shown in Figure 2.19. They have the same architecture for both human action recognition and dynamic hand gesture recognition.

2.6 Depth information in hand gesture recognition

Depth information is the distance value from the user to the depth camera (e.g. Microsoft Kinect, Intel Realsense, etc.). This information helps to generate depth image and used as skeleton features to different gesture recognition systems [89]. The depth image has a standard size, but for every pixel, it is known that how particular distances away the object are from the camera. 3D image is considered as depth image which has depth value. For those reasons, we can quickly calculate

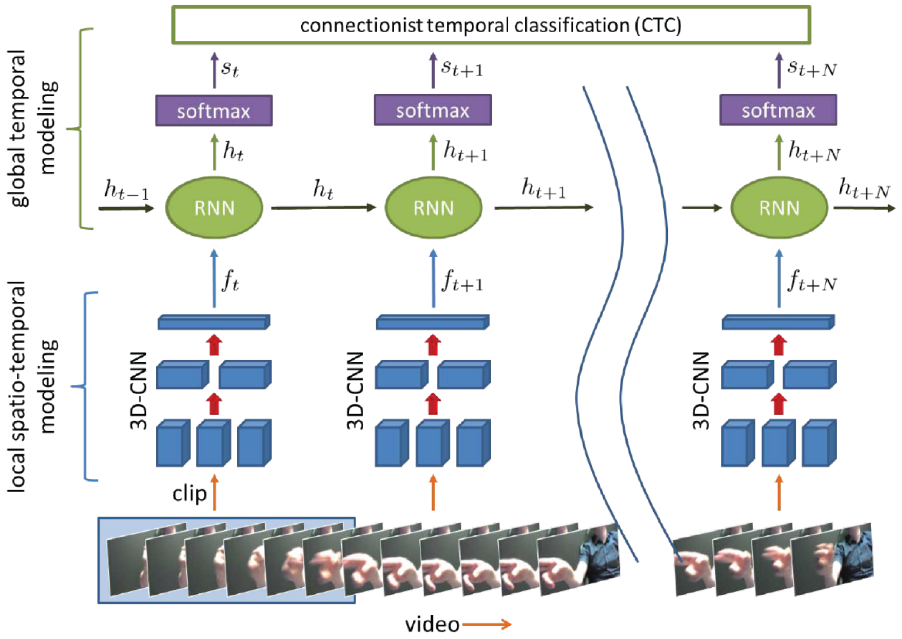


FIGURE 2.18: Classification of dynamic gestures with R3DCNN. A gesture video is presented in the form of short clips C_t to a 3D-CNN for extracting local spatial-temporal features, f_t. These features are input to a recurrent network, which aggregates transitions across several clips. The recurrent network has a hidden state h_{t-1}, which is computed from the previous clips. The updated hidden state for the current clip, h_t, is input into a softmax layer to estimate class-conditional probabilities, s_t of the various gestures. During training, CTC is used as the cost function. Image reproduced from [21]

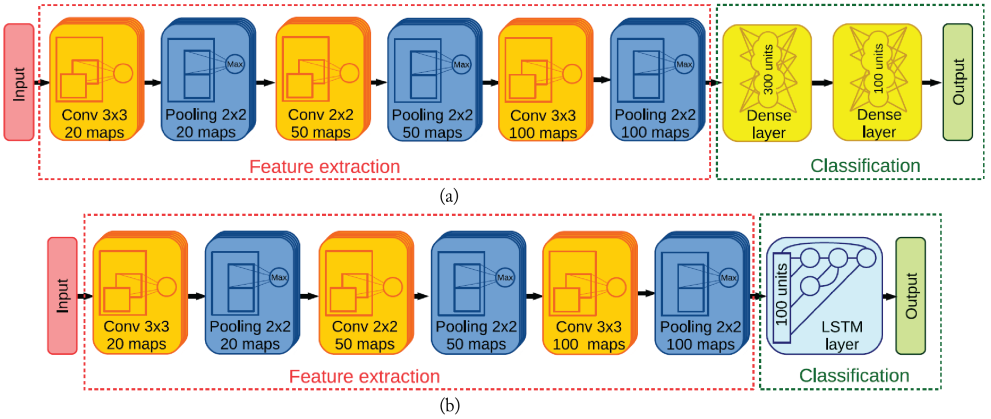


FIGURE 2.19: (a) The structure of the network during the pretraining stage consists of a CNN attached to a MLP and (b) The structure of the network during the final training stage consists of a CNN attached to a LSTM. Image reproduced from [22]

the length of an object. 3D reconstruction is the method through which shape and appearances of real objects are captured from a set of 2D images. It is widely used in fields such as computer vision, computer graphics, 3D reconstruction, and robotics.

Depth information provided by depth camera like Kinect, Intel RealSense and others contributes the gesture recognition research into three ways, in the hand segmentation process, in the depth-based feature representation techniques, and contributing as an input modality in a gesture recognition model. In the literature, we have found that, depth values or depth images contributed a lot in the recent computer vision-based hand gesture recognition approaches. A brief overview of different depth sensors is given in section 2.2. Microsoft Kinect camera provides 16-bit depth images (320×240 pixels) and 8-bit color images (640×480 pixels) of the same object. The value of each pixel in the depth image is the distance or depth value between the 3D world point and the sensor. Kinect version 1 gives 20 joint points while version 2 gives 25 joint points consisting of coordinate values with a range of detection from 0.5 to 4.5 meters. Intel RealSense camera gives 22 joint points of the hand skeleton structure. The joint points represent the 3D spatial positions of the body skeleton. Depth values are significant in segmentation because they are not affected by environmental factors like background color, different illumination conditions, cluttered objects, and also they faster the segmentation process with less error. Depth-based hand segmentation and localization of hand ROI are briefly written in section 2.5.1. The depth information characterize a particular gestural event effectively while doing the same using only RGB image will not give accurate results. For example, the "Reach out" hand gesture RGB image do not contain enough information changes in frame sequences. However, the corresponding depth pixel changes or variations of distance values carry important distinguishable characteristics. So, the use of right modality or combination of modalities in a gesture recognition system can give better recognition accuracy as mentioned in [37]. A detailed study on depth value utilization in recognition approaches can be found in [45].

Chapter 3

Recognition of static hand gestures using depth information

Symbolic gestures are the hand postures with some conventionalized meanings. They are static gestures that one can perform in a very complex environment containing variations in rotation and scale without using voice. The gestures may be produced in different illumination conditions or occluding background scenarios. Any hand gesture recognition system should find enough discriminative features, such as hand finger contextual information. However, in existing approaches, depth information of hand fingers that represents finger-shapes is utilized in limited capacity to extract discriminative features of fingers. Nevertheless, if we consider finger bending information (i.e., a finger that overlaps palm), extracted from depth map, and use them as local features, static gestures varying ever so slightly can become distinguishable.

In this chapter, we present our idea on how the depth-map information can be utilized to generate depth silhouettes with variation in contrast to achieve more discriminative keypoints. The approach, in turn, improved the recognition accuracy to recognize 10 numeric symbols (0-9). We will discuss how the Scale Invariant Feature Transform (SIFT) algorithm is used to produce robust feature descriptors out of those depth silhouettes. Our process of creating unified dimensional feature vector from a benchmark static hand gesture dataset and classification model used will be presented in section in 3.2. Then, the comparative result analysis among depth images, binary image, and images consisting the hand finger edge information generated from the same dataset are described in the experiment section in 3.3. Finally, we conclude the chapter summarizing the research achievements, few discussion points, and future scope in chapter 3.5.

3.1 Background study and related works

Gesture-based interaction has been introduced in many HCI applications which allow users to interact intuitively through computer interfaces in a natural way. Rather than using traditional unimodal inputs, blending alternative style of interactions, such as hand gestures along with mouse and keyboard introduces more degree of freedom (DoF) to the computer users. Nowadays, hand gesture-based interaction is a prominent area of research which has a huge impact in the design and development of many HCI applications like controlling robots through hand gestures, manipulating virtual objects in an augmented reality environment, playing virtual reality games through different hand movements, communicating through sign languages etc. We need these types of interaction to achieve interaction design goals like effectiveness, efficiency, affordance, feedback.

Hand gesture can be defined as the movement of hands and fingers in a particular orientation to convey some meaningful information [23] like pointing to some object through index fingers, expressing victory sign or OK sign, waving hands, grasping an object etc. Symbolic hand gestures represent some specific symbols like 'OK' sign or gesture that represents numeric symbol '1' (raising the index finger and bending all other fingers). In most of the cases, these gestural movement conveys single meaning in each culture having very specific and prescribed interpretations. More importantly, symbolic gestures are alternative to verbal discourse structure, different from everyday body movement which is consciously perceived. These gestures are observed in the spatial domain and are called static hand gestures characterized by the position of fingers (finger joint angle, orientation, finger bending information). Unlike static hand gestures, dynamic gestures are considered in the temporal domain, presenting gesture as a sequence of hand shapes which includes starting through ending hand pose (e.g. hand waving, boxing).

There are different approaches to capture and recognize these gestures. Computer vision-based approach impose restrictions on the gesturing environment, such as special lighting conditions, simple and uncluttered background, and occlusions (the gesturing hand is occluded by other parts of the body) [23]. Due to these restrictions segmentation of hand may cause the reduction in hand gesture recognition accuracy. Hand poses, generated in the process of gesticulation, can also be detected by means of wearable sensor like data-gloves. The data-gloves are embedded with the accelerometer, gyroscope, bend sensor, proximity sensor, and

other forms of inertial sensors [90]. These sensors collect hand-finger motion information as multi-parametric values. However, the sensor-based gesture recognition approaches have limitations in terms of naturalness, cost, user comfort, portability, and data preprocessing.

The recent advancements in stereo vision camera that utilizes depth perception from smaller to larger distances have opened a huge scope for the researchers to work with depth information [91]. Traditional web cameras do not provide the depth values (the distance of the gesturing hand from the camera). Depth information can help eliminating occlusion problems easily, can faster the segmentation process with less error. In an occluded background, using depth information it is possible to extract the gesturing hand movement information including other important features (e.g. finger bending information) which can be effectively utilized in feature representations. Moreover, static gesture can be performed by the users with varying hand size, changes in hand position (orientation, rotation), different illumination conditions. Scale Invariant Feature Transform (SIFT) [1] is an algorithm that works better for these types of variation. The algorithm generates key points from images and provides 128-dimensional feature vectors.

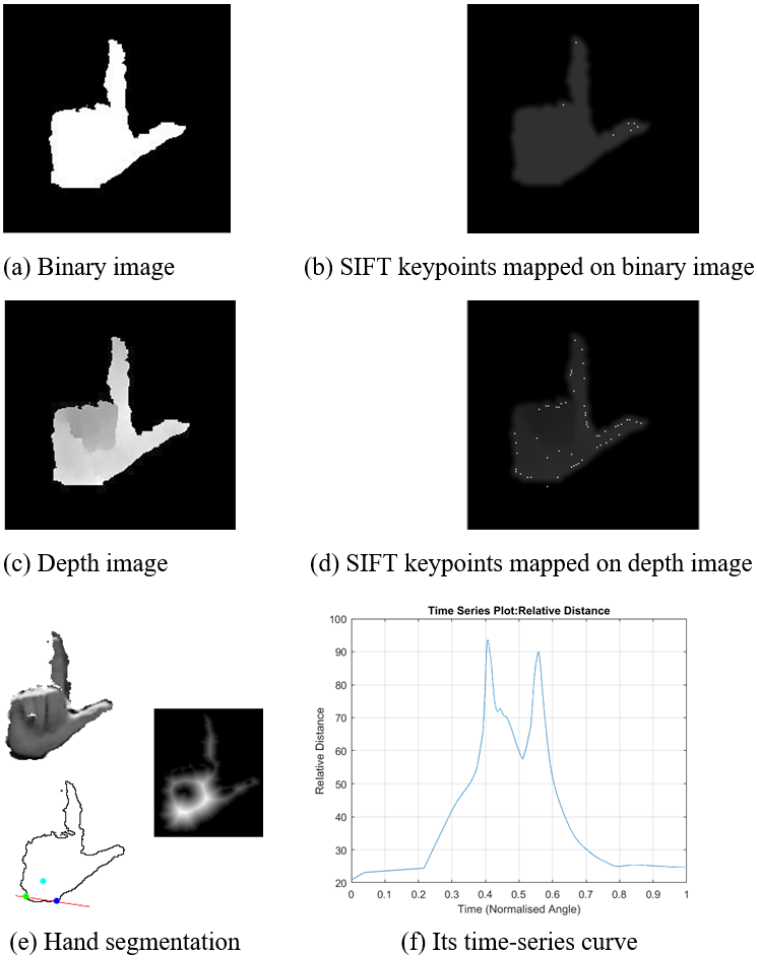
In this research work, we try to recognize symbolic hand gestures representing 10 numeric symbols from 0 to 9. These are very close gestures, differing only in slight variations (e.g. the difference between numeric symbol 2 and numeric symbol 3 is due to the presence/absence of one finger only) of finger positions. With the help of depth data stream, after a quick and robust segmentation process, we have calculated depth threshold based on which the contrast varying depth images are generated according to the depth map of the individual gesture. This process was applied to 100 image instances per gesture. In each image for the same gesture, we got the different number of SIFT keypoints. By combining the keypoints, we have generated bag-of-feature (BoF) vector with the help of the k-means clustering technique to generate uniform dimensional feature vectors and classified using a multiclass SVM.

From the depth sensors, the most common features used in hand posture recognition [41] are skeleton joint positions, hand geometry, hand finger shape, area, distance features, depth pixel values, etc. Generally, these features can be categorized as local features or global features. The major challenges of these feature descriptors are variations of gesturing hands while articulating an emblem or symbolic gesture. A gesture may slightly differ in terms of hand shape and size, variations in translation or rotation of the fingers for the same gesture. A robust

hand gesture recognition system should be invariant to the scale, speed, and the orientation of the gesture performed.

The approaches that are followed by static gesture recognition system from binary images in [92] and time-series curves in [2], do not facilitate the possibility of extracting local finger context information. The authors in [92], have captured RGB images from webcam, converted them to binary images and applied SIFT algorithm to determine the recognition accuracy. In binary images, the finger context information - shape, orientation, bending fingers, occlusion cannot be preserved - a limitation that can be overcome by utilizing depth map information of the gesturing hand. SIFT Keypoints are important feature points which are well distributed and contain information about not only thumb and baby fingers but also about finger bending information of index, middle, and ring fingers. Figure 3.1, shows the differences of SIFT keypoints in gesture 8 mapped in to the binary image (7 key points) (Figure 3.1(b)) and in to depth image (56 key points) (Figure 3.1(d)). This information is not present in the case of binary image or time-series curve. SIFT works on local oriented features rather than topological shapes of opening fingers which are considered as the global features. In [2], global features are used to generate time-series curves (Figure 3.1(f)) after the segmentation process as shown in Figure 3.1(e) from the hand shape represented in binary image. The edit-distances are calculated to apply distance-based matching algorithm, such as Finger-Earth Mover's Distance (FEMD). Edit-distance-based matching algorithms are not completely rotation, orientation invariant because they are measured by comparing time series trajectories based on the proximity distance and not based on the local shape information. Moreover, the temporal information are better for dynamic gesture recognition rather than static gesture recognition [93].

Local features measure the characteristics of a particularly important region of the object, superior in discriminating fine details. In [94], shape descriptor-based algorithm and weak learning-based strong classifier were applied to recognize three symbolic gestures (palm, fist, six). Their goal was to get orientation invariant property of those gestures. They have used SIFT features as local features in weak classifier for hand detection and trained each classifier independently. The accuracy, in this case, depends on the large set of training images which they have not considered. They have used a varying number of training images for individual gestures. They have not considered the fact that, SIFT features extracted from the different gesturing image can form a natural group of clusters having feature vectors of the unified dimensions appropriate to feed into a classifier that can



(a) Binary image (b) SIFT keypoints mapped on binary image

(c) Depth image (d) SIFT keypoints mapped on depth image

(e) Hand segmentation (f) Its time-series curve

FIGURE 3.1: Differences in the number of SIFT keypoints in both (a) Binary image and (c) Depth image and the use of finger bending information

recognize more than two classes. We have achieved this by clustering feature descriptors and generating BoF features. In [95, 96], the researchers have considered Haar-like features, applied learning-based techniques to recognize hand gestures. They required a huge number of images for training and testing with high computational power and they have not considered the scale-invariant property for object detection.

Global features measure the characteristics of the whole image and face difficulties in capturing fine details. An example would be the contour representation of a hand gesture image (e.g. the hand contour image of Figure 3.1(e)) which gives hand finger shape information from the whole image. The limitations of contour-based recognition methods are that they are not robust on local distortion, occlusion, and clutter [97]. To extract the complete hand posture information while a finger and a palm are overlapped, such as bending fingers, as shown in

Figure 3.1(d), the consideration of hand contour as the global feature representations is not enough. The similar problems are also mentioned in the recognition approaches like skeleton-based recognition methods [98], shape contexts based methods [99] and inner-distance methods [100]. A solution to these problems was proposed using a novel distance-based measurement technique called Finger Earth Mover's Distance (FEMD) [2]. They represented the shape of hand fingers as a global feature (the finger cluster) by analyzing time-series curve. In the curve, the Euclidean distance between each contour point and the center point is considered in one dimension and the angle of these contour points made with the initial point relative to the center point is considered as another dimension. Figure 3.1(f), shows the time-series curve of the topological hand shape considered as finger parts and matches those fingers only, not the whole hand shape. Features only from opening finger parts may not give good recognition results. Rather features including bending finger parts as local features will play a significant role to improve the recognition accuracy. We have considered those features in our proposed approach. Moreover, for gesture recognition, they [2] have applied template matching with minimum dissimilarity distance which may not give improved recognition accuracy on both changes in orientation and rotation of a particular pose. We propose to overcome this problem using local features found as SIFT keypoints. Edit-distance-based time series matching approaches are more applicable for dynamic gesture recognition due to their spatio-temporal features, rather than static symbolic gesture recognition. Template-based approaches are good to recognize the shape as a whole but lack in terms of invariance. SIFT algorithm is known to be robust for its distinctiveness and invariance to rotation, scale, and translation in object recognition. Depth image acquired using Kinect depth sensor suffers from low grey level contrast that can cause an unstable set of keypoints. Recently in [101], the researchers used Kinect-based depth map information to discard the SIFT keypoints that are located at the boundaries of an object. They applied Canny's edge detection algorithm [102] on depth images and generated an object model to store depth values and distance to the nearest depth edge for the remaining SIFT keypoints. They have used Euclidean distance based nearest neighbor algorithm to rank the keypoints matches and performed RANSAC-based homograph estimation for object pose estimation. Their aim was to identify predefined objects in the surrounding environment for the visually impaired. To extract a stable set of SIFT keypoints different techniques were proposed by the researchers. Preprocessing on the medical image (retina image) was done to reduce the number of SIFT keypoints in [103].

In [104], the researchers have extracted the SIFT keypoints from both the color and the depth image and tried to find out the correspondence of SIFT keypoints between those two images. They have combined SIFT descriptor with Harris corner detector to compute SIFT features at predefined spatial scales. They enhanced the depth image contrast by applying histogram equalization without utilizing the depth values explicitly of the gesturing hand to generate contrast varying depth images. However, we have considered the depth map information to determine the contrast level and generate depth silhouettes accordingly.

SIFT algorithm along with its different variants like PCA-SIFT [105], SURF [106], GLOH [107] has been applied in various applications such as, image stitching, object recognition, image retrieval etc. SIFT and SURF algorithm was also applied in simultaneous localization and mapping (SLAM) with RGB-D Kinect sensor on robots [24]. SURF is the fast approximation of SIFT that uses box filter instead of Gaussian filter. However, SURF is not good at different illumination conditions[106]. To improve the time complexity of SIFT several alternatives were proposed, such as Binary Robust Independent Elementary Features (BRIEF) [108], Oriented FAST and Rotated BRIEF (ORB) [109] that uses binary descriptor instead of floating point descriptor to achieve faster performance suitable for real-time applications.

In [110], the authors showed the comparisons among different image matching algorithms, such as SIFT, SURF, and ORB. They have manually performed transformation and deformation on the images in respect to rotation, scaling, fish eye distortion, noise, and shearing. The comparison was done based on different evaluation parameters, such as the number of keypoints in images, execution time, matching rate. For most of the scenarios they have found SIFT performed best. The researchers in [111], tried to use depth map to perform smoothing process in the scale-space. They smoothed the scene surface considering smoothing quantity as a function of the distance given by the depth map so that 'the further a given pixel is, the less it is smoothed'. They tried to inject the smoothing filter in the SIFT algorithm and determined the repeatability score to evaluate the keypoint detection performance. Their goal was to find the keypoint repeatability under viewpoint position changes. However, the dataset we have used in our research was generated using single depth camera without changing the viewpoint positions.

Bag-of-Feature (BoF) representation was used in [112] to obtain a global information of visual data out of arrays of local point descriptors generated by SIFT algorithm. SIFT algorithm can extract higher dimensional feature points from

the images even with lower resolutions but compromises the efficiency in terms of computation. To address this problem, BoF approach has been applied in reducing feature dimensions, redundancy elimination, and to extract global information from local SIFT features [112]. Moreover, the BoF approach has been considered as an efficient method to represent visual contents in hand gesture recognition [113]. The local feature points extracted from SIFT are fed into clustering algorithm to learn visual codebook and then each feature vector is mapped to a visual codeword represented by a sparse histogram. We have applied this technique to depth images for the classification using a multi-class SVM.

3.2 Methodology

Our proposed methodology of symbolic gesture recognition system consists of different steps like, (1) Hand segmentation and depth silhouette generation, (2) SIFT keypoints extraction, (3) Clustering keypoints and generating BoF descriptors, (4) Symbolic gesture recognition using SVM.

The architectural diagram of the proposed approach is shown in Figure 3.2. The standard dataset [2] has considered 640×480 image resolution to capture the RGB image and the depth map of gesturing hand using Microsoft Kinect. Depth values are stored in millimeters. After calibration, we have applied the segmentation process as described in [2], except generating grey-scale variations on depth images.

3.2.1 Hand Segmentation

Segmentation is the process of removing the non-interesting area from the pertinent object. Many of the techniques in hand region segmentation worked on color space-based detection like skin-color detection, YCbCr/HSV color space filtering and so on. These color-based techniques have limitations due to the noise, lighting variations, background complexities. However, utilizing depth map information combined with color information improves the segmentation process which in turns gives better recognition accuracy.

Before segmenting the hand shape or region of interest (ROI), some pre-processing is performed. This involves calibrating the RGB and Depth Images. The RGB image is also converted into grey-scale. To extract the region of interest, first, we locate the smallest depth value from the depth image. This corresponds to the

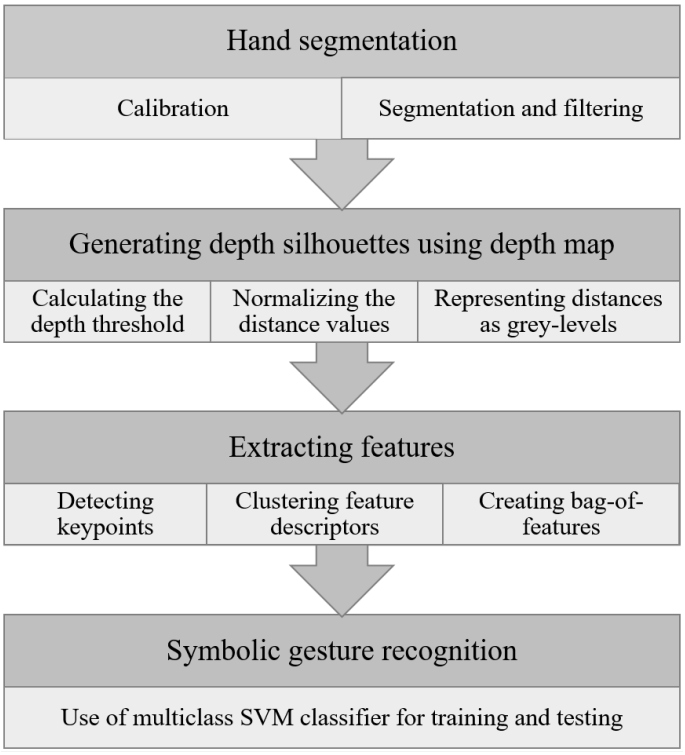


FIGURE 3.2: The architecture to recognize symbolic gestures

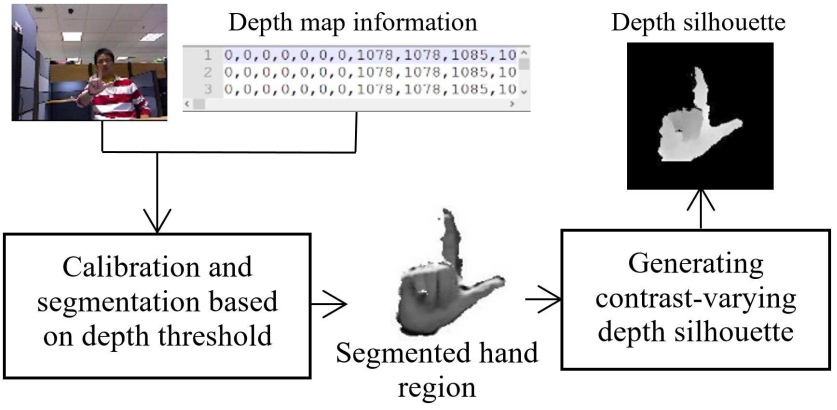


FIGURE 3.3: Hand Gesture Segmentation

closest point of the hand from the camera plane. We call this value minimum-distance. Next, an empirical threshold value is added to the minimum-distance to give the segmentation threshold. This segmentation threshold is then used to segment the hand region from the rest of the image. This approach has proven to be robust in cluttered and noisy environments [114]. It is important to note that the hand should be the closest object to the camera for proper segmentation. The segmentation threshold is the sum of a minimum distance and a depth threshold.

The minimum distance is easily obtained from the depth image as the minimum value in the depth matrix. The depth threshold is estimated based on different possible orientations of the hand shape.

After multiple measurements and testing, an upper bound is chosen as the depth threshold, such that, the sum of the depth threshold and the minimum distance will allow us to isolate or segment the hand shape including the black belt from the rest of the image. In our scenario, the depth threshold was estimated at 200 mm. The depth threshold is useful for filtering cluttered background containing an overlapped image (e.g. gesturing hand is overlapped with the face having the same color). We followed the same segmentation process as described in our previous work in [115]. However, in this research, the segmentation process is applied to a larger and challenging dataset [2]. Earlier, we used smaller dataset containing only 5 (five) static hand gestures representing numeric symbols 1 to 5 in a restricted environment, collected from a limited number of users.

3.2.1.1 Generating depth silhouettes using depth map

The images from the Kinect depth stream are in 640×480 resolution which does not show enough contrast variations. Keypoints with low contrast will not give enough gradient variations to identify finger bending information. If we can generate contrast variation according to the depth values, then we can get more discriminative keypoints. These keypoints would be the salient features to improve the recognition accuracy. So, we have done some preprocessing where the depth values of gesturing hands were used to produce grey-scale levels. The closer a point is, the brighter is its shade. To do that, we cropped out depth values of the hands and got an m-by-n matrix with depth values of hands and its background.

Let $dist(x, y)$ is the distance of a point in the millimeter at (x, y). $f(x, y)$ is the corresponding grey level of the generated image used in extracting the key features by SIFT. Now, we select η as the number of grey levels between $greyLevel_{min}$ and $greyLevel_{max}$. We also selected η number of distance segments between $dist_{min}$ (minimum distance) and $(dist_{min}+dist_{th})$; where $dist_{th}$ is the distance we assumed the hand would be from $dist_{min}$, the depth threshold. We let the background be black in the generated image to get the better result using SIFT. We have applied

(3.1) to generate the grey-scale image using only the depth values.

$$f(x,y) = \begin{cases} 0, & \text{if } dist(x,y) > dsit_{min} + dist_{th} \\ greyLevel_{min} + (\lfloor(\frac{dist(x,y)-dist_{min}}{dist_{th}-dist_{min}} \times \eta) + 0.5\rfloor & \\ \times \lfloor \frac{greyLevel_{max}-greyLevel_{min}}{\eta} \rfloor), & \text{Otherwise} \end{cases} \tag{3.1}$$

We can see from the equation that any point in the depth image within the threshold distance is going to be a non-black pixel depending on the grey-levels determined from depth information. To assign grey-levels to those pixels we segmented the depth values in η levels. Any Distance value under the threshold is rounded off and normalized. The normalized distance values are converted into appropriate grey-levels. After that, we find a grey-scale image which is the depth silhouette of a hand with the dark background and the grey-levels corresponding to the depths of different parts of the subject hand. To emphasize on the contrasts, η number of segments were used. If we had used all the 256 levels of the grey image, the contrasts would not be prominent enough to get fair results. We considered $\eta = 10$ grey-scale levels from 155 to 255, dividing the levels equally to get a good contrast ratio. The number of levels was heuristically determined based on the assumption that more levels of grey will mean that the hand segments' contrast will be low. Thus, one of our main focuses (to represent distances in distinctive grey levels) would be undermined. Representing the distances using fewer grey-levels would have the similar effect as the binary images. The shape would be distinct but the local features would be lost. Moreover, the grey-scale images with proper contrast are useful enough to distinguish the curves and angles of finger joints in different gestures. Both of the characteristics helped the SIFT to generate feature descriptors for the gestures, indifferent of the orientation of the hands.

For each gesturing image, we have extracted depth values within 200 from the depth image of the resolution 640×480. Actually, the 200 region contains the gesture information which we have used to generate the depth silhouettes. The process of segmentation and grey-scale varying depth silhouette generation are shown in Figure 3.3.

3.2.2 Feature Extraction

Features to be extracted by the feature extraction algorithm should present a high degree of invariance to scaling, translation, and rotation. Feature representation

depends on the algorithm to be used for classification. We have used SIFT algorithm to represent the features as 128-dimensional feature points that are extracted from the depth images.

3.2.2.1 SIFT features

The SIFT algorithm detects keypoints from a multi-scale image representation consisting of blurred images at different scale. The keypoint location and the scale values of each keypoint are accurately determined using the Difference of Gaussian (DoG). Then the key points are filtered by eliminating edge points and low contrast points. After that, the orientation of the keypoint is determined based on the local image gradient within an image patch. Finally, The keypoint descriptor is computed which defines the center, size, and orientation of normalized patch [1]. We have used the SIFT implementation code as in [116].

Features generated by SIFT algorithm are invariant to scales and robust against changing position of object, slight rotation of object and object in noisy and varying illumination condition in different images. These feature points can be found in the high-contrast regions and we have generated those contrast varying images based on depth values of the gesturing hand. SIFT algorithm effectively determines the keypoints on those depth images and represent them as feature descriptors.

The main objective of our approach is to improve the recognition accuracy for static gestures using depth information compared to binary and time-series representation of the images. We have utilized depth information and generated depth silhouettes which can be fed to any keypoint detector and descriptor-based algorithm, such as SIFT, SURF, ORB etc. However, we have chosen SIFT to generate training and testing images. The training images with corresponding keypoints mapped over the gesturing image is shown in Figure 3.4. The first and third columns in Figure 3.4 represent the depth silhouette generated using depth map information of the gestures 1-10 (G1-G10) of the numeric symbols 0-9. The second and fourth columns in Figure 3.4 represent the corresponding hand gestures G1-G10 with 27, 41, 51, 61, 77, 101, 55, 56, 32, and 80 SIFT keypoints respectively.

While extracting the keypoints we have found that, the number of keypoints varies according to the type of gestures. As different symbolic gestures consist of a different number of fingers to be articulated, hence we got these variations. We

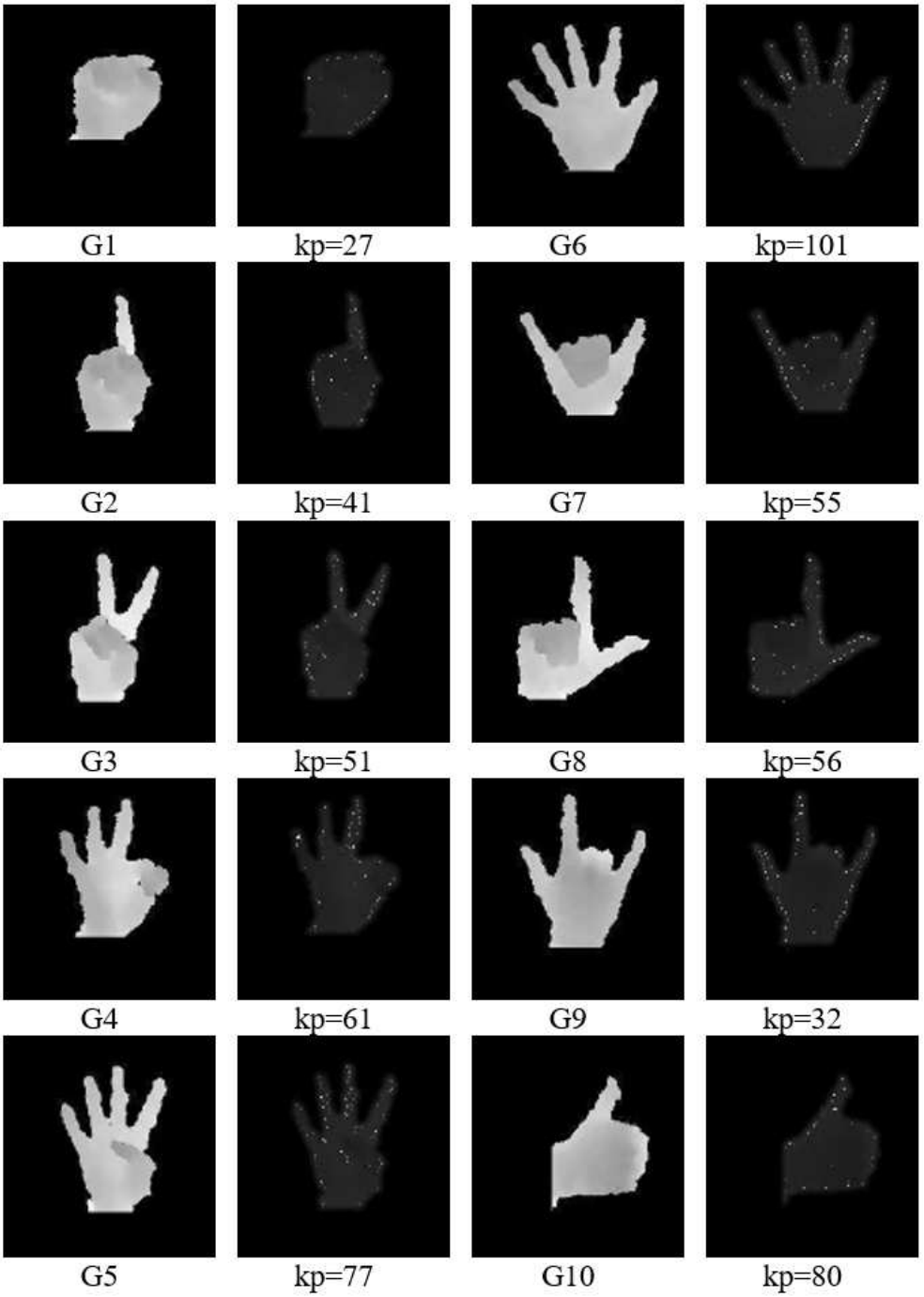


FIGURE 3.4: Example images containing generated depth silhouettes (first and third columns) and the corresponding SIFT keypoints mapped in to depth images (second and fourth columns) showing numeric symbols (0-9) representing the gestures (G1-G10)

captured 100 images per gesture as the candidates to generate keypoint descriptors and we got 41273 keypoints by considering 1000 images in total training images. The distribution of the number of keypoints per gesture is shown in Figure 3.5.

The keypoint descriptors that we have found are 128-dimensional feature vectors.

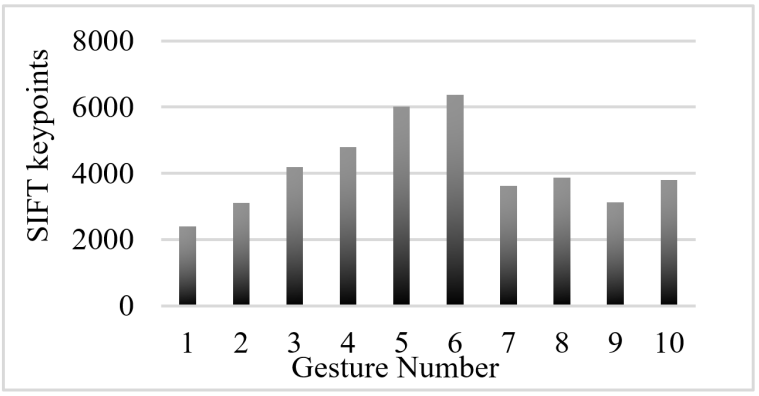


FIGURE 3.5: Number of SIFT keypoints at $\sigma = 1.8$

Due to the changes in orientation, scale, illumination of the same gesturing image by multiple persons the number of keypoints varies. Moreover, the dimensions of the gesturing images become larger which increases computations. Hence, we have used the strategy of a bag-of-visual-words and clustering technique to reduce dimensions.

3.2.2.2 Clustering feature descriptors

The dimension of the feature vector in each gesturing image varies based on the number of keypoints found for each gesture. The problem is, we need unified dimensional feature vectors as the training set to classify using multiclass SVM [117]. The depth image that has 27 keypoints, the dimension of that image becomes $27 \times 128 = 3456$ and if another image from the same gesture contains 80 keypoints then the dimension becomes $80 \times 128 = 10240$. So, we have used the bag-of-word for which we need clustering to reduce the dimensions. The basic k-means clustering served our purpose because k-means converge faster than hierarchical-based clustering approaches. It also gives efficient performance for larger datasets. The keypoint distributions for different gestures are found to be almost Gaussian and distinctive as shown in Figure 3.5. In the concept of bag-of-word, the clusters are defined as codebooks and the size of the cluster determines the convergence property of the clustering technique. If we took smaller codebook size then, bag-of-word vectors may not contain all the important keypoints. The larger codebook size may raise the overfitting problem. As the keypoints in depth images are well distributed containing information about opening finger parts as well as bending finger parts, intuitively, we should get better accuracy.

To build our k-means clustering model, we have chosen 1600 as the cluster size which is the size of the visual vocabulary. An individual feature vector is assigned based on the nearest mean value while partitioning the feature vectors. After that, the codevectors were updated to reform the clusters until the grouping stops.

The goal of the k-means clustering approach is to minimize total intra-cluster distance using (3.2).

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2 \tag{3.2}$$

Where k is the cluster size, n is the number of instances, c is the cluster centroid of cluster j. An illustration of k-means clustering is shown in Figure 3.6 for five keypoints: A, B, C, D, and E to form two clusters.

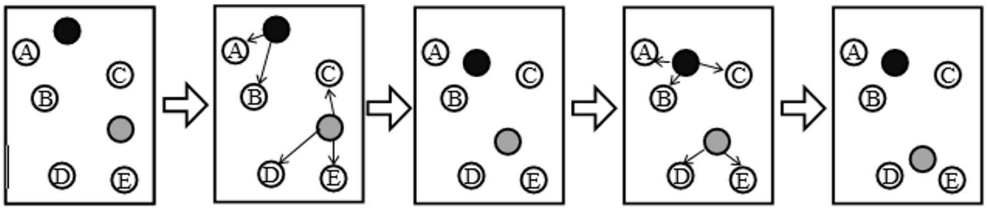


FIGURE 3.6: Demonstration of k-means clustering

We develop the cluster model from each of the training images consisting feature vectors and encoded each of the keypoints with the clustered index. Keypoint and the cluster centroid are mapped according to the minimum distance criteria based on Euclidean distance measurement.

We got k disjoint subgroups of keypoints after assigning the keypoints to the corresponding cluster centers. So, the dimension of each training image consisting n keypoints ($n \times 128$) reduced to $1 \times k$. k determines the cluster numbers.

3.2.2.3 Creating bag-of-features

We have created the bag-of-feature representation of each training image from the SIFT feature extracted. In order to learn visual vocabulary, we have built the k-means clustering model. Keypoints from each training image is mapped to the centroid of the corresponding cluster to represent visual vocabulary - is known as feature vector quantization (VQ) process [118]. After that, we have represented each training image by the frequencies of visual words and found a unified dimensional histogram vector. The histogram representations of images

of each gesture are ready for the classification. The process of creating Bag-of-features is shown in Figure 3.7.

We updated the feature extraction process which is applied to two types of images, one is the depth image and the other one is the edge image, generated from the same dataset [2]. This is because we tried to establish more reliability in our approach through experimental evaluation compared to our previous work [115].

3.2.3 Recognition of gestures using SVM classifier

The bag-of-feature vectors are now the input feature vectors for the classification algorithm. In order to recognize the performed symbolic gestures, we have applied a multiclass SVM training algorithm which is a supervised machine learning algorithm. It performs non-linear mapping and transforms the training dataset into higher dimensional datasets. The algorithm tries to find out an optimal hyperplane which is linear.

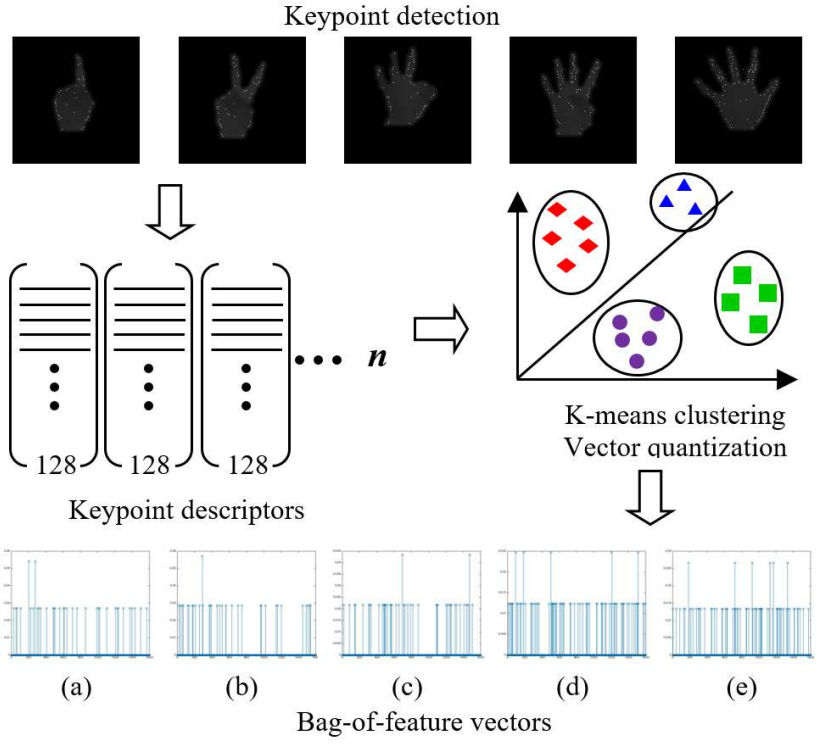


FIGURE 3.7: Generating bag-of-feature for training. (a)-(e): Bag-of-feature generated of gesture 2-6 from individual depth silhouette for 1600 clusters.

SVM determines the support vectors those are closest to the separating hyperplane. The margins are also defined by those support vectors. Maximum separation is ensured by the maximum margin hyperplane.

We have applied the one-against-all approach to implement the SVM classifier [69] that built the model in respect to the training set supplied with group vector (class label indicator from gesture class 1 to 10).

3.3 Experimental results

In order to evaluate the symbolic gesture recognition results, we have considered NTU hand gesture recognition dataset [2] which is a benchmark dataset in static hand gesture recognition. The dataset was collected using Kinect depth camera from 10 subjects. Each subject has performed 10 symbolic gestures 10 times. So, the dataset contains total 1000 instances. Each gesturing instance contains a color image and the corresponding depth map. The dataset was prepared in a very challenging real-life environment containing the situations like the cluttered background, pose variations in terms of rotation, scale, orientation, articulation, changing illumination, etc.

We have conducted the 5-fold cross-validation process to evaluate our results. In each fold 4 of the image groups were used as training set and one of them were used as validation testing set. Each fold contains 20 images and we permuted the process, calculating the accuracy of SVM classifier. All the experiments were executed on an Intel Core I7 2.60 GHz CPU having 16 GB RAM.

Our system is robust to cluttered background due to the process of segmentation where the depth threshold and minimum hand finger distance from the depth camera are used to determine the segmentation threshold. Good contrast varying depth silhouettes guarantees SIFT keypoints to be extracted in different scale-rotation-orientation changing conditions as shown in Figure 3.8.

SIFT extracted local features which produce good recognition results compared to global features considered in FEMD based approach [2]. We tested our results in two types of images produced from the same dataset. Binary images and image with edge information. The former was generated along with depth silhouette by converting the depth silhouettes into binary images and the latter was generated applying Canny's edge detection algorithm [102] on depth silhouettes. Example binary and edge images are shown in Figure 3.9. The image contains internal finger bending edge overlapped with palm and the external hand shape edge, but this information is not present in binary image or time series images. So, the accuracy of our approach should vary on these different datasets.

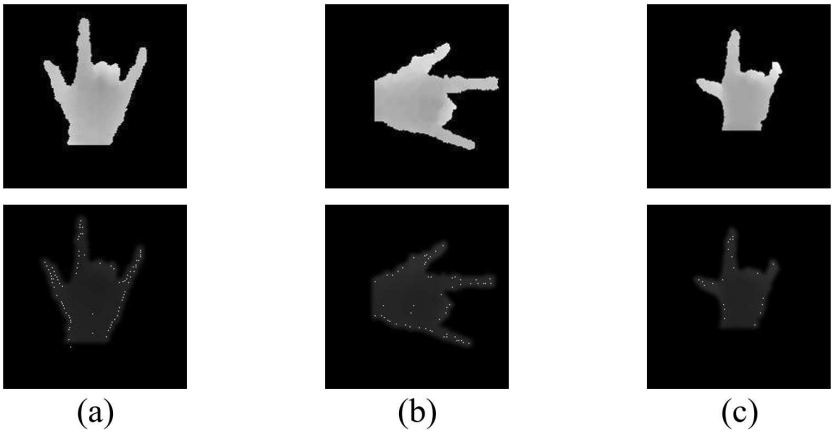


FIGURE 3.8: SIFT features are robust to orientation changes (b) and scale changes (c) along with normal pose (a).

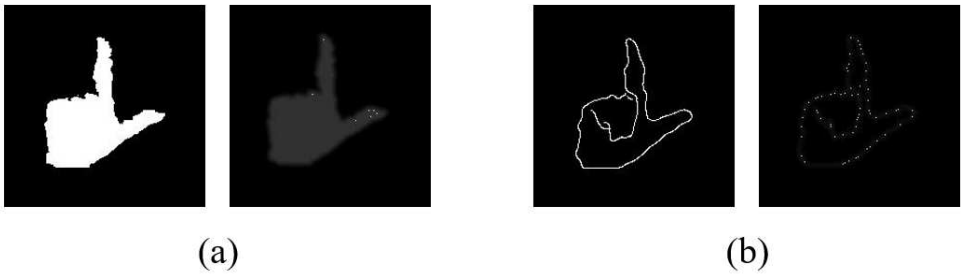


FIGURE 3.9: SIFT keypoints on binary image (a) and edge image (b).

Previously in [115], we demonstrated that the SIFT works better on depth images rather than binary images for static hand gesture recognition consisting of symbolic gestures (numeric symbol 1-5). The dataset used in the previous work was generated by ourselves in a constrained environment. To create the dataset, we considered a limited number of hand gestures from a limited number of users. The comparison of experimental results was not performed among depth images, binary images, and edge images. However, in this research work, we have compared our experimental results among all the images and also compared the result with FEMD-based approach [2], got higher accuracy for depth images (recognition accuracy is shown in Figure 3.10.). Moreover, we elaborated the processes of depth silhouette generation with equations which illustrates the fact that, the intensity of a pixel in grey-scale depends on the distance of that pixel from the depth camera. This, in turn, determines the contrast of the image based on depth values suitable for key point detector and descriptor-based algorithms.

To evaluate the accuracy of our approach, we generated different SIFT keypoints by varying the sigma (scaling parameter) value and found the highest accuracy at

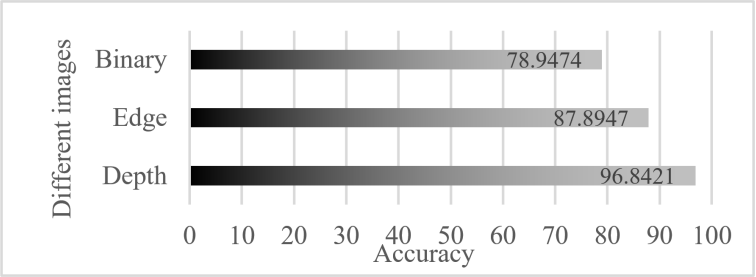


FIGURE 3.10: Accuracy comparison among different images.

$\sigma = 1.8$. The mean accuracy at different σ values is shown in Figure 3.11.

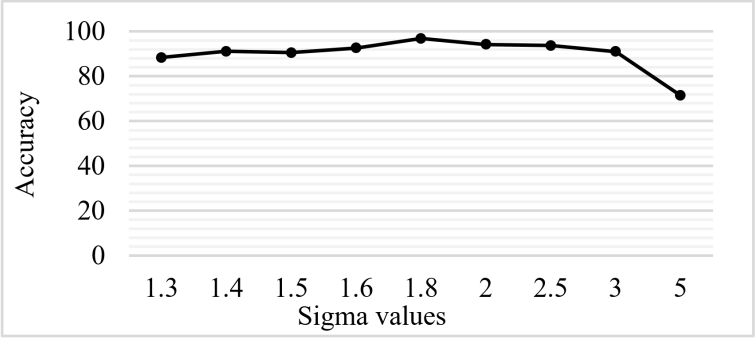


FIGURE 3.11: Accuracy at different sigma values.

With the increased value of sigma, we found more keypoints (Figure 3.12) which results in spurious DoG extrema considered as less stable and not linked to any particular structure in the image. These cause the differences in accuracy.

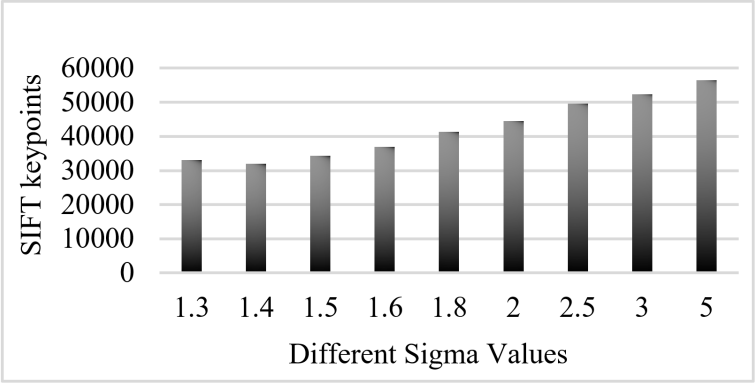


FIGURE 3.12: Number of SIFT keypoints at different Sigma values.

We evaluated the accuracy with the different number of clusters. We considered 100, 200, 400, 800, 1200, 1600, and 2000 clusters to validate our proposed method and compared the results for depth, binary, and edge images. The comparison

result is shown in Figure 3.13. We observed that, the accuracy increments com-

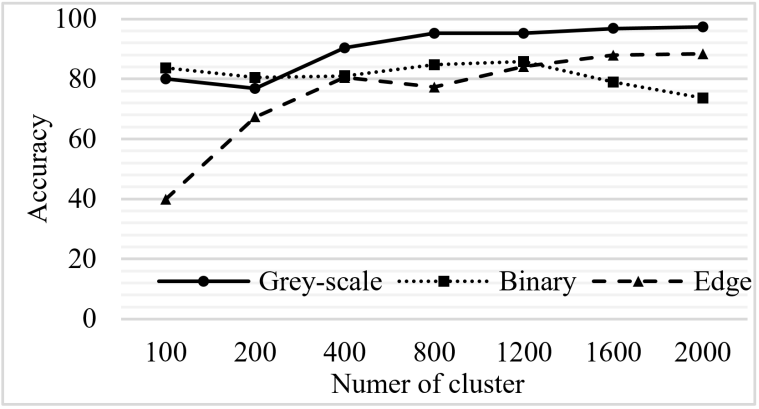


FIGURE 3.13: Overall accuracy comparison among different images.

mensurate with the higher number of clusters. The highest accuracy we attained have been with a cluster size of 1600. This phenomenon can be traced back to depth images which significantly contributes to the salient keypoints identification for it is the depth images from which we can distinguish the positions of each fingers. However, the same cannot be said for binary images or images containing only edge information. FEMD has considered the shape distance metric which matches only opening finger parts or finger shapes, not the whole hand. While making a pose the bending finger parts are also important to distinguish slightly varying gestures, which can be found in the local features. To avoid local distortion we have chosen the correct scale factor. We have presented the input hand as a contrast varying grey-scale image depending on the depth map information but FEMD has presented the hand image as a global feature using time-series curves. Shape contour presentation introduces lower accuracy in terms of scale, rotation or orientation changes which we have overcome through depth images and got accuracy up to 96.8421% whereas the FEMD has produced 93.2%. The confusion matrix of our approach and FEMD is given in Figure 3.14.

We have also calculated True Positive (TP), True Negative (TN), False Positive (FP), False Negative (FN), and based on these the F-Score values using (3.3).

$$F - Score = \frac{2 \times TP}{2 \times TP + FP + FN} \tag{3.3}$$

The class-wise F-Score comparison between our approach and FEMD is given in Figure3.15.

1	95	0	0	0	0	0	0	0	0	5
2	0	98	0	0	0	0	0	1	0	1
3	0	0	94	0	0	1	0	0	0	5
4	0	0	0	98	1	1	0	0	0	0
5	0	0	0	0	97	2	0	0	0	1
6	0	0	0	2	0	91	0	0	0	7
7	0	0	0	0	0	0	100	0	0	0
8	0	0	0	0	0	0	0	100	0	0
9	0	0	0	0	5	0	0	0	95	0
10	0	0	0	0	0	0	0	0	0	100
	1	2	3	4	5	6	7	8	9	10

(a)

1	95	1	0	0	0	0	0	3	1	0
2	3	86	4	2	0	0	1	4	0	0
3	0	2	94	2	0	0	2	0	0	0
4	0	0	4	87	6	0	3	0	0	0
5	0	0	0	7	89	3	1	0	0	0
6	1	2	0	0	0	95	0	0	2	0
7	0	0	1	0	0	1	96	2	0	0
8	6	2	0	0	0	0	0	92	0	0
9	1	0	0	0	0	1	0	0	98	0
10	0	0	0	0	0	0	0	0	0	100
	1	2	3	4	5	6	7	8	9	10

(b)

FIGURE 3.14: Confusion matrix of (a) proposed approach and (b) FEMD-based approach.

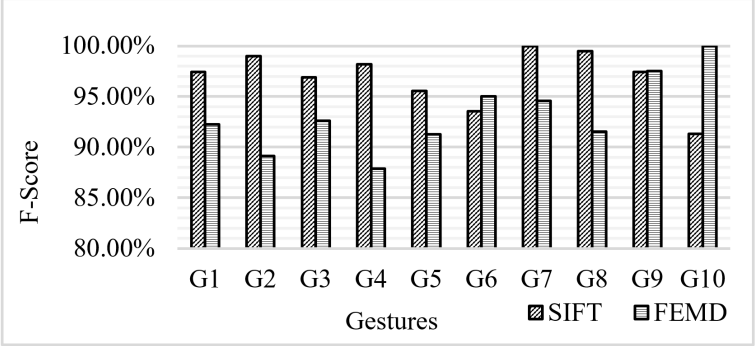


FIGURE 3.15: F-Score comparison between proposed approach and FEMD [2]

From the Figure 3.14(a), we find that the accuracy of Gesture 2, 4, 5, 7, 8 have been improved significantly as expected because SIFT features are found more robust in the benchmark dataset. Moreover, we prove this by comparing the results with binary and edge images. In binary or edge images, a small variation in the shape may cause significant changes on the tangent vectors at the points on the shape. Since we are considering local hand-finger features for the hand poses we are getting better results. Shape changes over time-series data are not

required to be considered. Recognition accuracy of Gesture 4 and 5 have increased to 98% and 97% respectively compared to FEMD-based approach. In gesture 6 and 10, we are getting most confusing results. Gesture 6 is all finger open gesture and contains a maximum number of keypoints (6374) as shown in Figure3.5 and includes no bending finger information. The same is for Gesture 10 and it is the only gesture in the dataset which contains no bending finger information like other gestures. The pose was given by the user opposite to other gestures, the bending fingers were facing towards the user, not the camera.

3.4 Limitations

In this study, we applied SIFT algorithm in grey-scale varying depth images. Here, the SIFT algorithm extracts keypoint from equally distributed high contrast regions of the depth image. However, the computational time of the SIFT algorithm is higher than other keypoint descriptor-based algorithms like ORB, SURF [110]. The feature dimensions are determined by the number of cluster centroids that we took using k-mean clustering algorithm. These keypoints are sparsely indexed to different clusters. So, if centroid varies, the cluster assignments may also vary. The depth quantization equation in 3.1 depends of a number of parameters like η, $greyLevel_{min}$, $greyLevel_{max}$. These parameters are empirically determined parameters based on the benchmark dataset in [2]. In this dataset we found the $dist_{min} = 5mm$ and $dist_{th} = 200mm$. However, these values may vary or need to be adapted based on the dataset. So, researcher who will apply these quantization technique, need to study these parameters on the working dataset.

3.5 Conclusion

In this study, we applied SIFT algorithm in grey-scale varying depth images. Here, the SIFT algorithm extracts keypoint from equally distributed high contrast regions of the depth image. However, the computational time of the SIFT algorithm is higher than other keypoint descriptor-based algorithms like ORB, SURF [110]. The feature dimensions are determined by the number of cluster centroids that we took using the k-mean clustering algorithm. These key points are sparsely indexed to different clusters. So, if the centroid varies, the cluster assignments may also vary.

Preparing depth silhouettes of the gesturing hand is one the factors that affect the accuracy of gesture recognition system. With the help of depth map information, we were able to produce those gesturing images using fast and effective segmentation process. Choosing the right cluster size is also important. Our empirical results indicate that 1600 is the most desirable number of clusters to attain the best accuracy. This large number of clusters is contributed by the fact that images with only edge information or binary images contain far less keypoints than that of depth images. The number of training samples that we have taken were sufficient to develop the cluster model as well as the SVM classification model.

In the next two chapters, we focus on depth-map utilization technique in dynamic hand gesture recognition systems, first in air-writing recognition approach and second in deep-learning-based multimodal dynamic hand gesture recognition approach. We emphasize the fact that our technique of depth quantization also works in increasing recognition accuracy.

Chapter 4

A system to recognize motion-oriented movement information as dynamic gestural events using depth-map in on-air English Capital Alphabet (ECA) writing tasks

On-air writing can be considered as a time-dependent event where hand gesture is produced in a natural environment through index finger movement. A sequence of such movements containing several time steps in 3D space can be utilized to construct an English Capital Alphabet (ECA). While Previous researches investigated 2D features, we believe that depth information may play a significant role along with other features in recognition of these dynamic gestures. We have captured hand finger motion information using a depth camera and represented them as depth images for each ECA. The hand finger trajectory data were extracted from the depth image and a combination of depth-based features and non-depth features were generated, depth variation was performed in the depth-based features, and then all the feature values were converted into time-series data. Dynamic Time Warping (DTW) distances were determined between a template ECA and a test ECA for each ECA collected from 15 participants. These distance-based features were then fed into a multi-class SVM for training and testing and got the

recognition accuracy of 80.77% without depth and 88.21% with depth-based features. To cope with the over-fitting problem we applied the resampling technique and got the highest recognition accuracy of 96.85% and at last, we applied some feature selection techniques to analyze the recognition results.

In this chapter, first we will introduce the background study and related works, then, we discuss on the methodology we followed to recognize on-air writing-based dynamic gestures. Actually the depth quantization approach that we have applied to recognize symbolic gestures, we tried to apply this technique in feature generation from our own constructed dataset. At last, we conclude the chapter by describing the result analysis.

4.1 Background study and related works

Air-writing can be defined as a motion-oriented activity of hand or finger in the free space to represent a linguistic character. The idea of recognizing 'air-writing' was incubated by Amma in [39] where he tried to recognize ECAs using wired device. The recognition of on-air alphabet writing is a part of broader gesture recognition research [119], kind of dynamic gesture recognition and air-writing might seem to be similar to online handwriting recognition [120] task. In this process, a user can lift his/her hand from the touchpad. However, in air-writing, it is difficult to differentiate which movements are part of writing and which movements are not. Consequently, many different extra strokes are mixed up with the actual writing complicating the recognition process. Moreover, while writing in the air, the hand may be near the face or the body and their similar color might be confusing due to occlusion. To overcome this problem many researchers have used special markers [39] around the writing finger. A special version of air-writing can be to write on a surface (which is not touchpad), because people feel natural writing on a surface.

The use of depth information (user distance from the camera) provided by depth camera (e.g. Microsoft Kinect, Intel Real Sense) helps to segment the hand where the traditional cameras will fail. Thanks to the depth camera for making hand segmentation and tracking process easier and faster without ambiguity. The depth information help to generate depth image and used as skeleton features to different gesture recognition systems [89]. More importantly, this depth information can be effectively utilized to represent depth-based features along with non-depth features. On-air writing requires index finger to move not only left/right (X-axis) or up/down (Y-axis) but also forward/backward (Z-axis). In the free space,

a user is not using any 2D surface for writing, so, there are some variations of depth due to the writing process which includes motion-oriented movements of the hand muscles. These variations in depth can be utilized as important features to improve air-writing recognition accuracy. However, considering the index finger movement information in Z-axis may generate an inhomogeneous distribution of smaller depth values degrading the recognition performance. So, converting the actual depth values into a scaled range of varying depth values, homogeneously distributed into certain levels should give good recognition results.

On-air-writing makes the writing process natural, unconstrained, and at the same time challenging. When someone writes an alphabet, s/he writes it as a sequence of strokes to represent the English alphabets. The best algorithm for air-writing should be able to segment the strokes accurately from the air gestures. However, in air-writing, many extra movements of the user match with perfect strokes [121] and hence become part of the writing.

In this study, we propose a system to recognize unconstrained air-writing of 26 ECAs. To facilitate unconstrained writing, we did not impose any restrictions on the user, such as 'write slowly' or 'try to write perfectly'. We represented the hand trajectories, that is, the hand movement sequence as a series of data points $(x_t,\ y_t,\ d_t)$, where $(x_t,\ y_t)$ is the position of the hand and d_t is the depth value at time sequence t. The depth values are quantized at certain levels. Those data points are converted into the time-series representation of a particular alphabet suitable for extracting features. We have determined 12 time-series features and represented those as point vectors (x and y dimensions), the depth value of the corresponding point, quantized depth value, point-wise distances, theta value, velocity, log-normal probability density functions (mean and standard deviation), freeman chain codes (4, 8, and 16). We have generated those 12 features for each alphabet from 22 users. Out of them, data from 15 users were used to generate DTW distance features and we found data from 7 users are almost perfect as we expected to consider them as templates for the DTW algorithm. Hence, one best data for each alphabet was taken manually as a template from 7 users apart from those 15 users. After normalizing those distance values we fed them into a multiclass SVM classifier. The main research contribution of this study are as follows:

1. Generation of a unique depth-based air-writing dataset consisting of 26 ECAs in an unrestricted environment.

2. Introducing DTW-based distance features for air-writing. The index finger movement trajectory was captured using depth information while writing an ECA letter and represented as time-series data.

3. Utilizing the depth information as significant features (depth value provided by Kinect as one feature and the quantized depth value as another feature) to capture the motion-oriented movement of the hand while performing natural writing in the air.

This study is the extended work of our previous research in [115], where we used only DTW-based classification considering data from one user with 5 variations. In our previous work, we did not consider DTW distances as features for a multiclass classifier. However, in this work, we took ECA gestures from 15 users, created a larger data set, and determined the DTW distance features for SVM training and testing. Moreover, we have utilized the depth information as significant features which contributed to the improvement of recognition accuracy.

Human gesture is an important input modality for communication with computers in designing gesture-based interfaces. A typical hand-gesture recognition system uses a camera (typical stereo camera) to read the hand movement data, performs the hand tracking, and then recognizes a meaningful gesture to control any devices or applications.

Air writing means gesture-based writing on the air through movement of hand fingers by which a computer system can recognize language-specific characters and other symbols in natural handwriting [121]. In the process of air-writing, each movement of the hand becomes a stroke. So, alongside the actual writing, many noises are introduced into the writing. The authors in [121], defined English characters as a sequence of strokes. The capital alphabet "A", for instance, is composed of three strokes mainly "/", "\" and "-". If the discrete strokes can be pulled out from the seemingly continuous movement of the hand, it is possible to infer the characters. The basic set of strokes for constructing the alphabet is shown in Figure 4.1.

In [39], the researchers showed how a wearable device can recognize hand gestures for air writing. The Air-writing glove fits at the back of the hand. It has motion sensors, accelerometers, and angular rate sensors equipped with a smartphone. The signals are recorded and transmitted via Bluetooth. A wearable hand motion tracking system captures movement signals using an accelerometer and gyroscope. However, converting the acceleration signal into important features to recognize

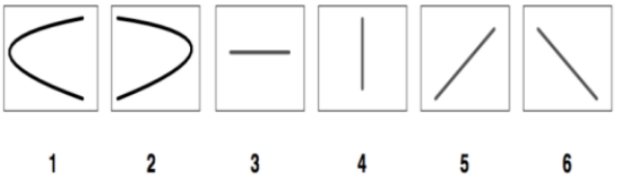


FIGURE 4.1: Basic strokes for English characters

strokes can be erroneous due to the drift of inertial sensors. Moreover, wearing a special device makes the air writing system cumbersome and not natural. Sensors attached to a glove record hand movements, a computer system captures relevant signals and translates them into text, which can then create an email, text message, or any other type of mobile app [122].

YIN et. al [123] use an online approach, attentive contexts-vector (AC-Vec), and an offline approach, attentive contexts-convolutional neural network (AC-CNN), for character recognition. Kim et. al. [124] showed a way to recognize different people's handwriting on continuous images based on the similarity of the different shapes of characters or digits based on the strokes and the ligature model. They did not use the concept of bare handwriting without using any special input pen. They tried to generate virtual 3D characters from 2D shapes using the ligature model and then used the Bayesian model to recognize real on-air writing. In our approach, we are using an unconstrained environment to write English alphabets, creating a training model using real on-air writing gestures. We are using the character shape and movement information as features in the form of time-series curves. On-air writing alphabets can be considered as signals produced at a particular time duration. So the alphabets are special curves with time variations. For example, a person can take 3 seconds to write the character 'A' and another person may take 5 seconds to write the same. Dynamic Time Warping (DTW) is a popular technique for matching variable-length signals and the DTW-algorithm is able to compare two curves in a way that makes sense and helps in matching the same patterns of the curves [125].

Researchers in [18] tried to recognize air-written Persian digits representing numbers from 0-9. They have addressed the research issues related to ligature stroke. The gradient variations on the trajectories are used as features for the recognition task. Though they said these features are scale, rotation, and translation invariant but there is a scope of further investigations considering the scale, rotation,

translation-invariant features provided by the algorithms like scale-invariant feature transform (SIFT), Speeded up robust features (SURF), Oriented FAST and rotated BRIEF (ORB), Gradient location and orientation histogram (GLOH), etc [126]. They have implemented an analytical classifier and compared the results with other state-of-the-art classifiers.

In [126], the researchers worked on symbolic hand gesture recognition and tried to recognize hand postures of 0-9 numeric symbols using depth information. They tried to extract informative SIFT features from contrast varying depth image and the contrast variation was performed through depth-map quantization. They hypothesized that the depth-map quantization process can give better recognition accuracy which was not previously explored in static hand gesture recognition. They applied the idea in the benchmark static hand gesture dataset and got better recognition accuracy compared finger-earth mover's distance (FEMD)-based [2] method. However, we found a scope to utilize the depth-map information effectively in air-writing recognition. Section 4.2.3 contains the description of our selected set of features for on-air writing recognition.

4.2 Methodology

Any recognition system must have data collection i.e. image Acquisition, preprocessing step which may include segmentation, feature extraction, and then classification. Sometimes a post-processing step may be required before classification for feature dimension reduction, feature selection for further analysis. Figure 4.2 shows an overview of our proposed system.

4.2.1 Image Acquisition

We placed a depth camera (Microsoft Kinect) in front of the individual user and asked to write an upper-case English letter considering an imaginary writing board. All the users tried to apply their self-writing style so the font-size and speed of writing highly varied. This caused the dataset very much challenging in terms of feature generation for training and testing the classification model. We asked every user to write from 'A' to 'Z' in a sequence one after another. Then we have isolated every alphabet, with the help of depth and RGB values found on the gesturing image signal. Usually, a user pause writing two consecutive letters that gives the user feel comfortable. We were able to accumulate ECA data from

22 users. However, we have manually observed, contemplated, and analyzed the individual ECA written by each user and found the best ECA data from 7 users to be considered as a template that we have mentioned earlier in Section 4.1. The rest of the ECA from 15 users were used for distance feature generation.

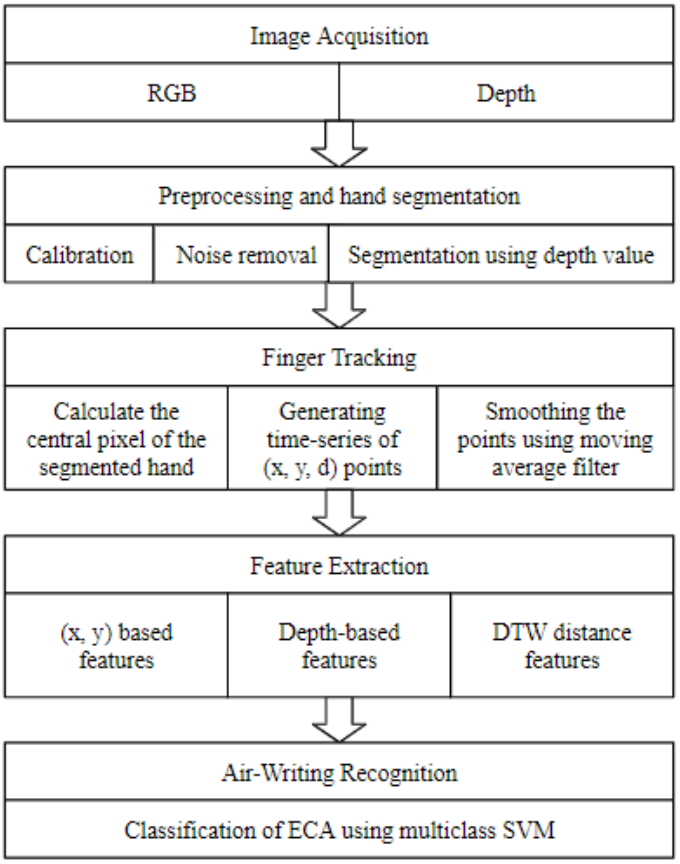


FIGURE 4.2: Proposed Approach

4.2.2 Segmentation and Pre-processing step

We assume the hand is the front-most object while the user is writing on the air. The pre-processing steps were as follows: at first, the hand from the background was separated by using depth information which is the part of hand segmentation process as shown in Figure 4.3(c).

From the segmented hand image, we have extracted the x, y coordinate of a point in that image by calculating the middle pixel location between the starting and the end position that contains non-zero pixel value. We have followed this process for each image frame and got the consecutive points of the hand movements as shown in Figure 4.3(d).

We have considered these middle points while writing a letter and generated an image consisting of that letter as shown in Figure 4.3(e). These points contain hand-movement trajectory location in the process of air-writing and the number of consecutive points represent the time-series data. So for writing each letter, we have traced out the written points (x, y) and their corresponding depth values (d), represented as points (x_t, y_t, d_t) at time t. Tracking the hand motion from image to image gave us a series of points (x_t, y_t, d_t). Those sets of points are the time-series information of a particular alphabet.

Air-writing process may lead to uncontrollable jerky movement [127] of the hand fingers. We found the written letters are not in a legitimate shape. Hence, the generated raw image is smoothed using a moving average filter [128]. The written letter after smoothing is shown in Figure 4.3 (f). The overall process of writing the ECA letter "A" is shown in Figure 4.3. The corresponding time series curve is given in Figure 4.4. In Figure 4.4, the X-Axis represents the consecutive points in total 3 seconds taken to write the ECA, "A" by a particular user and the Y-axis represents the corresponding pixel values in the gesturing image.

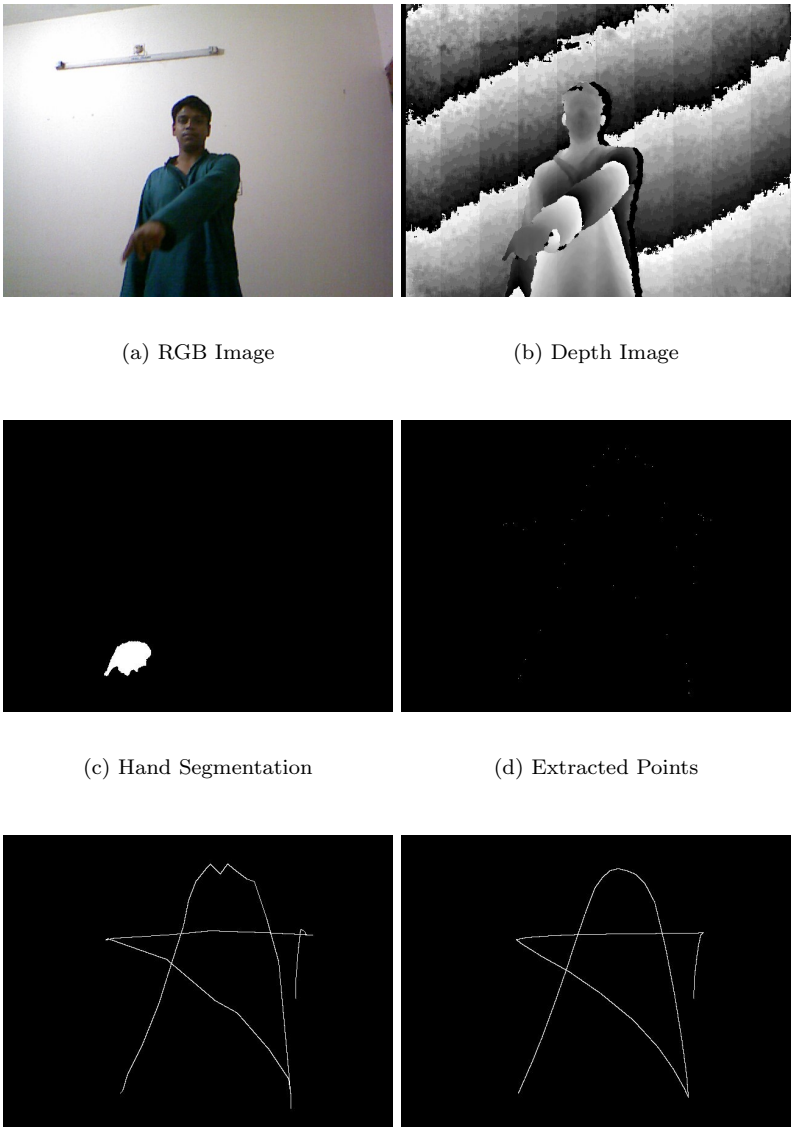
4.2.3 Feature Extraction and Classification

After converting the air written alphabet to a time-series of x, y, and d; the task is to classify them. As finding a stroke feature proved to be very difficult, we propose to classifying them based on time-series data. So, we investigated the use of DTW as the classifier. Our earlier work in [129] was about matching 2D trajectory (x, y) of an alphabet with templates and come up with a decision based on DTW distance using the equation 4.1.

$$\begin{aligned} &ClassifiedClassLabel(trajectory(x,y)) = \\ &argmin(dist(trajectory(x_{template}, y_{template}), \\ &\qquad\qquad trajectory(x,y))) \end{aligned} \tag{4.1}$$

Here, we get the minimum DTW distance between the template trajectories and (x, y) whose class is being identified.

The decision taken from the DTW distances was not that accurate with a small number of users [129]. When we increased the number of users from 5 to 15, the accuracy reduced to half. Then we looked for other features besides point vectors such as point-wise distance, theta value of points, velocity, log-normal probability



(a) RGB Image (b) Depth Image

(c) Hand Segmentation (d) Extracted Points

(e) 'A' without smoothing (f) 'A' with smoothing

FIGURE 4.3: Air-writing process to generate the letter 'A'

density, and freeman chain code that are used regularly in online handwriting recognition. We have also included quantized depth information where the depth value was converted into a range of 155-255 and 10 levels. The quantization process that we have followed is the same as shown in [126]. Each of the depth value for an ECA was quantized using 4.2

$$Q(Z) = DL_{min} + \left(\left\lfloor \left(\frac{D(Z) - D_{min}}{D_{th} - D_{min}} \times \eta \right) + 0.5 \right\rfloor \times \left\lfloor \frac{DL_{max} - DL_{min}}{\eta} \right\rfloor \right) \tag{4.2}$$

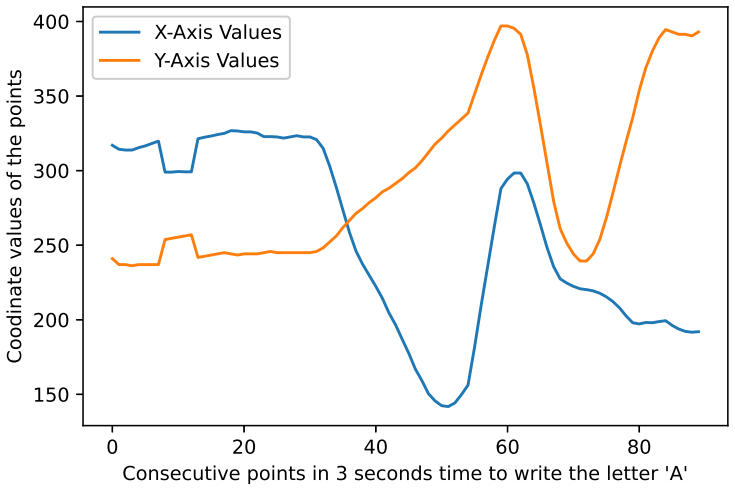


FIGURE 4.4: Time-series representation of 'A'

Here, $D(Z)$ is the distance of point at the (x, y), $Q(Z)$ is the quantized depth value of the corresponding depth-map. η is the quantization levels between DL_{min} and DL_{max}. The movement of the fingers that we have considered is the distance values between D_{min} and the depth threshold D_{th} within which the hand finger movements are found.

In total 12 features were represented as time-series information. An example time-series representation is shown in Figure 4.5 and in Figure 4.6. The list of features is given in Table 4.1. The DTW distances of the 12 time-series features were compared with the alphabet templates and directly used for classification. Still, the result was not significant to report. Phase shift in signals reduces the accuracy of recognition as the DTW algorithm does always care about the phase differences [130]. However, the geometric shape information may be required to preserve as important features to learn [131]. In such a situation, the state-of-the-art analysis suggested us to use all-pair comparison and use the DTW distance features for learning with another classifier. We choose SVM as a multi-class classifier for this purpose.

In Figure 4.5 we show the kinect-image pixel value including depth image for particular characters. Similarly figure 4.6 shows a simple derivative feature from X and Y axis. Here we selected point wise distance.

TABLE 4.1: Features used for on-air handwriting recognition

Features	Description
Feature 1 and Feature 2 (F_1, F_2): Point vector of alphabets	The point vector generates 2 time series features: one for x-dimension and one for y-dimension.
Feature 3 (F_3): Depth value of the point	Depth value was extracted from the hand trajectory and smoothed. As there as fewer movements in the depth, other derived features such as velocity was not calculated from the depth information.
Feature 4 (F_4): Quantized depth value	The quantized depth value within 155-255 using the equation 4.2
Feature 5 (F_5): Pointwise distance of point vector	This is the euclidean distance of consecutive two trajectory points (x, y).
Feature 6 (F_6): Theta value of point	This feature helps to measure pixel-wise angular distances in polar coordinate.
Feature 7 (F_7): Velocity of point	This feature helps to generate data point from pointwise distance which shows the speed within that distance either forward or backward.
Feature 8 and Feature 9 (F_8, F_9): Lognormal probability density function calculation mean and standard deviation of data point	This function is calculated based on the average and standard deviation. The Point vector that we are taking is based on time sequences which are always positive. So, the log-normal distribution function of the two dimensions will always give non-negative values. Moreover, alphabet writing does not follow normal distribution so, log-normal distribution can be a good feature for time-series classification.
Feature 10, Feature 11, Feature 12 (F_{10}, F_{11}, F_{12}): Freeman chain code of 4, 8, 16	Freeman chain code is a shape-based matching technique found to be successful in recognizing digits or characters [132].Chain code represents the sequence of direction changes between adjacent points of a curve. In [133], freeman directional code was generated for dynamic hand gesture for recognition.

4.2.3.1 DTW Distances as derived Features

After separating template and user data from the entire dataset we have converted every image to a time-series curve or signal which is shown in Figure 4.4. In the x-axis, the consecutive points that we have extracted as the point vector within a particular time in seconds are arranged to represent time-series data. At 30fps, different users have taken different amounts of time to write the same letter. In Figure 4.4, we have shown the time-series data generated from 90 frames which took 3 seconds to write the English Capital Alphabet (ECA), A. We have represented every signal to a feature vector for each ECA. The list of 12 features is given in Table 4.1.

DTW gives us minimum distances between two time-series curves. When a user writes an alphabet in an imaginary blackboard, the user does not necessarily round-up with the same length of input for the alphabets. It also varies in case of

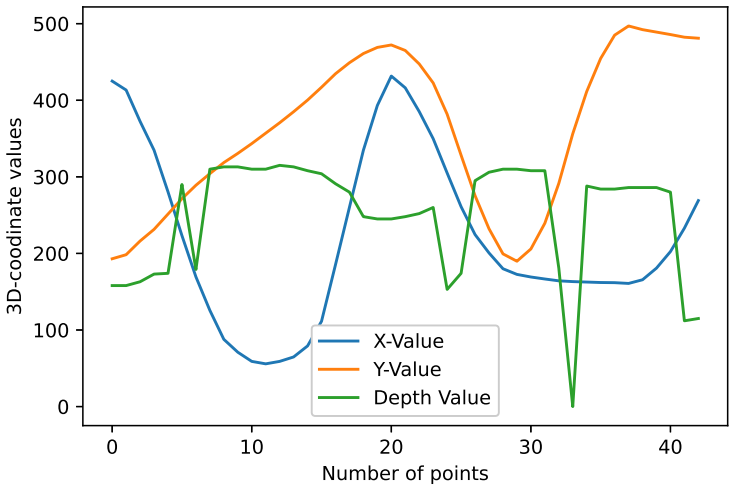


FIGURE 4.5: Time-series of point vector and the depth value for the letter, 'A'

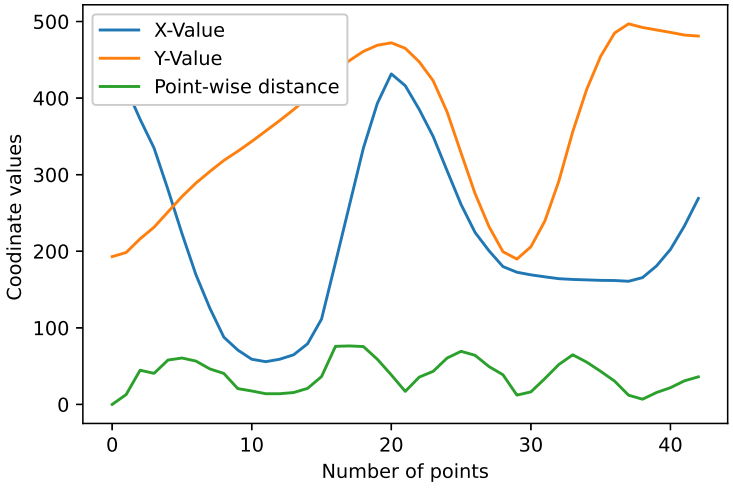


FIGURE 4.6: Time-series of point vectors and the point-wise distance values for the letter 'A'

writing the same alphabet by different users. DTW algorithm is the right one to apply in this scenario so that we can find the minimum distance between the two alphabets.

In our proposed approach, we compare an alphabet represented as a point vector, to all alphabet's point vectors using the DTW algorithm. That means, 12 time-series features of an alphabet were compared with corresponding features of the template which gives 12 distance values. Comparing an alphabet with all 26 templates generate $12 \times 26 = 312$ distance features. We have used basic DTW

equations to calculate the distances. The class label for these 312 distance features is given as the alphabet under consideration. We define the set of users as S, where, $|S| = 15$. For our 15 users, we have calculated 12 DTW distances (between 12 source features and 12 template features for each alphabet) to produce a 312-dimensional feature vector and normalized them between 0 to 1. So, each user writing 26 letters produces $15 \times 26 = 390$ instances and 312-dimensional feature per instance. Let Σ be the time-series representation of ECA characters 'A' to 'Z' generated by each user $s \in S$, i.e.

$$\Sigma = \{U^A,\ U^B,\ U^C, \ldots, U^Z\} \tag{4.3}$$

and T be the time-series representation of the template of each alphabet, i.e.

$$T = \{T^A,\ T^B,\ \ T^C\ \ \ldots,\ T^Z\} \tag{4.4}$$

Each user generated ECA character and the template ECA character contains 12 features, i.e.

$$F = \{F_1,\ F_2,\ F_3\ \ \ldots,\ F_{12}\} \tag{4.5}$$

where features in the user generated ECA and template ECA possibly have different lengths. Each feature consists of normalized values from 0 to 1.

$$F_i = \{x \in R, 0 \leq x \leq 1, \forall i \in [1, 12]\} \tag{4.6}$$

We determine the pair-wise minimum DTW distance between U^A and T^A, between U^A and T^B, and so on up to between U^A and T^Z for all the 12 features. These are the distance features generated by taking the distances between U^A and all the template elements of T. This makes the first instance of the first user writing ECA character 'A' which we denote as follows:

$$\begin{aligned} S_1^A = [&DTW(U^A_{F_{1\ldots12}}, T^A_{F_{1\ldots12}}), DTW(U^A_{F_{1\ldots12}}, T^B_{F_{1\ldots12}}), \\ &\ldots DTW(U^A_{F_{1\ldots12}}, T^Z_{F_{1\ldots12}})] \end{aligned} \tag{4.7}$$

Let $DU^{A,A}_{F_{1\ldots12}}$ are the DTW distance features between U^A and T^A, $DU^{A,B}_{F_{1\ldots12}}$ be the DTW distance features between U^A and T^B, and so on. We continue to generate these 12 features for each character up to ECA character 'Z' and the features for the last DTW distance between U^A and T^Z are $DU^{A,Z}_{F_{1\ldots12}}$. Thus we get $12 \times 26 = 312$ features for the first user writing ECA character 'A'.

So, for the first user generating 26 ECA characters we get 26 samples and the first

training sample is S_1^A, the second training sample is S_1^B and so on. For all the 15 users, $\tau^{A...Z}$ will produce a training set of size 390×312 with 26 class labels. We can represent this training set using the matrix as per Eq. (4.9).

$$S_{i\in S}^{A...Z} = \begin{Bmatrix} DU_{F_{1...12}}^{A,A} & \dots & DU_{F_{1...12}}^{A,M} & \dots & DU_{F_{1...12}}^{A,Z} \\ \dots & \dots & \dots & \dots & \dots \\ DU_{F_{1...12}}^{M,A} & \dots & DU_{F_{1...12}}^{M,M} & \dots & DU_{F_{1...12}}^{M,Z} \\ \dots & \dots & \dots & \dots & \dots \\ DU_{F_{1...12}}^{Z,A} & \dots & DU_{F_{1...12}}^{Z,M} & \dots & DU_{F_{1...12}}^{Z,Z} \end{Bmatrix} \tag{4.8}$$

$$\tau^{A...Z} = \begin{Bmatrix} S_1^{A...Z} \\ \dots \\ S_7^{A...Z} \\ \dots \\ S_{15}^{A...Z} \end{Bmatrix} \tag{4.9}$$

4.3 Experimental results

We have evaluated the proposed system in our own generated air-writing dataset consisting of 26 ECAs gestures performed by 22 users (all are male). There were no pre-instruction or guidelines on font size, speed of writing which makes the dataset more challenging. The number of samples per ECA varies among different users as we can see in Figure 4.7.

The maximum number of samples (102) required to write an ECA is 'E' while the minimum number of samples required to write the ECA characters are 'I' and 'L' by most of the users. A total of 19350 samples were collected to build the ECA dataset. Moreover, if we analyze the writing speed as given in Table 4.2, there were also variations in the duration of writing the same ECA by different users. To write 'E', the average number of users required the maximum amount of time 1.69 seconds whereas the minimum 0.86 sec and 1.01 sec time required to write 'L' and 'I' respectively.

From all the samples features were extracted as described in section 4.2.3. To prepare the training set for the classification we need unified dimensional feature vectors. However, we have considered DTW distances as features for training and testing considering the following reasons:

TABLE 4.2: Information on the speed of ECA air-writing by 22 users in Seconds

ECA	Avg	Std	Max	Min	ECA	Avg	Std	Max	Min
A	1.41	0.48	3.00	0.73	N	1.07	0.28	1.67	0.43
B	1.42	0.49	2.23	0.67	O	1.01	0.29	1.83	0.63
C	0.92	0.33	1.70	0.50	P	1.10	0.33	2.17	0.57
D	1.19	0.43	2.63	0.63	Q	1.22	0.41	2.13	0.60
E	1.69	0.69	3.40	0.97	R	1.22	0.37	2.27	0.77
F	1.38	0.45	2.43	0.73	S	1.19	0.39	2.20	0.73
G	1.33	0.56	2.53	0.50	T	1.06	0.36	2.17	0.63
H	1.40	0.50	2.60	0.77	U	1.02	0.34	1.97	0.57
I	1.01	0.30	1.97	0.57	U	1.02	0.34	1.97	0.57
J	1.14	0.36	1.93	0.63	W	1.27	0.44	2.83	0.70
K	1.34	0.45	2.43	0.67	X	1.12	0.40	2.70	0.67
L	0.86	0.28	1.53	0.43	Y	1.16	0.48	3.00	0.67
M	1.12	0.32	2.00	0.70	Z	1.13	0.32	1.80	0.63

1. Air-writing is a temporal activity that can be represented as time-series data but due to variation of movement time, while writing, the length of two ECA varies.

2. Variable-length time-series values can be represented as features if they are in fixed-size; DTW distance features generated using Eq.(4.9), gave us a unified dimensional feature vector for training and testing.

3. All-pair comparison of the features among ECAs helps the classifier to learn the information related to phase differences.

The dataset contains 312 DTW distance features with 390 instances. However, 25% and 50% resampling applied in the dataset gave us 468 and 572 instances respectfully. To analyze the impact of depth information in recognition results, we have prepared two sets of datasets, one is without depth information (390×260) and the other one is with depth information (390×312) including their two sets of re-sampled versions (468×260 and 572×260; 468×312 and 572×312). Moreover, we have prepared three sets of dataset containing features related to correlation analysis. Thus we have prepared 12 datasets to conduct the evaluation. The description of the datasets is given in Table 4.3. So, in general we can divide our dataset in to two groups: Dataset with depth information (in Table 4.3, Dataset 1, 3, 5, 7, 9, and 11) and datasets without depth information (in Table 4.3, Dataset 2, 4, 6, 8, 10, and 12). We have used 12 datasets to understand the significance of depth information from different perspectives, like, taking all the features, taking only the depth features, taking the re-sampled features, taking features after correlation analysis.

Table 4.3: 12 datasets for air-written ECA recognition

Datasets	Dimensions	Dataset Description
Dataset 1	390×260	Without depth information
Dataset 2	390×312	With depth information
Dataset 3	468×260	Without depth, 25% resampled
Dataset 4	468×312	With depth, 25% resampled
Dataset 5	572×260	Without depth, 50% resampled
Dataset 6	572×312	With depth, 50% resampled
Dataset 7	390×120	Without depth, attribute selected using correlation
Dataset 8	390×155	With depth, attribute selected using correlation
Dataset 9	390×135	Without depth, attribute selected using Information Gain
Dataset 10	390×171	With depth, attribute selected using Information Gain
Dataset 11	390×139	Without depth, attribute selected using Gain Ratio
Dataset 12	390×171	With depth, attribute selected using Gain Ratio

The cross-validation process was performed by splitting the dataset into k-folds to train the model on all the samples except the (k-1) folds and evaluated the model on the fold that was not used for training. This process was repeated k-times and recorded the average accuracy. 2-Fold, 5-Fold, 10-Fold, and 15-Fold cross-validations were performed in the datasets and the recognition accuracy is recorded. All the experimental evaluations were conducted on the Intel Core I7 2.60 GHz CPU of 16 GB RAM. We collected hand gesture images of each ECA at a resolution of 640×480 pixels at 30 fps with the help of Kinect in a C# and Microsoft dot net based system. All the pre-processing steps, finger tracking, feature extraction, and dataset preparation for machine learning were implemented using Matlab software. We perform machine learning analysis using Weka [134] software.

To measure the classification accuracy we have determined True Positive Rate (TPR), False Positive Rate (FPR), precision, recall, F-measure, Receiver Operating Characteristics (ROC) area value, and Precision-Recall Curve (PRC) area value. The classifier performed well as we can see the average ROC area value for 10-Fold cross-validation is within 0.9 to 1. Moreover, the average PRC value for all the datasets is between 0.8 to 0.9. The classification result for dataset 8 has been summarized in the confusion matrix as given in Table 4.4. This dataset contains the minimum number of features with depth information compared to other dataset containing depth features. Moreover, after the dataset 6 (572x312, 50% resampled, accuracy 96.85%), this is the dataset with depth features for which we got the maximum accuracy (88.2%) in 10-fold cross-validation. The average accuracies that we have got for TPR, FPR, precision, recall, F-measure, ROC,

and PRC are 88.2%, 0.5%, 89.6%, 88.2%, 88.4%, 99.5%, and 93.9% respectively. We got the highest F-Measure score for the letters 'C', 'H', 'I', and 'Q' which is 100% and lowest 66.11% for the letter 'N'. The result is generated considering 155 normalized DTW distance features.

TABLE 4.4: Confusion Matrix of Dataset 8

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C	0	0	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D	0	0	0	10	0	0	0	0	0	0	0	0	0	4	1	0	0	0	0	0	0	0	0	0	0	0
E	0	1	0	0	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F	0	0	0	0	0	14	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
G	0	0	0	0	0	0	12	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0
H	0	0	0	0	0	0	0	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
J	2	0	0	0	0	0	0	0	0	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
K	0	1	0	0	0	0	0	0	0	0	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L	0	0	0	0	0	0	0	0	0	0	0	13	0	0	0	0	0	0	0	0	0	1	1	0	0	
M	0	0	0	0	0	0	0	0	0	0	0	0	13	2	0	0	0	0	0	0	0	0	0	0	0	0
N	0	1	0	1	0	0	0	0	0	0	0	0	1	12	0	0	0	0	0	0	0	0	0	0	0	0
O	1	0	0	0	0	0	0	0	0	0	0	0	0	0	14	0	0	0	0	0	0	0	0	0	0	0
P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0	0	0	0	0	0	0
Q	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0	0	0	0	0	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	14	0	0	0	0	1	0	0	0
S	0	1	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	10	0	0	1	0	0	0	1
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	13	0	0	0	0	0	0
U	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	2	0	0	0	0
V	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	12	0	0	0	0
W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12	3	0	0
X	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	12	0	0
Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	13	0
Z	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	11

We can see the cross-validation comparison results of the prepared 12 datasets divided into two groups: Dataset with depth information (in Table 3, Dataset 1, 3, 5, 7, 9, and 11) and datasets without depth information (in Table 3, Dataset 2, 4, 6, 8, 10, and 12). We have used these 12 datasets to understand the significance of depth information from different perspectives, like, taking all the features, taking only the depth features, taking the re-sampled features, taking features after correlation analysis. After performing the k-fold cross validation of each of the datasets, we were able to achieve higher accuracy for depth-feature-based datasets. The comparison results are shown in Figure 4.8. The highest accuracy we were able to achieve is 96.85% for dataset 6 for which the dataset was generated by taking random subsamples from the original dataset (dataset 2) with replacement. The datasets considering depth information always gave better accuracies compared to datasets without depth information. Taking all the features gave us 9.16%, 5.16%,

5.40%, and 5.35% accuracy improvements in 2-Fold, 5-Fold, 10-Fold, and 15-Fold cross-validations respectively over the datasets that do not contain depth features.

We tried to understand the impact of different features to justify our recognition accuracies. All the features except F_3 and F_4 are derived features from F_1 and F_2. The features F_1 and F_2 only gave us the recognition accuracies of 25.1282%, 27.6923%, 29.3208%, and 30.7992% for 2-Fold, 5-Fold, 10-Fold, and 15-Fold cross-validations respectively. However, if we add one-by-one feature the accuracies improved significantly. For example, if we add features F_5, F_6, F_7, F_8, F_9, and F_{10} with feature F_1, F_2 which are without depth features F_3, F_4, we got 157%, 181%, 169%, 160% improvement and with depth features we got 173%, 197%, 182%, 171% (for 2-Fold, 5-Fold, 10-Fold, 15-Fold cross-validations respectively) improvements. We justified the use of taking 3 freeman chain code features F_{10}, F_{11}, and F_{12}, we wanted to take only one feature out of these three features. However, taking three features gave us overall accuracy improvement around 2.62% compared to taking any one feature.

We applied feature selection techniques to cope with overfitting problems and ranked the features based on information gain, gain ratio, Pearson's correlation. We remove features that do not contain significant information. We tried to understand the relationship between different features and their corresponding class labels, tried to analyze which features and how many features contribute more to recognition accuracy. We found that features with depth information contribute more to recognition accuracy for all the feature selection techniques we applied. In the case of information gain and the gain ratio of the attribute with respect to class, we got 48 features with a ranked value greater than 0, resulted in 73.33% accuracy. Removing 100 worst-ranked features for both of the techniques gave us 80.77% accuracy. However, we got the highest accuracy by removing 141 features starting from the last, which means, using the top 171 features we found the highest accuracy 85.13% with depth features. We removed the features without depth information within these 171 features and got the highest 73.85% accuracy out of 135 features. We tried to select the features based on Pearson's correlation coefficient values, with cut-off value 0.07 gave us 155 features, 88.21% accuracy with depth features, and 82.05% accuracy using 120 features without depth features. After this empirical analysis we found that in the case of information gain and gain ratio methods, the difference between with-depth features and without-depth features is 36 and 32 respectively whereas in case of Pearson correlation the difference is 35. However, Information gain or Gain ratio based methods gave us 171 features including depth features and Person correlation method gave us

155 features. So, we retain a minimum number of features with depth values using Pearson correlation technique in case of dataset 8 which gave us the highest recognition accuracy. We found that features with depth information always gives better results compared to features without depth information. The highest difference we got for dataset 10 is 15% and the lowest difference we got for dataset 4 is 2% and in an average, for all the datasets we got 7% difference in 10-Fold cross-validation.

4.4 Limitations

In this study, we proposed an air-writing dataset that contains unconstrained writing of 22 users. The dimension of the dataset for which we got an accuracy of 88.2051% is 390×155. Here if we could increase the number of users around 40 to 50, then the machine learning model might learn more information. As the number of features was 312 and the samples were 390 in dataset 1, we had to reduce the number of features based on correlation analysis using a threshold. However, more studies could be performed to determine the threshold in correlation analysis. Moreover, we could feed the raw writing images into deep-learning-based models to analyze classification results on spatio-temporal information. However, in that scenario, the number of samples needs to increase a lot. The depth quantization equation in 4.2 used in this air-writing research also takes the empirical values of DL_{max}, DL_{min}, D_{th}, and D_{min}. These values may vary based on the input environment.

4.5 Conclusion

In this research study, we tried to recognize on-air hand-written characters of English Capital Alphabets (ECAs) through a Kinect depth camera. It is a vision-based spatio-temporal activity in which the hand trajectory vectors were generated and utilized for each of the gesturing images. we have created a unique dataset in a complex natural environment with the help of 15 users. Each of the ECAs is presented as time-series values containing 12 discriminating features. Then, all pair DTW distances were calculated and a total of 312 distance features represented each of the alphabets. We got 390 instances containing the 312 feature dimensions for SVM training and testing. However, we also analyzed the recognition accuracy by removing features from the ranked feature list based on information gain,

gain ratio, and correlation analysis. With these, we have generated 12-datasets with a different number of features based on feature analysis. We have performed 2-Fold, 5-Fold, 10-Fold, and 15-fold cross-validation and found high recognition accuracy of 96.84% by resampling instances and also 88.21% using 155 ranked feature list based on feature correlation analysis for our selected number of features. These results we have achieved considering depth information as important features compared to non-depth features. In the future, we will continue our work to recognize small-letter English alphabets as well as Bangla alphabets including word recognition.

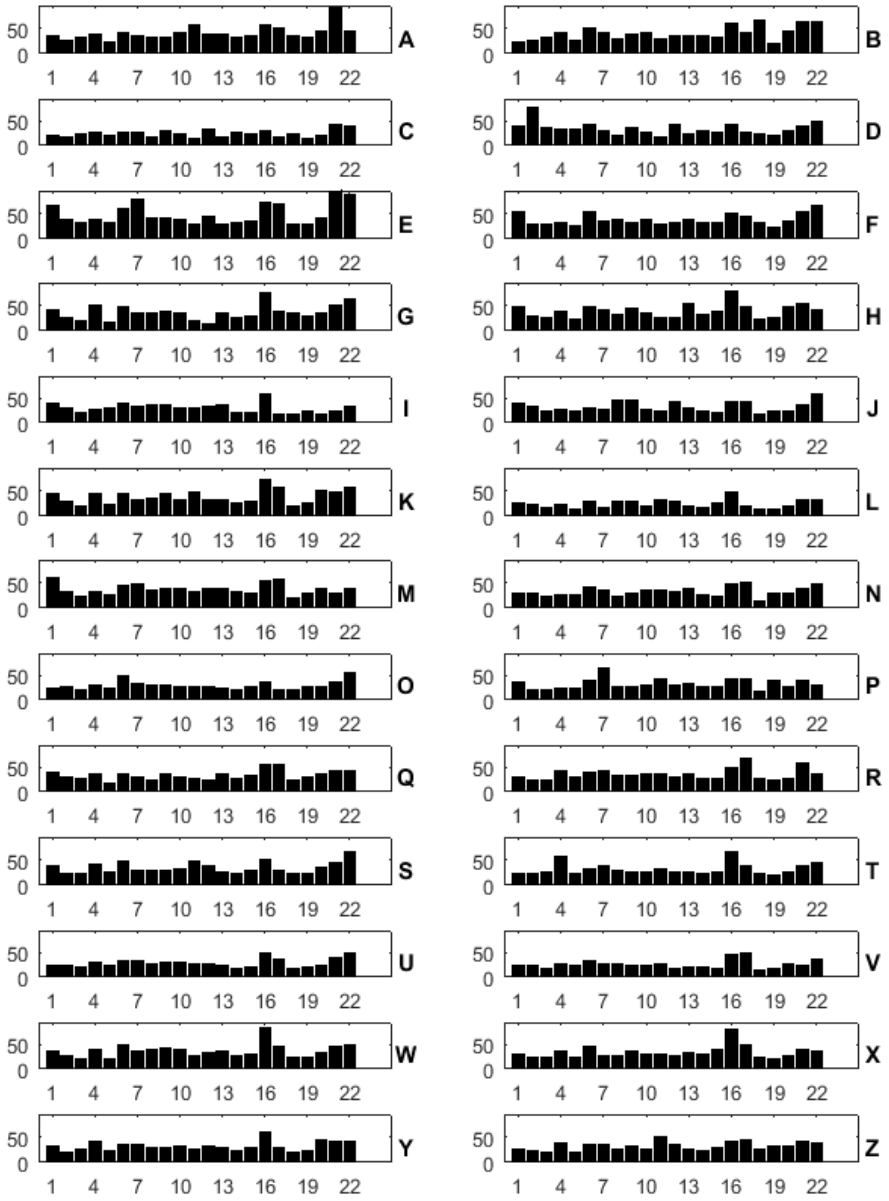
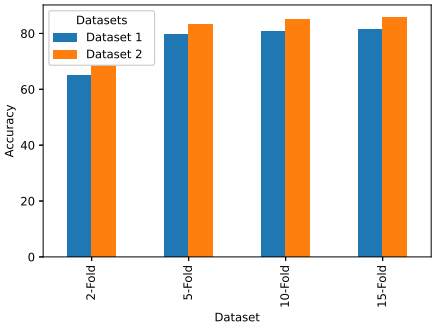
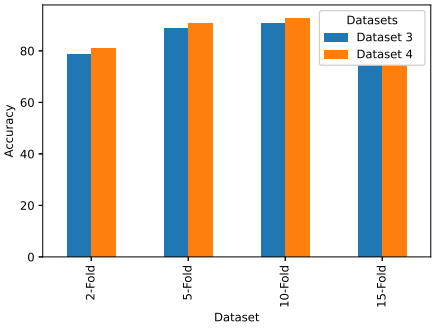


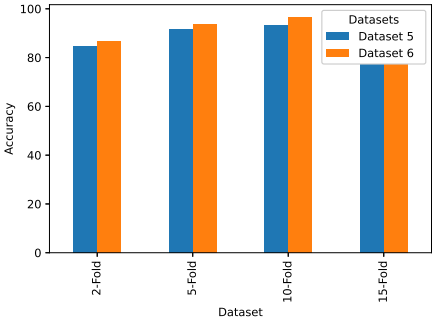
FIGURE 4.7: Sample distribution of 22 users for each ECA (A to Z) where x-axis represents user number and y-axis represents the number of samples per user



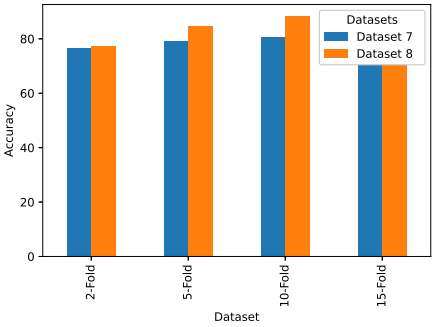
(a) Dataset 1 vs. Dataset 2



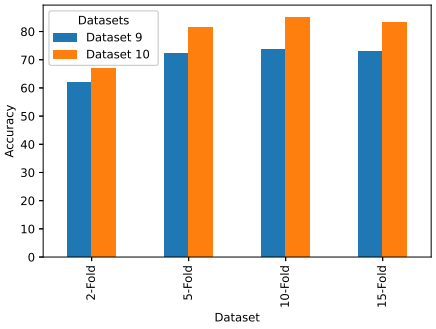
(b) 25% resampled, Dataset 3 vs. Dataset 4



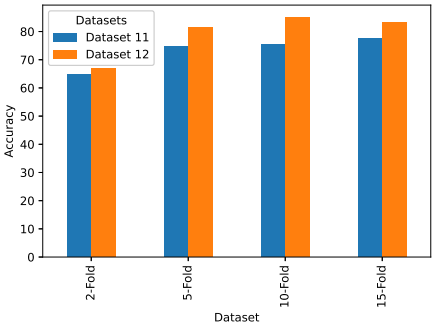
(c) 50% resampled, Dataset 5 vs. Dataset 6



(d) Attribute selected using correlation, Dataset 7 vs. Dataset 8



(e) Attribute selected using Information Gain, Dataset 9 vs. Dataset 10



(f) Attribute selected using Gain Ratio, Dataset 11 vs. Dataset 12

FIGURE 4.8: Comparison of cross-validation results of the 12 datasets grouped by without depth features and with depth features as described in Table 4.3 where x-axis represents cross-validation folds and y-axis is represents accuracy

Chapter 5

A multimodal deep Learning-based dynamic hand gesture recognition using depth information

Any spatio-temporal movement or reorientation of the hand, done with the intention of conveying a specific meaning, can be considered as a hand gesture. Inputs to hand gesture recognition systems can be in several forms, such as depth images, monocular RGB, or skeleton joint points. We observe that raw depth images possess low contrasts in the hand regions of interest (ROI). They do not highlight important details to learn, such as finger bending information (whether a finger is overlapping the palm, or another finger). Recently, in deep-learning-based dynamic hand gesture recognition, researchers are tying to fuse different input modalities (e.g. RGB or depth images and hand skeleton joint points) to improve the recognition accuracy. In this paper, we focus on dynamic hand gesture (DHG) recognition using depth quantized image features and hand skeleton joint points. In particular, we explore the effect of using depth-quantized features in Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) based multi-modal fusion networks. We find that our method improves existing results on the SHREC-DHG-14 dataset. Furthermore, using our method, we show that it is possible to reduce the resolution of the input images by more than four times and still obtain comparable accuracy to that of the original resolution.

In this chapter, first, we discuss the related research on deep learning-based multimodal dynamic hand gesture recognition in section 5.1, second, we elaborate on our multimodal approach consisting of gray scale varying image and hand skeleton joint point as input in a two stream CNN-LSTM-based fusion network and recognition methodology in section 5.2, third, we describe the result analysis in section 5.3, and at last, we give the conclusion remark in section 5.5.

5.1 Background study and related works

In our daily lives, we both consciously and subconsciously use numerous hand gestures. Human hands are dynamic and highly dexterous, allowing hands and hand movements to encode or represent a large variety of information. This capacity of hand gestures to represent information is second only to that of natural language. To account for physical disabilities related to speech, we use the symbolic sign language, where hand gestures play a significant role.

Hand gestures are particularly suitable for interaction based applications. Although their embedding capacity is lower than that of natural language, speech controlled interaction has to consider the problem of vocal fatigue, or language barriers (for example, a person not knowing English may not be able to interact with their English-based system). In contrast, hand gestures are easy and natural to use. They can often be understood intuitively (such as pointing to a person or an object) - which is why people resort to gestures if there is a language barrier in communication. Due to these reasons and more, gesture-based interaction have long been introduced to many Human Computer Interaction (HCI) applications. They play a key role in the rapidly growing field of ambient intelligence, assisting us in interacting with smart homes and smart appliances. Gestures are also important in applications such as sign language communication, interacting with virtual objects in virtual environments, controlling robots through hand gestures, playing virtual reality games with hand movements, etc. Thus, the development of robust hand gesture recognition systems can be considered as a key area of HCI research.

Formally, a hand gesture can be defined as the movement of the hands and fingers, in some particular orientation, with the intention of conveying meaningful information. This information can be something like some specific object (indicated by pointing fingers), or perhaps some intention (thumbs up indicating approval), or even specific symbols (fingers representing digits). Although many hand gestures

are universal, they can also be culture or context specific. Symbolic gestures are generally static, that is, they exist only in the spatial domain. For example, the index finger representing the number 1. This is a time invariant or static hand gesture. There are also dynamic hand gestures, which represent broader meanings, like waving the hand to mean hello. These gestures work in both the spatial and temporal domain. We may think of such dynamic gestures as a sequence of static gestures which together correspond to a new meaning.

There are multiple approaches to hand gesture recognition. Computer Vision (CV) based approaches based on regular images require restrictions on the gesturing environment, such as special lighting conditions, simple and uncluttered background, and absence of occlusions. Alternative sensor based approaches utilize gloves embedded with accelerometers, gyroscopes, bend sensors, proximity sensors, and other forms of inertial sensors. However, this sensor-based gesture recognition approach has limitations in terms of naturalness, cost, user comfort, portability, and data preprocessing. Advances in stereo vision and infrared (IR) cameras have lifted a lot of constraints on CV-based approaches by making depth information available for use. On depth images, it is possible to recognize gestures with a combination of a feature extraction mechanism and a discriminating system. For example, Scale-Invariant Feature Transform (SIFT) can be used to form feature vectors, which can then be fed into a classification model like a Support Vector Machine (SVM)[126].

Despite their effectiveness in understanding spatial data, CNNs however are not the most suitable solution to dynamic hand gesture recognition. This is because DHG recognition is also distinctly time-dependent. Recurrent Neural Networks (RNN) are a subset of Deep Learning methods which deal with temporal features. In particular, Long Short Term Memory (LSTM) are highly useful in modelling long range dependencies, which may be the case in dynamic hand movements. As such, combinations of CNN and RNN based networks excel at the DHG recognition problem.

Deep learning models do not require hand-crafting features, but they still benefit from good preprocessing. While studying the depth images used as inputs to existing CNN-RNN systems, we observed that there is not much emphasis on the hand region of interest (ROI). The fingers and the palm of the hand occupy a relatively similar depth value. We believe this apparent lack of contrast hides some meaningful information, which may be useful to gesture recognition models. For example, when the fingers overlap against the palm, this isn't really visible in

the corresponding depth image (as seen in Figure 5.1). However, by quantizing the depth values into specific depth levels, the contrast between fingers and palms is increased and we gain additional information.

FIGURE 5.1: (left) Original image (right) Image with quantized depth levels

In [126], the researchers verified the usefulness of this method regarding static hand gesture recognition. They found that the gray-scale variations based on depth values in depth images gives higher recognition accuracy compared to without gray-scale variation up to 3.6%. However, In case of dynamic hand gesture, user perform gestures in 3D surface and certain gestures may vary only in Z-axis. Dynamic gestures include motion-oriented movements of the hand muscles that can be utilized as important depth features to improve gesture recognition accuracy. Researchers in [135], have utilized this concept in the recognition on-air hand writing recognition of English capital alphabets (ECA). They showed that the varying depth values distributed into certain levels based on the actual depth value gave better recognition results if they are combined with other non-depth features.

In this work, we focus on applying our method to multi-modal CNN and RNN based fusion networks [136], for the task of dynamic hand gesture recognition.

Gesture and activity recognition has been an actively researched field throughout the past decade. Due to the development of various types of sensors, it was possible to study several forms of input modalities for the task of hand gesture recognition, such as color, depth, acceleration, infrared, etc. Furthermore, advances in the field of machine learning and deep learning have also had a significant effect on gesture recognition research.

Before the commercialization of depth-aware sensors, the predominant type of input to hand gesture recognition systems was color or RGB data, due to their relative ubiquity. One such early color-based approach was by Iwai et al. [137], who utilized colored gloves along with decision trees to perform gesture recognition.

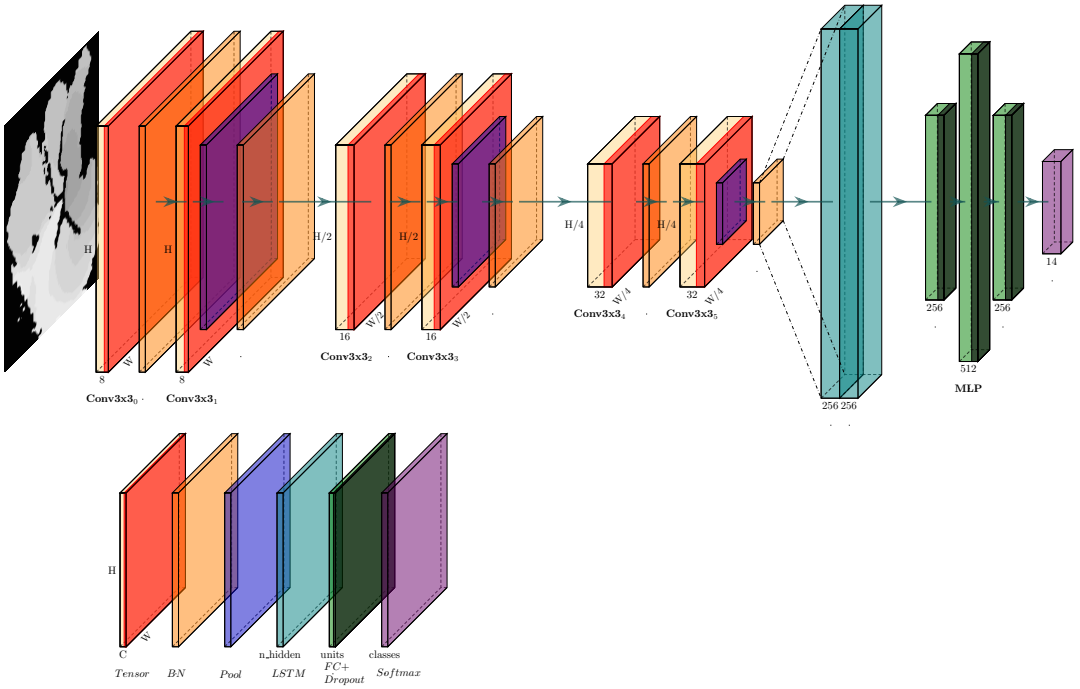


FIGURE 5.2: Depth CNN-LSTM

In recent times, many core CV problems have benefited from advances in Deep Learning (DL). Much of this success can be attributed to the development of Convolutional Neural Networks (CNN), which are translation-invariant and excel at extracting spatial features. The inclusion of Deep Learning methods in Computer Vision problems reduces the need of choosing and crafting good features. As such, the vision-based approach to gesture recognition has also adopted the usage of CNNs.

The research work by Lai et al. 2018 in [138] focuses on developing a multi-modal network for the dynamic hand gesture recognition problem. The inputs to the model consists of 16-bit depth images and 2D skeleton joint points. Lai et al. explored several types of fusion methods in their work, including feature level fusion, score level fusion, and decision level fusion.

The hand finger joint points pass through LSTM layers, while the corresponding depth images pass through CNN layers and then LSTM layers. This produces

two independent feature maps. In feature level fusion, these two resultant feature responses are concatenated, and then passed through a multi-layered perceptron (MLP). In score level fusion, each feature map is passed through a separate MLP. The final two logit function responses are then combined by either taking their maximum or their average. In decision level fusion, rather than combining the logit outputs, the confidence responses of the final softmax layer are combined instead.

Empirically, Lai et. al. has shown that decision level fusion does not work well in practice. Feature level fusion and score level fusion have better performances, with score level (average) showing the best results.

The research work in [136], followed deep leaning-based approach for temporal 3D pose recognition based on a combination of CNN and LSTM networks. They proposed double stage training (CNN, then LSTM) where, CNNs are designed to detect spatial patterns related to the position of the skeleton joints in 3D space and the LSTM used to capture the spatio-temporal patterns related to the time evolution of the 3D coordinates of the skeleton joint. However, they did not consider multi-modal inputs, it is still of some interest to us as it uses a combination of CNN and LSTM layers for the task of dynamic hand gesture recognition. They have considered uni-modal inputs in the form of 3D skeleton joint points. Their model contains CNN layers followed by LSTM layers, to facilitate both spatial and temporal feature extraction.

Different researchers in [139],[140], [141], tried to use variety of input modality and applied deep learning methods to learn human action or gestural features. However, there is research scope in multi-modal data fusion in deep learning techniques.

The research works in [126], the researchers verified their method on static hand gesture recognition. They applied depth quantization on depth images to increase contrast between palm and fingers. Using the contrasted gray-scale depth image, we applied the SIFT algorithm to produce robust feature descriptors. These features, represented as 128-dimensional feature vectors, were fed into an SVM classifier. They showed that using depth quantization on the input depth image resulted in improved accuracy.

5.2 Methodology

Our proposed system consists of: (1) quantization of depth values into discrete gray levels, and (2) a multi-modal Convolutional-Recurrent Neural Network (CRNN) architecture which takes in a sequence of image-frames and a corresponding sequence of 2D skeleton joint points as input, and performs the dynamic hand-gesture recognition.

FIGURE 5.3: (left) Original (right) Gray-scale Variation

5.2.1 Gray-scale Variation

The depth images in the SHREC-DHG-14/28 dataset [142] are 16-bit images. Of the available pixel range, only a very small portion is actually used by the hand gestures. Furthermore, the hand ROI does not possess enough contrast to highlight some features which may be useful to the recognition process.

For instance, information about finger position, orientation, overlap and motion (over multiple frames) can be useful in fine-grained gesture recognition. We thus address this issue with our preprocessing method, termed Gray-scale Variation, which aims to increase the contrast in the hand region of interest.

The operation is pixel-wise, and can be formulated as:

$$f(x,y) = G_{min} + \left(\left\lfloor \left(\frac{D(x,y) - D_{min}}{D_{th} - D_{min}} \times \eta \right) + 0.5 \right\rfloor \times \left\lfloor \frac{G_{max} - G_{min}}{\eta} \right\rfloor \right) \quad (5.1)$$

Where $f(x,y)$ denotes an output pixel, and $D(x,y)$ denotes an input pixel from the input depth image.

From a high level overview, the Gray-scale Variation operation reassigns depth values into η discrete buckets, or *gray levels*, thus creating several sharply contrasted regions. Moreover, the amount of output contrast is subject to some pre-specified parameters.

We initially choose G_{min} and G_{max}, two parameters which determine the effective range of the output pixels. We also choose the parameter η, which represents the number of discrete gray-scale quantization levels in the output image. As such, there are η unique depth values in the output, evenly distributed between G_{min} and G_{max}.

The input to the operation is a hand ROI from a depth image, denoted as D in equation (5.1). $D(x,y)$ and $f(x,y)$ represent input and output pixels respectively, while D_{min} is the minimum value in the input hand ROI (ignoring the zero-valued background pixels). As stated earlier, we select η as the number of grey levels between G_{min} and G_{max}. Consequently, we also select η depth segments between D_{min} and $(D_{min} + D_{th})$ — where D_{th} is the distance, we assumed the hand would be from D_{min} and the depth threshold.

TABLE 5.1: Recognition Rates (%) on the DHG-14 dataset

Method	Fine			Coarse			Both		
	Best	Worst	Avg ± Std	Best	Worst	Avg ± Std	Best	Worst	Avg ± Std
FL-Fusion-Concat [cite]	90.00	48.00	72.90 ± 10.30	98.89	78.89	86.83 ± 4.68	87.86	67.86	81.86 ± 5.38
SL-Fusion-Avg [cite]	92.00	52.00	76.00 ± 10.51	97.78	81.11	90.72 ± 4.64	95.00	72.86	85.46 ± 5.16
GVAR-FL-Fusion (ours)	100.0	50.0	86.89 ± 12.43	100.0	74.28	91.13 ± 7.013	100.0	74.44	89.61 ± 7.53

As the operation is dependent upon pre-specified parameters, it is necessary to understand the rationale behind setting those parameters. A very low value of G_{min} would make it difficult to distinguish the hand from the background, and if G_{min} and G_{max} are not sufficiently spaced apart, the range of possible values would be compressed (and thus not have as much contrast as intended). The choice of η also affects the quality of the output — for example, if we use all available gray-levels (256), we would not obtain any useful contrast. However, if we use too few gray-levels, (like perhaps 2-4) we may lose a significant amount of spatial information.

Empirically, a well-balanced choice of G_{min} and G_{max} are 155 and 255 respectively, with $\eta = 10$ levels between them.

We apply equation (5.1) on our input depth hand ROIs and get quantized gray-scale hand ROIs, which are supplied to our model as inputs alongside 2D skeleton joint points.

5.2.2 Proposed Architecture

The neural network architecture that we use can be divided into two main sub-networks: (1) A CNN + LSTM network which processes gray-scale image sequences, and (2) an LSTM network which processes 2D skeleton joint points. We then explore two forms of fusion: (a) feature-level fusion, where we concatenate the feature maps from the two components before passing them to a dense classifier, and (b) score-level fusion, where we pass each component's feature map through a separate classifier head, finally taking the average of the logits (prior to the softmax operation).

The CNN + LSTM sub-network is composed of two components — a CNN component and an LSTM component. The CNN component is composed of three convolutional blocks, B_{conv}. Each block B_{conv} consists of two (3×3) 2D-convolutional layers, each followed by ReLU non-linearity. The second ReLU is further followed by a (2×2) max-pool layer.

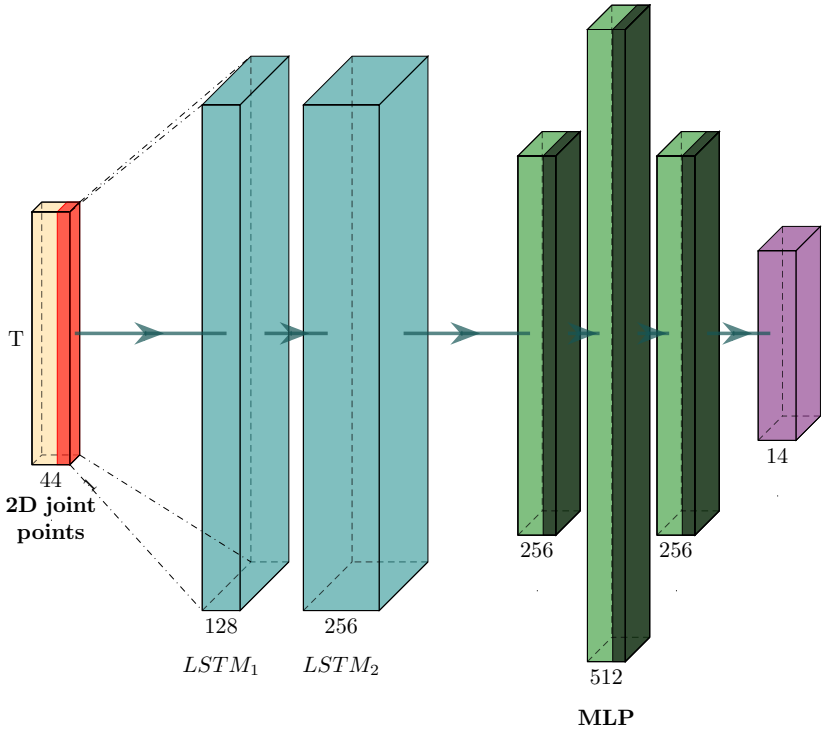


FIGURE 5.4: Joint LSTM

We also added two Batch-Normalization layers into the block — first after the initial ReLU, and second after the max-pool layer. The rationale behind this placement is that Batch Normalization is intended to *normalize* the inputs to convolutional layers, and thus they are placed immediately prior to them. The LSTM component of the CNN + LSTM consists of two LSTM layers, each with 256 hidden units. The overall depth based CNN+LSTM architecture can be observed in Figure 5.2.

It is to be noted that the CNN component is used for extracting spatial features, while the trailing LSTM component is used for extracting temporal features. Thus, the CNN is applied in a time-distributed manner. An arbitrary input image-sequence tensor may be of dimensions (BS, T, C, H, W) — where BS represents batch size, T represents the time-step size or sequence-length, and C, H, W represent the channel, height and width resolutions of the images respectively. This input tensor is passed to the CNN component in the form $(BS * T, C, H, W)$; the CNN component is actually sequence independent. The resultant feature-maps are reshaped back to the form $(BS, T, features)$ before being passed to the LSTM component.

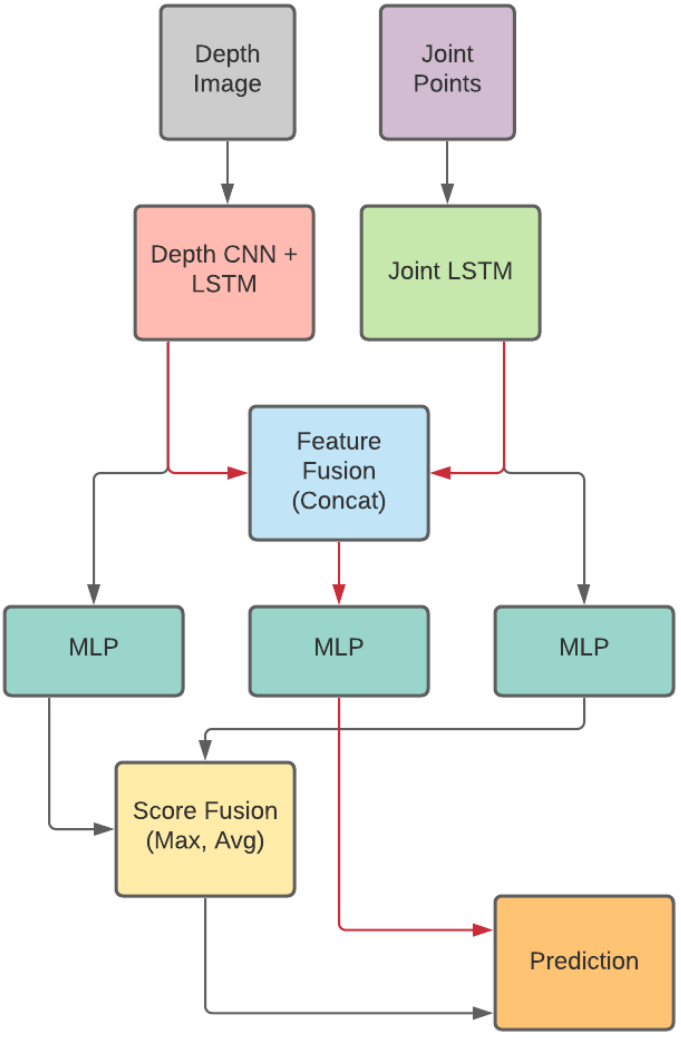


FIGURE 5.5: Overview of fusion methods

Previously, Lai et al.[138] did not utilize any Batch-Normalization layers in their proposed model. We believe using Batch-Norm is critical to speeding up the training process for this sort of model layout. Because of the time-distributed manner in which 2D-Convolutions are applied on image sequences, we are effectively performing the convolutions on a large batch size, $BS * T$. This means that the model would be prone to internal covariate shift, and would thus take longer to converge. Adding Batch-Norm layers into the depth CNN reduces the necessary training time significantly.

The joint based LSTM network consists of simply two LSTM layers, with 128 and 256 layers respectively. The inputs to the joint LSTM are a sequence of 2D

skeleton joint points, supplied in the form of a tensor with dimensions $(BS, T, 44)$. Figure 5.4 shows an overview of the joint LSTM.

As stated earlier, we explore two forms of combining the resultant features from the two sub-networks. In feature-level fusion, the two feature tensors are combined by concatenating, and are then passed into the MLP. In score-level fusion, no such feature concatenation is done. As such, we have two separate dense MLP classifiers in the score-level fusion method. The MLP classifiers consist of three fully-connected or dense layers with 256, 512 and 256 units respectively, followed a dense layer with N_C units (where N_C represents number of gesture classes). ReLU non-linearity is used in between the layers.

Previous works [138] demonstrated a notable variance problem on the SHREC task. As such, we used Dropout layers (with a drop probability of 0.5) in between the dense MLP layers, in order to regularize the model.

5.3 Experimental results

We conducted our experiments with the methods described above. We follow a similar experimental setup to [22, 138, 143, 144, 145], using a 20-fold Leave-One-Out Cross Validation strategy, where the model is trained on 19 subjects and evaluated on the remaining one in each fold.

5.3.1 Dataset

Our work primarily focuses on the SHREC-DHG-14/28 dataset [142]. The dataset consists of 14 types of dynamic hand gestures, performed two ways: with one finger and with two fingers. The gestures are performed by 28 different people, with each person repeating a gesture between 1 and 10 times, in the two ways described above. This leads to a total of 2800 data instances.

Each hand gesture instance consists of a sequence of depth image frames, a sequence of 2D skeleton joint points, and a sequence of 3D skeleton joint points. There are primarily 14 target gesture labels. The 14 gestures are further categorized into fine and coarse grained gestures, which can be seen from Table 5.2. The fine grain gestures involve more acute movements of the fingers, while the coarse grain gestures involve motion of the entire hand or arm.

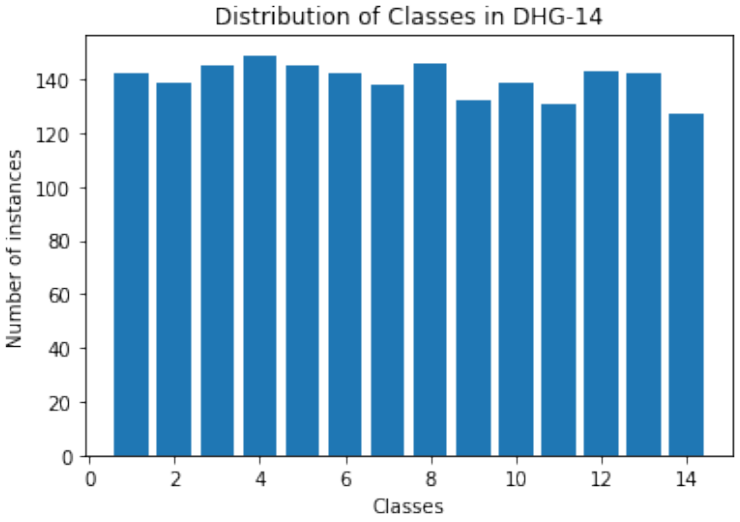


FIGURE 5.6: Distribution of Classes in DHG-14

FIGURE 5.7: Depth Image frame with Corresponding Joint Point

The dataset also comes with a train and test split, with 1960 (70%) data instances in the training set and the remaining 840 (30%) in the test set. Furthermore, the training and test sets were formulated in such a way that the training test consists data from exactly 20 of the total 28 performers. This is reasoning behind the twenty-fold cross validation scheme used in [22, 138, 142]; the training set consists of 20 unique subjects. Because each validation fold in the leave-one-out method contains data from an unseen performer (thus of a slightly different data distribution), the results of this evaluation process give a fair idea of the learning algorithm's robustness.

TABLE 5.2: Gesture Recognition Classes in DHG Dataset

Class	Gesture	Grain
0	Grab	Fine
1	Tap	Coarse
2	Expand	Fine
3	Pinch	Fine
4	Rotation Clockwise	Fine
5	Rotation Counter-clock	Fine
6	Swipe Right	Coarse
7	Swipe Left	Coarse
8	Swipe Up	Coarse
9	Swipe Down	Coarse
10	Swipe X	Coarse
11	Swipe V	Coarse
12	Swipe +	Coarse
13	Shake	Coarse

5.3.2 Data Preparation

First, we extracted all the hand ROI from the depth image frames (the ROI coordinates are available in the dataset). As opposed to the 227×227 resolution used in [138], we resized our images to a much smaller 50×50 resolution. Furthermore, to speed up the training process, we applied the grayscale-variation preprocessing method over the entire depth-image dataset in prior, thus creating a transformed dataset.

For training our model, we utilize the depth image frames and corresponding 2D skeleton joint points (not using the 3D skeleton data). A time-step size of 32 is used — this means that we use exactly 32 frames from a given data instance. It is possible for data items to contain both more and less than 32 frames. As such, for sequences with less than 32 frames, we pad with blank frames, and for sequences with more than 32 frames, we perform evenly distributed sampling between the start and end frame.

5.3.3 Experimental Design

We ran several experiments with our proposed models, under different settings. First, we evaluated the method proposed in [138] with exact settings, except using input depth frames of 50×50 resolution instead of 227×227. Second, we evaluated our approach, on the similar 50×50 resolution images, preprocessed by the grayscale variation operation. We follow some similar parameters to [138]

– a timestep size of 32 and a batch size of 16 is used. We use a smaller initial learning rate of 0.03 and train our model on 50 epochs (as opposed to the 100 epochs training done in [138]).

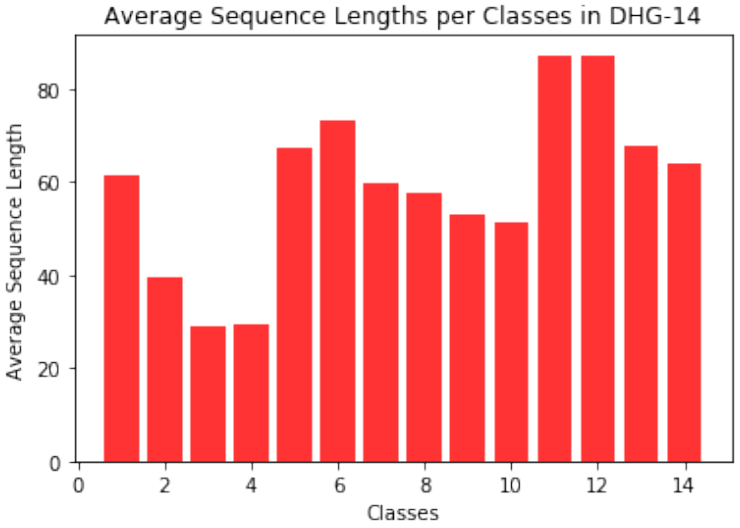


FIGURE 5.8: Distribution of Sequence Lengths in DHG-14

5.3.4 Result Analysis

It is apparent from Table 5.1 that despite using significantly fewer parameters and inputs at much lower resolutions, our proposed system shows significant improvement in generalization on the DHG-14 dataset. The average performance of our feature level fusion setup shows an improvement of almost ∼8% over the feature level fusion shown in [138].

More importantly, it was previously noted in [138] that multimodal fusion models struggle to perform on fine gestures, as seen on Table 5.1. It can be observed from our results that we've shown a drastic improvement in the performance of the model on fine-grained gestures, having an average accuracy about ∼14% higher. We argue that the reasoning behind this improvement is a combination of factors — such as, a more regularized model, addressing the previously ignored internal covariate shift, and our proposed preprocessing method.

In particular, it can be observed from Figure 5.4 that the GrayscaleVariation operation reduces the contrast present in the hand ROI, and also highlights the fingers and extremities of the hand. We believe this representation of the depth images allows the model to extract some additional useful information, which is

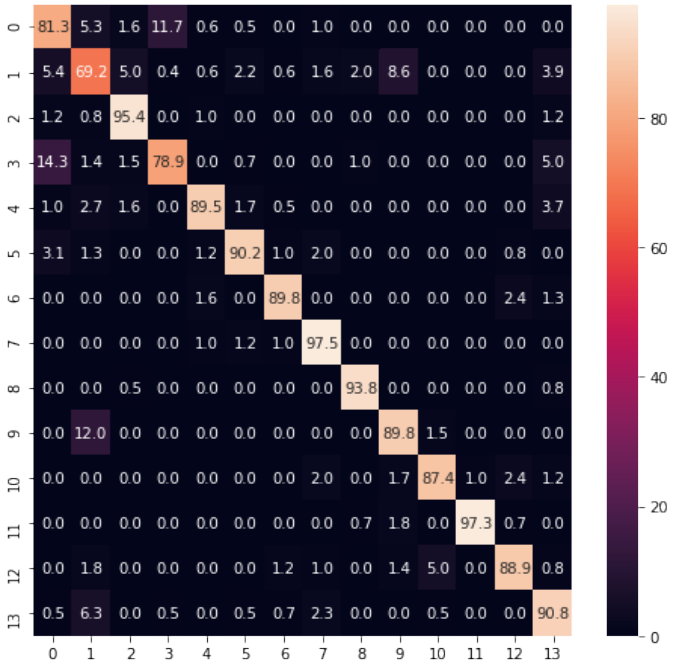


FIGURE 5.9: Averaged confusion matrix for GVAR-feature-fusion

critical to the performance of fine gestures, where finger movements dominate the gesture sequence.

Figure 5.9 shows the confusion matrix, averaged over the twenty trained models. Although from Table 5.1 we do see that our method has significantly improved results, the confusion matrix actually provides us with deeper insights. We can see that there is still some room for improvement — the model shows a comparatively poor performance on class 1, or Tap. It is possible that there may be some relationship between the length of the gestures and performance, as the worst performing classes have a comparatively low average sequence length, as observable from Figure 5.8.

5.4 Limitations

We have tested our proposed greyscale variation and CNN+LSTM method for dynamic hand gesture recognition on the DHG-14/28 dataset only. For a better idea of the robustness of our method, it is necessary to conduct rigorous experiments on multiple dynamic hand gesture benchmarks which have the depth-map modality. Furthermore, apart from depth-maps and 2D hand-skeleton coordinates (in image space), the DHG-14/28 dataset also contains another modality that we have

not utilized: 3D hand-skeleton coordinates (in world space). Since 3D skeletons also contain depth information, there is an additional scope to study the effect of quantizing the depth values of 3D coordinates instead of depth pixels. Lastly, deep learning experiments are subject to a large number of tunable hyperparameters, each of which may subtly affect accuracy. Due to the expensive 20-fold cross-validation experiment design, it was difficult for us to sufficiently search the hyperparameter space. It may be possible to obtain slightly better results with a good set of hyperparameters we have not explored.

5.5 Conclusion

Our work is primarily focused on studying the multi-modal fusion approach to dynamic hand gesture recognition. We can summarize our contributions into a few core points:

1. We proposed a new depth quantization method, Gray-scale Variation, which is useful in highlighting additional information in low-contrast depth frames.
2. The original multimodal fusion approach to dynamic hand gesture recognition was computationally expensive. In our approach, we performed gesture recognition where the number of input pixels of the depth image frames (2.5K pixels) are roughly equal to only about 5% of the number of input pixels in [138] (51K pixels). Because the model's number of trainable parameters are directly dependent on the input spatial resolution, our model is significantly smaller than the previous model — 6.9 million parameters, compared to 31 million parameters. This makes the model much more suitable for real time and edge applications.
3. We showed an increase in the overall accuracy of the multimodal fusion architecture, with the addition of Gray-scale Variation and our minor modifications to the model architecture. Our model requires half as much training time. Most notably, our approach shows a significant increase in the recognition accuracy of fine-grained gestures — an increase of about $\sim$14%.

Chapter 6

Conclusion

Since the inception of depth-map capturing technology, the depth values are continuing to study from different perspective by the research community. Specially, in the hand gesture recognition research area, primarily depth values were effectively used in hand segmentation and localization. In computer vision-based hand gesture recognition approaches, one of the prominent research challenges was hand ROI extraction from full image consisting of complex background (e.g. changes in illumination, cluttered or occluded objects, and so on) and object itself. With the help of depth-map information, it became very faster and accurate to find the interested objects. Later, the researcher started to utilize the depth-map information as salient features to be learnt by the machine learning algorithms. Starting from direct depth images generated from depth map to different types of features representation (e.g. coordinate values of the depth dimension along with pixel dimensions, depth matrix generated from depth image, depth-map projections, and so on). Moreover, 3D skeleton hand joint points, different orientation, translation, rotation information of those joint points, motion information of the joint points, shape of connected joints in the hand movements are being effectively utilized as feature sets to recognize both static and dynamic hand gesture recognition. Recently, the study of depth-map utilization is focusing on deep-learning based approaches due to the fact that, deep-learning environment setup became affordable and easier. Now-a-days researchers are trying to utilize huge input volume consisting of gestural image set to learn more high-level features or abstract level features. They are designing single-layered or multilayered deep-learning architecture to understand spatial relationships (e.g. using CNN) as well as temporal relationships (e.g. 3DCNN, LSTM) among the gestural images. Initially, the study was limited to only RGB image or RGB image-based features. However, researchers are now

also trying to learn the impact of depth-images or depth-based features in deep learning-based methods in different fusion-based techniques. Rather than perceiving from single modality of input, the research has shown that multiple modalities (e.g. RGB image, depth image, skeleton joint points, optical flow images etc.) can significantly improve the recognition results. Use of hybrid deep learning-based approaches (e.g. two-stream network, multi-stream network) are also trying to capture high-level features from multimodal input.

In this thesis, we propose to utilize depth information in pre-processing steps so that, rather than learning from the depth images directly, some significant information if provided earlier before learning then, the process can improve the recognition results consequently. We started to experiment with our proposed depth utilization technique (e.g. depth quantization process) firstly in static hand gesture recognition system. We have generated gray-scale varying depth images using depth-map information from the benchmark dataset using our proposed methodology and found that the machine model responding well. We have got better recognition accuracy by applying the depth quantization technique that in turn helped in extracting significant features from low contrast, low resolution depth images. The features we chose are robust in terms of scale, rotation, translation, and orientation invariant property. After that we extend our experiment to use the proposed technique in dynamic gestural event like on-air writing activity. While writing in the air with bare finger the direct depth dimension along with the depth-quantized value showed higher recognition accuracy. To do that, we have generated our own air-writing dataset and on that dataset we studied feature-selection analysis to understand the significant of depth-based features and non-depth features. We found that depth-based feature if merged with non-depth features can improve recognition results. We also analyze the comparative results of the direct depth-valued features and our proposed quantized depth-valued features. The result showed around 3.5% improvement over direct depth-valued features. At last, in the recent deep-learning-based approach, we experimented our proposed method in dynamic hand gesture recognition in the state-of-the-art dynamic hand gesture dataset. From that benchmark dataset we applied our depth-quantization technique to generate gray-scale variation depth images with a goal to use them as another input modality in addition to the hand skeleton joint points. We design a fusion-based deep-learning network consisting of CNN and LSTM which has taken those two input modalities and tried to learn the important features. We actually performed feature-level fusion and decision-level fusion by which we achieved

state-of-the-art accuracies in our proposed multimodal technique of dynamic hand gesture recognition.

We can particularly outline three future directions of this research. First, we have determined and used the distance threshold in the depth quantization process from each of the image frames of a particular hand gesture, which can be considered as the local threshold. However, there is a scope to choose an adaptive threshold considering all the gestural images of all the classes and calculate the global threshold to test the accuracy changes. Second, with air-writing, it is possible to increase the number of samples and then design a spatiotemporal-based deep learning model to recognize bare hand writing. Third, in the multi-modality-based deep learning approach of dynamic hand gesture recognition, we have not considered 3D hand skeleton joint points. Since 3D joint points also contain depth information, there is a further scope to study the effect of quantizing the depth values of 3D coordinates instead of depth pixels only.

Appendix A

Scale Invariant Feature Transform (SIFT) [1]

SIFT algorithm produces rotation and scale invariant 128-dimensional feature descriptors. SIFT algorithm works in four steps.

A.0.1 Detecting scale space extrema

We calculate the Laplacian of Gaussian (LoG) for the image , with diffetent sigma (σ) values which represents the scale parameter. We convolute the image with gaussian filter to produce a blurred image, L, using A.1.

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \tag{A.1}$$

where, $G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2/2\sigma^2)}$

SIFT algorithm uses Difference of Gaussian (DoG), using A.2, by taking the difference of blurred images for two different values (σ and $k\sigma$), i.e.

$$DoG(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma) \tag{A.2}$$

This process is done for different octaves of the image in Gaussian Pyramid. For our case, we got optimal values with initial $\sigma = 1.275$ and $k = \sqrt{2}$. With this process we get scale invariant representations of hand gesture features.

A.0.2 Keypoint localization and filtering

Once *DoG* images are found, they are searched for local extrema over scale and space. One pixel in the image is compared with its 8 neighbours as well as 9 pixels in next scale and 9 pixels in previous scale. Then, a pixel is selected if it is larger or smaller (extrema) than all 26 neighbours. This is a keypoint best representing in a scale. We remove the edge keypoints which are subject to aperture problem using Harris corner detection and reject points with low contrast using thresholding in the *DoG* images.

A.0.3 Orientation assignment

We have found that keypoints stable after localization and filtering. We assign orientatoin to each keypoint to get rotation invariant property. Since we already know the scale and location of the extrema, we calculate the gradient direction and magnitute of each pixel around the keyoint. Gradient magnitude and orientation are determined using A.3 and A.4.

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2} \tag{A.3}$$

$$\theta(x,y) = tan^{-1}\frac{L(x,y+1) - L(x,y-1)}{L(x+1,y) - L(x-1,y)} \tag{A.4}$$

A gradient histogram (orientation histogram) is created with 36 bins covering 360 degrees for each keypoint and 80% of points represent the directions as a keypoint direction.

A.0.4 Keypoint descriptor

Each keypoint has x, y, σ, m, θ. We create the keypoint descriptor by taking a 16×16 window of neighbourhood around the keypoint. It is devided into 16 sub-blocks of 4×4 size. For each sub-block, 8 bin orientation histogram is created. So a total of $4 \times 4 \times 8 = 128$ bin values are available as a vector to form the keypoint descriptor.

Appendix B

Finger-Earth Mover's Distance (FEMD) [2]

FEMD is a shape matching-based algorithm of gesturing images consisting of hand shapes represented as time-series curves. The hand finger shape or signature represents a cluster defined as R in B.1.

$$R = (r_1, w_{r1}), ..., (r_m, w_{rm}) \tag{B.1}$$

where, r_i is a cluster representative and $w_r i$ is the weight of that cluster. The angle interval between the end points of each finger segment is defined as r_i, $r_i = [r_{ia}, r_{ib}]$. The weight of the cluster w_{ri} is the normalized area within the finger segment. The figure in B.1 shows the process.

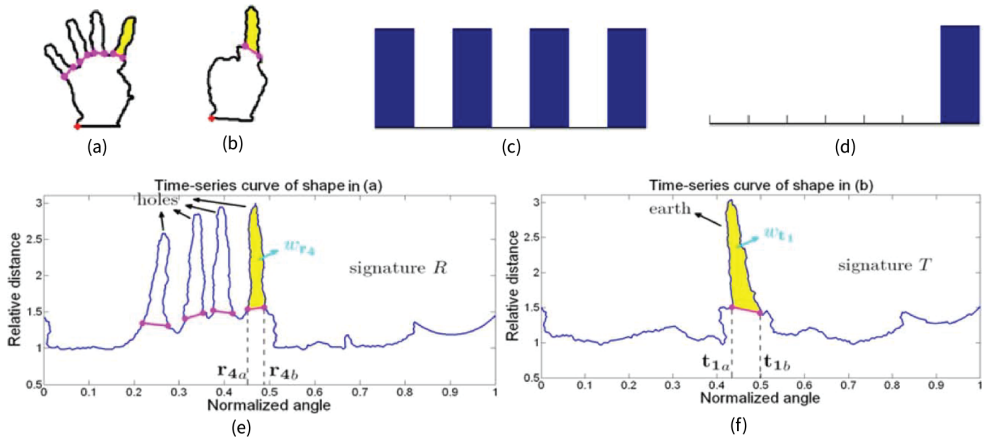


FIGURE B.1: (a)(b): two hand shapes whose time-series curves are shown in (e)(f). (c)(d): two signatures that partially match, whose EMD cost is 0. (e)(f): illustration of the signature representations of time-series curves. Image reproduced from [2].

The FEMD distance between two signatures R and T can be calculated using B.2.

$$\begin{aligned} FEMD(R,T) &= \beta E_{move} + (1-\beta)E_{empty} \\ &= \frac{\beta \sum_{i=1}^{m}\sum_{j=1}^{n} d_{ij}f_{ij} + (1-\beta) \mid \sum_{i=1}^{m} w_{ri} - \sum_{j=1}^{n} w_{tj} \mid}{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}} \end{aligned} \tag{B.2}$$

Where $D = [d_{ij}]$ is the ground distance matrix for R and T such that d_{ij} is the ground distance from r_i to t_j. d_{ij} is defined as the minimum moving distance for interval $[r_{ia}, r_{ib}]$ to totally overlap $[t_{ja}, t_{jb}]$ as in

$$d_{ij} = \begin{cases} 0 & r_i \; totally \; overlap \; with \; t_j \\ min(\mid r_{ia} - t_{ja} \mid, \mid r_{ib} - t_{jb} \mid) & otherwise \end{cases} \tag{B.3}$$

f_{ij} is the flow from r_i to t_j and constitutes the flow matrix F [2]. Finally, gesture recognition in FEMD is also achieved through template matching.

As FEMD is the dissimilarity measure, so the input hand is recognized as the class with which it has the minimum dissimilarity distance using B.4.

$$c = argmin \;\; \{FEMD(H, T_c)\} \tag{B.4}$$

Where H is the input hand signature and T is a template of class c.

Bibliography

[1] G. LoweDavid, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, 2004.

[2] 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.

[3] Depth perception: A comprehensive review of techniques used to estimate depth using machine learning and classical methods. [Online]. Available: https://beyondminds.ai/blog/depth-estimation/

[4] A. Islam, M. A. Hossain, and Y. M. Jang, "Interference mitigation technique for time-of-flight (tof) camera," in *Eighth International Conference on Ubiquitous and Future Networks (ICUFN)*, 2016, pp. 134–139.

[5] M. Van den Bergh, D. Carton, R. De Nijs, N. Mitsou, C. Landsiedel, K. Kuehnlenz, D. Wollherr, L. Van Gool, and M. Buss, "Real-time 3d hand gesture interaction with a robot for understanding directions from humans," in *2011 RO-MAN*, 2011, pp. 357–362.

[6] S. Desai and A. Desai, "Human computer interaction through hand gestures for home automation using microsoft kinect," in *Proceedings of International Conference on Communication and Networks*, N. Modi, P. Verma, and B. Trivedi, Eds. Singapore: Springer Singapore, 2017, pp. 19–29.

[7] M. Karabasi, Z. Bhatti, and A. Shah, "A model for real-time recognition and textual representation of malaysian sign language through image processing," in *2013 International Conference on Advanced Computer Science Applications and Technologies*, 2013, pp. 195–200.

[8] U. Lee and J. Tanaka, "Finger identification and hand gesture recognition techniques for natural user interface," in *Proceedings of the 11th Asia Pacific Conference on Computer Human Interaction*, ser. APCHI '13. New

York, NY, USA: Association for Computing Machinery, 2013, p. 274279. [Online]. Available: https://doi.org/10.1145/2525194.2525296

[9] J. Taylor, L. Bordeaux, T. Cashman, B. Corish, C. Keskin, T. Sharp, E. Soto, D. Sweeney, J. Valentin, B. Luff, A. Topalian, E. Wood, S. Khamis, P. Kohli, S. Izadi, R. Banks, A. Fitzgibbon, and J. Shotton, "Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences," *ACM Trans. Graph.*, vol. 35, no. 4, Jul. 2016. [Online]. Available: https://doi.org/10.1145/2897824.2925965

[10] D.-H. Lee and K.-S. Hong, "Game interface using hand gesture recognition," in *5th International Conference on Computer Sciences and Convergence Information Technology*, 2010, pp. 1092–1097.

[11] L. Gallo, A. P. Placitelli, and M. Ciampi, "Controller-free exploration of medical image data: Experiencing the kinect," in *2011 24th International Symposium on Computer-Based Medical Systems (CBMS)*, 2011, pp. 1–6.

[12] C. Nuzzi, S. Pasinetti, R. Pagani, G. Coffetti, and G. Sansoni, "Hands: an rgb-d dataset of static hand-gestures for human-robot interaction," *Data in Brief*, vol. 35, p. 106791, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352340921000755

[13] P. Molchanov, X. Yang, S. Gupta, K. Kim, S. Tyree, and J. Kautz, "Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[14] Y. Li, X. Wang, W. Liu, and B. Feng, "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://www.sciencedirect.com/science/article/pii/S0020025518301063

[15] B. Feng, F. He, X. Wang, Y. Wu, H. Wang, S. Yi, and W. Liu, "Depth-projection-map-based bag of contour fragments for robust hand gesture recognition," *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 4, pp. 511–523, 2017.

[16] O. Kopuklu, N. Kose, and G. Rigoll, "Motion fused frames: Data level fusion strategy for hand gesture recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018, pp. 1–9.

[17] N.-T. Do, S.-H. Kim, H.-J. Yang, and G.-S. Lee, "Robust hand shape features for dynamic hand gesture recognition using multi-level feature lstm," *Applied Sciences*, vol. 10, no. 18, 2020. [Online]. Available: https://www.mdpi.com/2076-3417/10/18/6293

[18] S. Mohammadi and R. Maleki, "Air-writing recognition system for persian numbers with a novel classifier," *The Visual Computer*, vol. 36, no. 5, pp. 1001–1015, 2020. [Online]. Available: https://doi.org/10.1007/s00371-019-01717-3

[19] Q. Gao, J. Liu, and Z. Ju, "Hand gesture recognition using multimodal data fusion and multiscale parallel convolutional neural network for humanrobot interaction," *Expert Systems*, vol. 38, no. 5, p. e12490, 2021, e12490 10.1111/exsy.12490. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/exsy.12490

[20] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *NIPS*, 2014.

[21] P. Molchanov, X. Yang, S. Gupta, K. Kim, S. Tyree, and J. Kautz, "Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural networks," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4207–4215, 2016.

[22] J. C. Nez, R. Cabido, J. J. Pantrigo, A. S. Montemayor, and J. F. Vlez, "Convolutional neural networks and long short-term memory for skeleton-based human activity and hand gesture recognition," *Pattern Recognition*, vol. 76, pp. 80–94, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320317304405

[23] A.-K. S. Hasan, Haitham, "Humancomputer interaction using vision-based hand gesture recognition systems: a survey," *Neural Computing and Applications*, vol. 25, no. 2, pp. 251–261, Aug. 2014.

[24] L. Zhang, P. Shen, G. Zhu, W. Wei, and H. Song, "A fast robot identification and mapping algorithm based on kinect sensor," *Sensors*, vol. 15, no. 8, pp. 19 937–19 967, 2015. [Online]. Available: https://www.mdpi.com/1424-8220/15/8/19937

[25] A. S. Al-Shamayleh, R. Ahmad, M. A. M. Abushariah, K. A. Alam, and N. Jomhari, "A systematic literature review on vision based gesture recognition techniques," *Multimedia Tools and Applications*, vol. 77, no. 21, pp. 28 121–28 184, 2018.

[26] Microsoft kinect. [Online]. Available: https://developer.microsoft.com/en-us/windows/kinect/

[27] Intel realsense. [Online]. Available: https://www.intelrealsense.com/

[28] Asus xtion pro live. [Online]. Available: http://xtionprolive.com/asus-xtion-pro-live

[29] T. Wang, Y. Li, J. Hu, A. Khan, L. Liu, C. Li, A. Hashmi, and M. Ran, "A survey on vision-based hand gesture recognition," in *Smart Multimedia*, A. Basu and S. Berretti, Eds. Cham: Springer International Publishing, 2018, pp. 219–231.

[30] T. Xu, D. An, Y. Jia, and Y. Yue, "A review: Point cloud-based 3d human joints estimation," *Sensors*, vol. 21, no. 5, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/5/1684

[31] T. Weise, S. Bouaziz, H. Li, and M. Pauly, "Realtime performance-based facial animation," in *ACM SIGGRAPH 2011 Papers*, ser. SIGGRAPH '11. New York, NY, USA: Association for Computing Machinery, 2011. [Online]. Available: https://doi.org/10.1145/1964921.1964972

[32] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *CVPR 2011*, 2011, pp. 1297–1304.

[33] J. Fu, D. Miao, W. Yu, S. Wang, Y. Lu, and S. Li, "Kinect-like depth data compression," *IEEE Transactions on Multimedia*, vol. 15, no. 6, pp. 1340–1352, 2013.

[34] Q. De Smedt, "Dynamic hand gesture recognition - From traditional handcrafted to recent deep learning approaches ," Theses, Université de Lille 1, Sciences et Technologies; CRIStAL UMR 9189, Dec. 2017. [Online]. Available: https://hal.archives-ouvertes.fr/tel-01691715

[35] Y. Li, X. Wang, W. Liu, and B. Feng, "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://www.sciencedirect.com/science/article/pii/S0020025518301063

[36] Q. Gao, J. Liu, Z. Ju, Y. Li, T. Zhang, and L. Zhang, "Static hand gesture recognition with parallel cnns for space human-robot interaction," in *Intelligent Robotics and Applications*, Y. Huang, H. Wu, H. Liu, and Z. Yin, Eds. Cham: Springer International Publishing, 2017, pp. 462–473.

[37] Y. SHI, Y. LI, X. FU, M. Kaibin, and M. Qiguang, "Review of dynamic gesture recognition," *Virtual Reality and Intelligent Hardware*, vol. 3, no. 3, pp. 183–206, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2096579621000279

[38] Y. Zhang, C. Cao, J. Cheng, and H. Lu, "Egogesture: A new dataset and benchmark for egocentric hand gesture recognition," *IEEE Transactions on Multimedia*, vol. 20, no. 5, pp. 1038–1050, 2018.

[39] C. Amma, M. Georgi, and T. Schultz, "Airwriting: A wearable handwriting recognition system," *Personal Ubiquitous Comput.*, vol. 18, no. 1, pp. 191–203, Jan. 2014.

[40] S.-K. L. Ue-Hwan Kim*, Yewon Hwang* and J.-H. Kim, "Writing in the air: Unconstrained text recognition from finger movement using spatio-temporal convolution," 2021.

[41] P. K. Pisharady and M. Saerbeck, "Recent methods and databases in vision-based hand gesture recognition," *Comput. Vis. Image Underst.*, vol. 141, no. C, p. 152165, Dec. 2015. [Online]. Available: https://doi.org/10.1016/j.cviu.2015.08.004

[42] B. Li, G. Li, Y. Sun, G. Jiang, J. Kong, Z. Ju, and D. Jiang, "A review of gesture recognition based on computer vision," in *Intelligent Robotics and Applications - 10th International Conference, ICIRA 2017, Wuhan, China, August 16-18, 2017, Proceedings, Part I*, ser. Lecture Notes in Computer Science, Y. Huang, H. Wu, H. Liu, and Z. Yin, Eds., vol. 10462. Springer, 2017, pp. 528–538.

[43] S. Desai, "Segmentation and recognition of fingers using microsoft kinect," in *Proceedings of International Conference on Communication and Networks*, N. Modi, P. Verma, and B. Trivedi, Eds. Singapore: Springer Singapore, 2017, pp. 45–53.

[44] G. Tian, Munir, X. Ma, and J. Ali Peng, "Kinect sensor-based long-distance hand gesture recognition and fingertip detection with depth

information," *Journal of Sensors*, vol. 2018, 2018. [Online]. Available: https://doi.org/10.1155/2018/5809769

[45] M. Oudah, A. Al-Naji, and J. Chahl, "Hand gesture recognition based on computer vision: A review of techniques," *Journal of Imaging*, vol. 6, no. 8, 2020. [Online]. Available: https://www.mdpi.com/2313-433X/6/8/73

[46] J. Tompson, M. Stein, Y. Lecun, and K. Perlin, "Real-time continuous pose recovery of human hands using convolutional networks," *ACM Trans. Graph.*, vol. 33, no. 5, Sep. 2014. [Online]. Available: https://doi.org/10.1145/2629500

[47] L. Guo, Z. Lu, and L. Yao, "Human-machine interaction sensing technology based on hand gesture recognition: A review," *IEEE Transactions on Human-Machine Systems*, vol. 51, no. 4, pp. 300–309, 2021.

[48] D. Indra, Purnawansyah, S. Madenda, and E. P. Wibowo, "Indonesian sign language recognition based on shape of hand gesture," *Procedia Computer Science*, vol. 161, pp. 74–81, 2019, the Fifth Information Systems International Conference, 23-24 July 2019, Surabaya, Indonesia. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050919318113

[49] A. K. G. and E. J. A., "Static hand gesture recognition using multi-layer neural network classifier on hybrid of features," *American Journal of Intelligent Systems*, vol. 10.

[50] Q. Chen, N. Georganas, and E. Petriu, "Real-time vision-based hand gesture recognition using haar-like features," *2007 IEEE Instrumentation & Measurement Technology Conference IMTC 2007*, pp. 1–6, 2007.

[51] Y. Huang and J. Yang, "A multi-scale descriptor for real time rgb-d hand gesture recognition," *Pattern Recognition Letters*, vol. 144, pp. 97–104, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167865520304165

[52] J. Li, J. Wang, and Z. Ju, "A novel hand gesture recognition based on high-level features," *International Journal of Humanoid Robotics*, vol. 15, no. 02, p. 1750022, 2018. [Online]. Available: https://doi.org/10.1142/S0219843617500220

[53] N. Pugeault and R. Bowden, "Spelling it out: Real-time asl fingerspelling recognition," in *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, 2011, pp. 1114–1119.

[54] A. Kuznetsova, L. Leal-Taix, and B. Rosenhahn, "Real-time sign language recognition using a consumer depth camera," in *2013 IEEE International Conference on Computer Vision Workshops*, 2013, pp. 83–90.

[55] C. Dong, M. C. Leu, and Z. Yin, "American sign language alphabet recognition using microsoft kinect," in *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2015, pp. 44–52.

[56] E. Ohn-Bar and M. M. Trivedi, "Joint angles similarities and hog2 for action recognition," in *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2013, pp. 465–470.

[57] C. Zhang and Y. Tian, "Histogram of 3d facets: A depth descriptor for human action and hand gesture recognition," *Computer Vision and Image Understanding*, vol. 139, pp. 29–39, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1077314215001216

[58] X. Chen, H. Guo, G. Wang, and L. Zhang, "Motion feature augmented recurrent neural network for skeleton-based dynamic hand gesture recognition," in *2017 IEEE International Conference on Image Processing (ICIP)*, 2017, pp. 2881–2885.

[59] J. Lin and Y. Ding, "A temporal hand gesture recognition system based on hog and motion trajectory," *Optik*, vol. 124, no. 24, pp. 6795–6798, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0030402613007699

[60] F. sheng Chen, C. ming Fu, and C. lin Huang, "y huang, c.: Hand gesture recognition using a real-time tracking method and hidden markov models," *Image and Video Computing*, pp. 745–758, 2003.

[61] Q. De Smedt, H. Wannous, and J.-P. Vandeborre, "Heterogeneous hand gesture recognition using 3d dynamic skeletal data," *Computer Vision and Image Understanding*, vol. 181, pp. 60–72, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1077314219300153

[62] J. Materzynska, G. Berger, I. Bax, and R. Memisevic, "The jester dataset: A large-scale video dataset of human gestures," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 2019, pp. 2874–2882.

[63] J. Wan, S. Li, Y. Zhao, S. Zhou, I. Guyon, and S. Escalera, "Chalearn looking at people rgb-d isolated and continuous datasets for gesture recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 761–769, 2016.

[64] Q. Miao, Y. Li, W. Ouyang, Z. Ma, X. Xu, W. Shi, and X. Cao, "Multimodal gesture recognition based on the resc3d network," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2017.

[65] E. Ohn-Bar and M. M. Trivedi, "Hand gesture recognition in real time for automotive interfaces: A multimodal vision-based approach and evaluations," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 6, pp. 2368–2377, 2014.

[66] K. Lai and S. N. Yanushkevich, "Cnn+rnn depth and skeleton based dynamic hand gesture recognition," in *2018 24th International Conference on Pattern Recognition (ICPR)*, 2018, pp. 3451–3456.

[67] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS'14. Cambridge, MA, USA: MIT Press, 2014, p. 568576.

[68] R. E. Nogales and M. E. Benalczar, "Hand gesture recognition using machine learning and infrared information: a systematic literature review," *International Journal of Machine Learning and Cybernetics*, vol. 12, pp. 2859–2886, 2021. [Online]. Available: https://doi.org/10.1007/s13042-021-01372-y

[69] M. Hearst, S. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intelligent Systems and their Applications*, vol. 13, no. 4, pp. 18–28, 1998.

[70] D.-Y. Hsiao, M. Sun, C. Ballweber, S. Cooper, and Z. Popović, *Proactive Sensing for Improving Hand Pose Estimation.* New York, NY,

USA: Association for Computing Machinery, 2016, p. 23482352. [Online]. Available: https://doi.org/10.1145/2858036.2858587

[71] H. G. Doan, H. Vu, and T. H. Tran, "Recognition of hand gestures from cyclic hand movements using spatial-temporal features," in *Proceedings of the Sixth International Symposium on Information and Communication Technology*, ser. SoICT 2015. New York, NY, USA: Association for Computing Machinery, 2015, p. 260267. [Online]. Available: https://doi.org/10.1145/2833258.2833301

[72] S. Ameur, A. B. Khalifa, and M. S. Bouhlel, "A comprehensive leap motion database for hand gesture recognition," in *2016 7th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT)*, 2016, pp. 514–519.

[73] N. H. Dardas and N. D. Georganas, "Real-time hand gesture detection and recognition using bag-of-features and support vector machine techniques," *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 11, pp. 3592–3607, 2011.

[74] N. Baranwal and G. C. Nandi, "An efficient gesture based humanoid learning using wavelet descriptor and mfcc techniques," *International Journal of Machine Learning and Cybernetics*, vol. 4, no. 2017, 2016.

[75] D.-Y. Hsiao, M. Sun, C. Ballweber, S. Cooper, and Z. Popović, *Proactive Sensing for Improving Hand Pose Estimation*. New York, NY, USA: Association for Computing Machinery, 2016, p. 23482352. [Online]. Available: https://doi.org/10.1145/2858036.2858587

[76] A. Clark and D. Moodley, "A system for a hand gesture-manipulated virtual reality environment," in *Proceedings of the Annual Conference of the South African Institute of Computer Scientists and Information Technologists*, ser. SAICSIT '16. New York, NY, USA: Association for Computing Machinery, 2016. [Online]. Available: https://doi.org/10.1145/2987491.2987511

[77] B. Gupta, P. Shukla, and A. Mittal, "K-nearest correlated neighbor classification for indian sign language gesture recognition using feature fusion," in *2016 International Conference on Computer Communication and Informatics (ICCCI)*, 2016, pp. 1–5.

[78] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651–666, 2010, award winning papers

from the 19th International Conference on Pattern Recognition (ICPR). [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167865509002323

[79] T. Starner and A. Pentland, "Real-time american sign language recognition from video using hidden markov models," 1995.

[80] J. Lee-Ferng, J. Ruiz-del Solar, R. Verschae, and M. Correa, "Dynamic gesture recognition for human robot interaction," in *2009 6th Latin American Robotics Symposium (LARS 2009)*, 2009, pp. 1–8.

[81] P. Senin, "Dynamic time warping algorithm review," 2008.

[82] G. Plouffe and A.-M. Cretu, "Static and dynamic hand gesture recognition in depth data using dynamic time warping," *IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 2, pp. 305–316, 2016.

[83] H. Cheng, Z. Dai, Z. Liu, and Y. Zhao, "An image-to-class dynamic time warping approach for both 3d static and trajectory hand gesture recognition," *Pattern Recognition*, vol. 55, pp. 137–147, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320316000157

[84] Y. Ye and P. Nurmi, "Gestimator: Shape and stroke similarity based gesture recognition," in *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, ser. ICMI '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 219226. [Online]. Available: https://doi.org/10.1145/2818346.2820734

[85] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[86] W. Tao, M. C. Leu, and Z. Yin, "American sign language alphabet recognition using convolutional neural networks with multiview augmentation and inference fusion," *Engineering Applications of Artificial Intelligence*, vol. 76, pp. 202–213, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0952197618301921

[87] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 4489–4497.

[88] C. R. Naguri and R. C. Bunescu, "Recognition of dynamic hand gestures from 3d motion data using lstm and cnn architectures," in *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2017, pp. 1130–1133.

[89] F. L. Siena, B. Byrom, P. Watts, and P. Breedon, "Utilising the intel realsense camera for measuring health outcomes in clinical research," *Journal of medical systems*, vol. 42, no. 3, p. 53, 2018.

[90] Y. Park, J. Lee, and J. Bae, "Development of a wearable sensing glove for measuring the motion of fingers using linear potentiometers and flexible wires," *IEEE Transactions on Industrial Informatics*, vol. 11, pp. 198–206, 2015.

[91] J. Suarez and R. R. Murphy, "Hand gesture recognition with depth images: A review," in *2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*, 2012, pp. 411–417.

[92] W.-S. Lin, Y.-L. Wu, W.-C. Hung, and C.-Y. Tang, "A study of real-time hand gesture recognition using sift on binary images," in *Advances in Intelligent Systems and Applications - Volume 2*, J.-S. Pan, C.-N. Yang, and C.-C. Lin, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 235–246.

[93] T. DOrazio, R. Marani, V. Ren, and G. Cicirelli, "Recent trends in gesture recognition: how depth data has improved classicalapproaches," *Image and Vision Computing*, vol. 52, pp. 56–72, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0262885616300853

[94] C.-C. Wang and K.-C. Wang, *Hand Posture Recognition Using Adaboost with SIFT for Human Robot Interaction.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 317–329. [Online]. Available: https://doi.org/10.1007/978-3-540-76729-9_25

[95] Q. Chen, N. D. Georganas, and E. M. Petriu, "Real-time vision-based hand gesture recognition using haar-like features," in *2007 IEEE Instrumentation Measurement Technology Conference IMTC 2007*, 2007, pp. 1–6.

[96] P. A. Viola and M. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, pp. 137–154, 2004.

[97] D. Lisin, M. Mattar, M. Blaschko, E. Learned-Miller, and M. Benfield, "Combining local and global image features for object class recognition,"

in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops*, 2005, pp. 47–47.

[98] K. Siddiqi, S. Bouix, A. Tannenbaum, and S. Zucker, "Hamilton-jacobi skeletons," *International Journal of Computer Vision*, vol. 48, pp. 215–231, 2004.

[99] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, pp. 509–522, 2002.

[100] H. Ling and D. W. Jacobs, "Shape classification using the inner-distance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 2, pp. 286–299, 2007.

[101] K. Matusiak, P. Skulimowski, and P. Strumillo, "Depth-based descriptor for matching keypoints in 3d scenes," *International Journal of Electronics and Telecommunications*, vol. 64, pp. 299–306, 2018.

[102] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, 1986.

[103] X. Meng, Y. Yin, G. Yang, and X. Xi, "Retinal identification based on an improved circular gabor filter and scale invariant feature transform," *Sensors*, vol. 13, no. 7, pp. 9248–9266, 2013. [Online]. Available: https://www.mdpi.com/1424-8220/13/7/9248

[104] S. Zhao, X. Xu, W. Zheng, and J. Ling, "Registration of depth image and color image based on harris-sift," in *Proceedings of the 2012 Second International Conference on Electric Information and Control Engineering - Volume 01*, ser. ICEICE '12. USA: IEEE Computer Society, 2012, p. 10311035.

[105] Y. Ke and R. Sukthankar, "Pca-sift: a more distinctive representation for local image descriptors," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, vol. 2, 2004, pp. II–II.

[106] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008, similarity Matching in Computer Vision and Multimedia. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1077314207001555

[107] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005.

[108] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," in *Computer Vision – ECCV 2010*, K. Daniilidis, P. Maragos, and N. Paragios, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 778–792.

[109] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International Conference on Computer Vision*, 2011, pp. 2564–2571.

[110] E. Karami, S. Prasad, and M. S. Shehata, "Image matching using sift, surf, BRIEF and ORB: performance comparison for distorted images," *CoRR*, vol. abs/1710.02726, 2017. [Online]. Available: http://arxiv.org/abs/1710.02726

[111] M. Karpushin, G. Valenzise, and F. Dufaux, "Keypoint detection in rgbd images based on an anisotropic scale space," *IEEE Transactions on Multimedia*, vol. 18, no. 9, pp. 1762–1771, 2016.

[112] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2, pp. 2169–2178, 2006.

[113] D. A. T. Leite, J. C. Duarte, J. C. Oliveira, V. De Almeida Thomaz, and G. A. Giraldi, "A system to interact with cave applications using hand gesture recognition from depth data," in *2014 XVI Symposium on Virtual and Augmented Reality*, 2014, pp. 246–253.

[114] Y. Li, "Hand gesture recognition using kinect," in *2012 IEEE International Conference on Computer Science and Automation Engineering*, 2012, pp. 196–199.

[115] H. Mahmud, M. K. Hasan, A.-A. Tariq, and M. Mottalib, "Hand gesture recognition using sift features on depth image." Venice,Italy: IARIA, April 2016, pp. 359–365.

[116] N. Cheggoju. Sift (scale invariant feature transform) algorithm.

[117] A. Mishra. Multi class support vector machine.

[118] A. Bosch, X. Muoz, and R. Mart, "Which is the best way to organize/classify images by content?" *Image and Vision Computing*, vol. 25, no. 6, pp. 778–791, 2007. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0262885606002253

[119] A. Wexelblat, "An approach to natural gesture in virtual environments," *ACM Transactions on Computer-Human Interaction*, vol. 2, no. 3, pp. 179–200, sep 1995.

[120] A. Priya, S. Mishra, S. Raj, S. Mandal, and S. Datta, "Online and offline character recognition: A survey," in *2016 International Conference on Communication and Signal Processing (ICCSP)*. IEEE, apr 2016.

[121] S. Agrawal, I. Constandache, S. Gaonkar, R. Roy Choudhury, K. Caves, and F. DeRuyter, "Using mobile phones to write in air," in *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '11. New York, NY, USA: ACM, 2011, pp. 15–28.

[122] euronews Knowledge. Future of texting: writing in the air! Youtube. [Online]. Available: https://www.youtube.com/watch?v=XMU4zh083l4

[123] Y. Yin, L. Xie, T. Gu, Y. Lu, and S. Lu, "Aircontour: Building contour-based model for in-air writing gesture recognition," *ACM Trans. Sen. Netw.*, vol. 15, no. 4, pp. 44:1–44:25, Oct. 2019. [Online]. Available: http://doi.acm.org/10.1145/3343855

[124] D. H. Kim, H. I. Choi, and J. H. Kim, "3d space handwriting recognition with ligature model," in *Proceedings of the Third International Conference on Ubiquitous Computing Systems*, ser. UCS'06. Berlin, Heidelberg: Springer-Verlag, 2006, pp. 41–56.

[125] R. Mullin, "Time warps, string edits, and macromolecules: The theory and practice of sequence comparison. edited by d. sankoff and j. b. kruskal. addison-wesley publishing company, inc., advanced book program, reading, mass., don mills, ontario, 1983. 300 pp. u. s. $31.95. ISBN 0-201-07809-0," *Canadian Journal of Statistics*, vol. 13, no. 2, pp. 167–168, jun 1985.

[126] H. Mahmud, M. K. Hasan, Abdullah-Al-Tariq, M. H. Kabir, and M. A. Mottalib, "Recognition of symbolic gestures using depth information," *Advances in Human-Computer Interaction*, vol. 2018, no. 1069823, 2018.

[127] M. Thomas, "A role for air writing in second-language learners acquisition of japanese in the age of the word processor," *Journal of Japanese Linguistics*, vol. 30, no. 1, pp. 86–106, 2014.

[128] J. F. Kenney and E. S. Keeping, "Moving averages," pp. 221–223, 1962.

[129] R. Islam, H. Mahmud, M. K. Hasan, and H. A. Rubaiyeat, "Alphabet recognition in air writing using depth information," *The Ninth International Conference on Advances in Computer-Human Interactions*, pp. 299–301, April 2016.

[130] Y.-S. Jeong, M. K. Jeong, and O. A. Omitaomu, "Weighted dynamic time warping for time series classification," *Pattern recognition*, vol. 44, no. 9, pp. 2231–2240, 2011.

[131] A. Jalalian and S. K. Chalup, "Gdtw-p-svms: Variable-length time series analysis using support vector machines," *Neurocomputing*, vol. 99, pp. 270 – 282, 2013.

[132] D. Freeman and D. B. Barrentine, "Method and system for operating a multi-function portable electronic device using voice-activation," Mar. 10 2015, uS Patent 8,977,255.

[133] N. Sreekanth and N. Narayanan, "Dynamic gesture recognitiona machine vision based approach," in *Proceedings of the International Conference on Signal, Networks, Computing, and Systems*. Springer, 2017, pp. 105–115.

[134] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *ACM SIGKDD explorations newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[135] H. Mahmud, R. Islam, and M. K. Hasan, "On-air english capital alphabet (eca) recognition using depth information," *The Visual Computer*, vol. 2021, 2021. [Online]. Available: https://link.springer.com/article/10.1007%2Fs00371-021-02065-x

[136] J. C. Nez, R. Cabido, J. J. Pantrigo, A. S. Montemayor, and J. F. Vlez, "Convolutional neural networks and long short-term memory for skeleton-based human activity and hand gesture recognition," *Pattern Recognition*, vol. 76, pp. 80–94, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320317304405

[137] Y. Iwai, K. Watanabe, Y. Yagi, and M. Yachida, "Gesture recognition by using colored gloves," in *1996 IEEE International Conference on Systems, Man and Cybernetics. Information Intelligence and Systems (Cat. No. 96CH35929)*, vol. 1. IEEE, 1996, pp. 76–81.

[138] K. Lai and S. N. Yanushkevich, "Cnn+rnn depth and skeleton based dynamic hand gesture recognition," in *2018 24th International Conference on Pattern Recognition (ICPR)*, 2018, pp. 3451–3456.

[139] P. Wang, W. Li, Z. Gao, J. Zhang, C. Tang, and P. Ogunbona, "Deep convolutional neural networks for action recognition using depth map sequences," *ArXiv*, vol. abs/1501.04686, 2015.

[140] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, S. A. McIlraith and K. Q. Weinberger, Eds. AAAI Press, 2018, pp. 7444–7452. [Online]. Available: https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17135

[141] Y. Zhu, Z. Lan, S. Newsam, and A. G. Hauptmann, "Hidden Two-Stream Convolutional Networks for Action Recognition," *arXiv preprint arXiv:1704.00389*, 2017.

[142] Q. D. Smedt, H. Wannous, J.-P. Vandeborre, J. Guerry, B. L. Saux, and D. Filliat, "3D Hand Gesture Recognition Using a Depth and Skeletal Dataset," in *Eurographics Workshop on 3D Object Retrieval*, I. Pratikakis, F. Dupont, and M. Ovsjanikov, Eds. The Eurographics Association, 2017.

[143] Q. De Smedt, H. Wannous, and J.-P. Vandeborre, "Skeleton-based dynamic hand gesture recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2016, pp. 1206–1214.

[144] K. Lai and S. N. Yanushkevich, "An ensemble of knowledge sharing models for dynamic hand gesture recognition," *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7, 2020.

[145] Y. Chen, L. Zhao, X. Peng, J. Yuan, and D. N. Metaxas, "Construct dynamic graphs for hand gesture recognition via spatial-temporal attention," 2019.