A LOW COST LEARNING BASED SIGN LANGUAGE RECOGNITION SYSTEM

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ABDULLAH HAKAN AKIŞ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

DECEMBER 2018

Approval of the thesis:

A LOW COST LEARNING BASED SIGN LANGUAGE RECOGNITION SYSTEM

submitted by **ABDULLAH HAKAN AKIŞ** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar _____
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Tolga Çiloğlu _____
Head of Department, **Electrical and Electronics Engineering**

Prof. Dr. Gözde Bozdağı Akar _____
Supervisor, **Electrical and Electronics Eng. Dept., METU**

Examining Committee Members:

Prof. Dr. İlkay Ulusoy
Electrical and Electronics Eng. Dept., METU _____

Prof. Dr. Gözde Bozdağı Akar
Electrical and Electronics Eng. Dept., METU _____

Prof. Dr. Ziya Telatar
Electrical and Electronics Eng. Dept., Ankara University _____

Assoc. Prof. Dr. Cüneyt F. Bazlamaçcı
Electrical and Electronics Eng. Dept., METU _____

Assist. Prof. Dr. Elif Sürer
Informatics Institute, METU _____

Date: _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: ABDULLAH HAKAN AKIŞ

Signature :

ABSTRACT

A LOW COST LEARNING BASED SIGN LANGUAGE RECOGNITION SYSTEM

Akış, Abdullah Hakan
M.S., Department of Electrical and Electronics Engineering
Supervisor : Prof. Dr. Gözde Bozdağı Akar

December 2018, 87 pages

Sign Language Recognition (SLR) is an active area of research due to its important role in Human Computer Interaction (HCI). The aim of this work is to automatically recognize hand gestures consisting of the movement of hand, arm and fingers. To achieve this, we studied two different approaches, namely feature based recognition and Convolutional Neural Networks (CNN) based recognition. The first approach is based on segmentation, feature extraction and classification whereas the second one is based on segmentation and CNN which learns the signs from the image itself. In order to calculate the recognition rate of the systems, tests are conducted using eNTERFACE dataset of 8 American Sign Language (ASL) signs. Detailed analysis is done to evaluate each step of both approaches. Experimental results show that the feature based SLR system and CNN based SLR system achieved recognition rate of 95.31% and 93.12%, respectively. Experimental results also show that CNN based SLR system achieved recognition rate of 94.29% when data augmentation is used to increase the training dataset.

Keywords: Sign Language Recognition, Hand Gesture Recognition, Histogram of Oriented Gradients, Support Vector Machines, Convolutional Neural Networks

ÖZ

DÜŞÜK HESAP KARMAŞIKLIĞINA SAHİP ÖĞRENME TABANLI İŞARET DİLİ TANIMA SİSTEMİ

Akış, Abdullah Hakan
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi : Prof. Dr. Gözde Bozdağı Akar

Aralık 2018 , 87 sayfa

İşaret Dili Tanıma (İDT) insan bilgisayar iletişiminde önemli bir rol alması sebebiyle aktif bir araştırma konusudur. Bu çalışma el, kol ve parmak hareketlerinden oluşan el işaretlerini tanımayı amaçlamaktadır. Bu amaç için öznitelik tabanlı ve Konvolüsyonel Sinir Ağları (CNN) tabanlı iki İDT sistemi gerçeklenmiştir. Öznitelik tabanlı İDT sistemi el bölgesi bölütleme, öznitelik vektörlerini çıkarma, ve SVM kullanarak sınıflandırma aşamalarından oluşmaktadır. CNN tabanlı İDT sistemi ise el bölgesi bölütleme ve CNN aşamalarından oluşmaktadır. Sistemlerin başarımı 8 işaret dili jesti içeren eNTERFACE veritabanı ile test edilmiştir. İki sistemin her aşamasını değerlendirmek için detaylı analiz yapılmıştır. Öznitelik tabanlı sistem ve CNN tabanlı sistem ile sırasıyla %95.31 ve %93.12 tanıma oranı elde edilmiştir. Veritabanı büyüklüğünü artırmak için veri artırma yöntemi kullanıldığında CNN tabanlı sistemin tanıma yüzdesi %94.29'a yükselmiştir.

Anahtar Kelimeler: İşaret Dili Tanıma, El İşareti Tanıma

To my mother...

ACKNOWLEDGMENTS

I wish to express my sincere appreciation to Assoc. Prof. Dr. Mehmet Mete Bulut for his valuable guidance and support. It was an honour for me to work with him. We will always remember you. Rest in peace.

I also wish to express my sincere appreciation and thanks to my supervisor Prof. Dr. Gözde Bozdağı Akar for her precious guidance, support, valuable advices and encouragement throughout my study.

I would also like to thank to my employer ASELSAN and my colleagues, especially Dr. Alper Sinan Akyürek for their support.

I want to thank my wife Selin for her support in every step of my life.

Finally, for their valuable support and precious love, I am grateful to my family.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

ANN	Artificial Neural Network
ASL	American Sign Language
BSL	Bangla Sign Language
CNN	Convolutional Neural Networks
DOG	Difference of Gaussian
EOH	Edge Oriented Histogram
FSM	Finite State Machine
FD	Fourier Descriptor
FCM	Fuzzy C-Means
GMM	Gaussian Mixture Model
GPU	Graphics Processing Unit
HMM	Hidden Markov Model
HOG	Histogram of Oriented Gradients
HCI	Human Computer Interaction
HRI	Human Robot Interaction
KNN	K-Nearest Neighbor
LOG	Laplacian of Gaussian
LRB	Left Right Banded
ISL	Indian Sign Language
SIFT	Scale Invariant Feature Transform
SLR	Sign Language Recognition
SURF	Speeded Up Robust Feature
SVM	Support Vector Machines
TSL	Thai Sign Language

CHAPTER 1

INTRODUCTION

1.1 Sign Language Recognition

Sign language is the hand gesture based visual language used by hearing-impaired people to communicate with other people. The sign language consists of the coordinated movement of different parts of our body [7]. These are hands, face and the body. Depending on the sign, single hand posture or combination of hand movement, face expression and body posture can play important roles in the sign language. Sign Language Recognition (SLR) is the method of translation of the systematic and coordinated movements of one's body into lingual or textual phrases. Sign Language Recognition (SLR) is an active area of research due to its important role in Human Computer Interaction (HCI) or Human Robot Interaction (HRI).

The most important channel in sign language is hand gestures. Hand gestures are the meaningful body motions consisting of the movements of hand, arm and fingers. Two main types of hand gestures exist: A static gesture and a dynamic gesture. Static gesture recognition focuses on the posture, shape of the hand, while dynamic gesture recognition in constract relies on the movement of the hands temporally [8].

According to data capture methods, SLR systems can be categorized into three main branches: Sensor based, Vision based and Depth based approaches. In these approaches, hand movements are captured by an external sensor connected to signer's body, a camera, a depth sensor, respectively.

1.2 Scope and Outline of the Thesis

In this work, our aim is to design sign language recognition system to recognize isolated signs of sign language and evaluate different approaches on sign language recognition and compare both the computational cost and the performances of feature based and CNN based approaches. We decided to use vision based approaches because they require only an inexpensive camera and provide more convenient and user friendly interaction. We only focus on hand gestures since hands are most important and dominant channels in sign language. Face expressions and body movements are out of scope of this thesis.

Although we implemented SLR systems on a desktop computer, these systems are aimed to be able to work on mobile platform which has computational and memory constraints. This limits the type of algorithms that can be used for our purposes. Our main objective is to obtain the highest possible accuracy, while keeping the cost and complexity to a minimum.

Signers wear glove with different colors while performing sign in our work. Because of this, segmentation problem is more simple compared to non-glove based segmentation and a simple and low cost hand segmentation algorithm is used. This enables us to evaluate only the recognition performance of the systems.

Two sign language recognition systems, which are feature based SLR system and Convolutional Neural Networks (CNN) based SLR system, are presented in this work. In feature based SLR system, we start with a preprocessing, where the hands are segmented from the background. After hand segmentation, features are extracted by using Histogram of Oriented Gradients (HOG). Finally, SVM classifier is used to classify feature vectors. In CNN based SLR system, hands are segmented as in the feature based SLR system. After hand segmentation, CNN is used to classify segmented hand images.

Many works in the literature are signer and dataset dependent. Our aim is to implement a system which is not dependent on the signer and specific set of gestures used for training and testing. In our work, in order to evaluate the performance of implemented SLR systems, American Sign Language (ASL) dataset consisting of 8 signs

is used. However, systems could be used with other datasets which are based on hand gestures.

Chapter 2 consists of a literature survey on sign language recognition. In Chapter 3, the implemented SLR systems are presented in detail. Chapter 4 provides experimental evaluation of the implemented schemes and their performance comparison. Chapter 5 concludes this work along with future discussion.

CHAPTER 2

LITERATURE SURVEY

2.1 Sign Language Recognition Approaches

Sign language recognition systems focus on hands movements, face expression and the body posture, while hand gesture recognition systems rely only on the movements of hands. Hand gesture recognition systems can be categorized into three main branches: Sensor based, Vision based and Depth based approaches.

2.1.1 Sensor Based Approaches

In this approach, hand movements are measured by an external device connected to signer's body. External device, which has comprehensive sensors on it, can easily sense palm of hand and fingers. This provides more accurate movement and posture detection, high processing speed, fast response compared to the other approaches. Even though these properties provide significant advantages to the algorithm designer, sensor based solutions have serious practical drawbacks especially from the user's perspective. The first one is that signer must wear external measurement device connected to computer by cables. This decreases the level of user friendliness of the system. The second drawback is that these devices are expensive to manifacture. The last drawback is that sensors require high quality calibration to sense data correctly [8] [9] [10] [11].

2.1.2 Vision Based Approaches

In this approach, hand movements are captured by camera. This method is more user friendly and convenient in the sense that it provides a natural interaction method compared to sensor based methods, since signer does not need any cumbersome equipment to wear. Only a camera is required to capture images. On the other hand, there are some challenges associated with this method. The first challenge is that locating hands and segmenting them from the background are complex tasks. Other skin-colored objects, lighting conditions and variations may affect the segmentation performance. A major disadvantage is the occlusion problem. Occlusion of parts of the signer's body must be dealt with for accuracy. In some of the vision based SLR systems, signer wears colored gloves in order to simplify the hand segmentation process. By the help of the colored gloves, segmentation errors resulting from other skin-colored objects are prevented. Also, segmentation process has lower computational cost than non-glove based systems. [9] [10] [7]

2.1.3 Depth Based Approaches

In this approach, depth images are used for hand gesture recognition. Cameras provide two-dimensional information on the captured space. To acquire data on the third axis, a second specialized equipment is necessary. There are two ways to acquire depth images. The first and the most frequently used one is by using depth measuring cameras such as: Microsoft Kinect, ASUS Xtion, Mesa SwissRanger. The other option is extracting depth information from stereo video cameras [11].

There are several advantages of depth based approaches. Other skin colored objects, lighting conditions and variations, complex background don't cause any major problems in hand segmentation and tracking due to the usage of depth information [10]. Without these challenges, hand segmentation and tracking can be done easily and accurately [8]. However, Depth cameras are much more expensive than basic cameras used in visual based approaches.

2.2 Sign Language Recognition

In general, sign language recognition is composed of three main steps: Segmentation, Feature Extraction, and Classification as shown in Figure 2.1. Firstly, hand segmentation techniques must be used on acquired data. In the second step, features that are describing the performed sign must be extracted from output data of the first step by using feature extraction techniques. In the classification step, developed model and algorithm classify the performed sign according to extracted features. In the following subsections, previous studies in the literature are given in details for each of the mentioned steps. Some of the SLR systems used in the literature review are summarized in in Table 2.1 and 2.2.

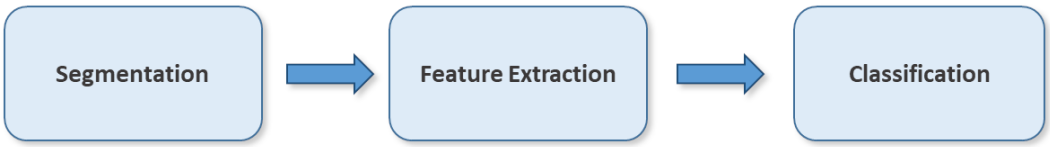

Figure 2.1: Hand Gesture Recognition Block Diagram

2.2.1 Data Acquisition and Hand Segmentation

Hand segmentation is the first step of sign language recognition. It is the process of segmenting the hands from the image. There are several methods, which are mainly composed of sensor based, vision based and depth based methods, used for hand segmentation in the literature.

In sensor based methods, hand shape and movements are measured by an external device connected to signer's body. The Polhemus sensor developed by Waldron and Kim [12] and PowerGlove used by Kadous [13] are the examples of sensors which measure location and orientation of the hands in the three dimensional space. In [14], in order to locate and track hands correctly, Vogler and Metaxas used magnetic sensor together with the vision based methods. Brashear et al. [15] developed a wearable device, which has hat mounted camera to acquire image from it and accelerometers on the hands.

In vision based methods, a camera is used to capture the signs. The captured image holds position, shape and motion features of the hands and fingers. After the capture step, segmentation process is required to segment hands from the background. Vision based hand segmentation methods can be categorized into two types: glove based and non-glove based methods.

In non-glove based methods, most of the proposed methods are based on skin color. HSV and YCbCr color space is known for better performance in hand image segmentation among other color spaces since they are robust to illumination changes [16] [17] [18]. In [19], Dawod et al. developed a new technique consisting of YCbCr conversion, CbCr mapping, shape enhancement and edge detection. In [20], Mo et al. proposes a new method which is the combination of improved Kalman filter and TSL skin color model. Firstly, other skin color objects are avoided by using improved Kalman filter model which estimates the center location of the motion. Then, TSL skin color model is used to segment the area which is predicted by Kalman filter. Finally, small holes are removed and boundary of hands are corrected by using image morphology processing to reduce the noise in the segmented image. In the work of Büyüksaraç et al. [21], Fuzzy C-Means clustering (FCM) and thresholding method is used for hand and face segmentation. FCM is a clustering technique which employs fuzzy partitioning, in an iterative manner. First, Fuzzy C-Means algorithm clusters the image according to the color information. Then, the mean value is chosen by thresholding according to the possible values a skin might have. In [22], Jin et al. used two-step hand segmentation technique. Firstly, Canny edge detection technique is applied on the image to detect the edges. Then, seeded region growing method is applied to segment hand region from the background. In [23], in order to segment hands from the background HSV thresholding and morphological operations are used. Firstly, HSV threshold values are determined based on hand skin color and HSV thresholding is applied to the image. Then, morphological operations such as dilation and erosion are applied on the image to remove the noise. In the work of Agrawal et al. [24], three hand segmentation algorithms are implemented; Gaussian mixture model (GMM) based segmentation algorithm, thresholding in YCbCr color space and Otsu algorithm. According to the evaluation results of segmentation algorithms, Otsu segmentation algorithm is found to have better segmentation results.

In glove based methods, signer wears colored gloves in order to simplify the hand segmentation process. By the help of the colored gloves, segmentation errors resulting from other skin-colored objects are prevented. In [25], two-step hand segmentation algorithm is used for colored glove segmentation. In the first step, 135 random snapshot images are taken from training videos. Ground truth images are created for each snapshot images. Then, hand pixel classifier is constructed by training with snapshot images and ground truth images to segment input images. In the second step, post processing operations are applied to the classified image, since the signer's clothing were classified as hand pixel at some images. In [26], signers wear multi colored gloves, in which fingers, palm of the hand and the back of the hand are colored differently in order to acquire hand pose and finger positions more precisely.

For depth based methods, hand segmentation part is not the most difficult and challenging step in gesture recognition. Hand segmentation algorithms are generally basic and easy to implement [11]. In the literature, most common way of segmentation is depth thresholding. In this method, hand is assumed to be the closest object to camera or within the predefined distance from the camera [27] [28] [29]. Another common method is the segmentation of the signer's body from depth image and distance of the hands are predicted by depth thresholding according to the signer's body [30]. Xia et al. [28] used real-time depth image data acquired by an active sensing hardware to recognize the 12 different gesture commands that are used to control the mobile robot.

After the release of the Kinect, which is a sensor device that provides color image capture, depth sensing and skeletal tracking, many sign language recognition systems leveraged it. In [31], Chen and Zhang used Kinect in order to acquire hand position and hand action information which are provided by Kinect SDK for Chinese sign language recognition. They showed that Kinect based method provides a relatively high recognition rate in real time compared to methods that use 2D cameras. Zhang et al. [32] used skeleton information from the video streams by Kinect. Four skeletons joints consisting of two hands and two elbows are used for isolated Chinese sign language recognition. They evaluated their approach with 450 phrases recorded by Kinect. According to the results, their method has a recognition accuracy of 88%. In [33], Kinect is also used to acquire 3D positions of the most important joints of

the body and hand regions of images. Their system had an accuracy rating of 89.33% and 98.33% recognition rates by using Kinect's skeleton features and skin color based features respectively.

2.2.2 Feature Extraction

Feature extraction is used to obtain meaningful information describing image through various image processing techniques. After this step, extracted feature vectors will be fed to the classification step. It is important to select appropriate feature vectors since the performance of the classifier is highly dependent on the features extracted. According to Sonkusare et al. [8], there are multiple criteria for features vector. The first one is that they must be rotation, translation and reflection invariant. The second criteria is that feature vectors should be easily computable and memory utilization of the feature vectors should be low. The last one is that the feature vectors representing similar features should not be used together. There are many feature vectors used for sign language recognition in the literature. Methods used in the literature can be mainly grouped as appearance based methods and model based methods. In appearance based methods, color, shape and texture features of images are extracted using image processing techniques. On the other hand, in model based methods sensor devices providing skeletal tracking or wearable sensor devices for acquiring hand trajectory and hand position information are used to model hands in the three dimensional space.

Appearance based feature descriptors can be divided into two parts: region based feature descriptors and texture based feature descriptors. Region based descriptors uses low level features such as area, bounding box, center of mass, width, and height. Since these features are highly dependent on the outer contour of hands, accurate segmentation must be done before extraction of low-level features. The main disadvantage is that these feature vectors only describe hand shape in general and do not provide inner hand shape information such as finger positions. Low-level features are used by many works in the literature. Bounding box, center of mass, aspect ratio, compactness, solidity, eccentricity, elongation, orientation are the mostly used structural shape descriptors in recent works [21] [25] [34] [33] [35]. In the work conducted

by Büyüksaraç et al. [21], due to the low resolution images and possible segmentation errors, bounding eclipse, bounding box and center of mass coordinates are chosen as feature vectors. In total, 23 features are extracted for each hand separately. In order to describe finger positions and inner hand shape details as well as outer contour information, texture based feature descriptors such as SIFT and HOG etc. are mostly used in the literature.

Scale Invariant Feature Transform (SIFT) [36] is a feature extraction method that is used to find and describe local features in images. Algorithm finds key points of the image that are invariant to scale and orientation. For each key points, location, scale and orientation are computed and stored in a feature vector. Many studies used SIFT feature descriptor to describe hand shape since SIFT is invariant to orientation, scale and varying illuminations [24] [37] [35]. Gupta et al. [37] uses SIFT to extract distinguishing invariant keypoints from input images for Indian Sign Language (ISL) alphabets. According to the results, SLR system is able to recognize sign alphabets with 78.84% average accuracy. [24] and [35] are the other examples of researches that are using SIFT feature descriptor.

Speeded Up Robust Feature (SURF) is also used to describe the hand shape and provides high recognition accuracy in [38] and [39]. SURF is developed by [40] to replace SIFT algorithm with its low computational cost. The main difference between SIFT and SURF algorithm is that SURF uses the Laplacian of Gaussian (LoG) with a filter box and SIFT uses Difference of Gaussian (DoG). By the help of this, calculation of SURF is faster compared to SIFT [22]. In [22], Jin et al. developed a mobile application which uses SURF features to recognize 16 static American Sign Language (ASL) words in real-time. Experimental results show that recognition rate achieved by using SIFT and SURF feature extraction methods are 97.13% and 92.25% respectively. Also, they demonstrated that SURF runs faster on the device compared to SIFT.

Histogram of Oriented Gradients (HOG), which is proposed by Dalal and Triggs [2], is a one of the most commonly used feature descriptor for object detection. HOG algorithm calculates the distribution of directions of gradients in the input image. Since the magnitude of gradients is larger around the corners and edges, shape can

be described accurately by HOG descriptor. It is used by many of the SLR systems in recent researches [37] [41] [42] [31] [35] [43] [44], because it is invariant to photometric and geometric transformations. [41] proposed Chinese Sign Language recognition through a Kinect sensor using HOG descriptor together with hand action features. They evaluated their system on dataset composed of 72 sign language words and achieved 89.8% average recognition rate.

Hu invariant moment, which consists of seven central moments, is also another feature extraction method preferred by many works since it provides feature vectors invariant to scale, translation and rotation [34] [45]. In [45], Hu invariant moments and hand orientation features are used together with local and global features. They achieved 91.20% recognition rate for finger-spelling recognition system. Edirisinghe et al. [46] claimed that Hu moments do not express detailed characteristic of image, instead giving a rough estimation of possible match. Because of that, they proposed feature vectors which are the combination of Hu moments, edge histogram descriptor and circularity shape parameter. With experimental results, they showed that proposed feature vectors are providing better recognition results and better system performance.

In [47], after segmenting hand region, Chen et al. use Fourier Descriptor (FD) to characterize spatial features. For extracting temporal features, they used motion analysis method. Then, spatial and temporal features are concatenated to construct feature vector.

Model based methods uses sensor devices that measures 3D information of important parts of the body. By the help of this, skeletal information, hand position, and joint positions can be gathered. In [13], Kadous uses features directly obtained by sensor device which measures position, shape and orientation of the hands in three dimensional space to recognize the sign language signs.

2.2.3 Classification

After the image segmentation and feature extraction, feature vectors are used as input to the classification. In this step, developed model and algorithm make a prediction

of the hand gesture performed. In the literature, there are several classification techniques such as Hidden Markov Model (HMM), Support Vector Machines (SVM), Artificial Neural Network (ANN), Finite State Machines (FSM), Convolutional Neural Networks (CNN), and K-Nearest Neighbor (KNN) Algorithm.

Hidden Markov Model (HMM) is a statistical model used to model spatio-temporal time series. It is known for having high recognition rate in dynamic gesture recognition [11] [48]. State transition from one state to another occurs probabilistically with time. States are not directly visible, but the outputs are visible. That is the difference of Hidden Markov model from the regular Markov model. HMM has three major solution steps: Evaluation, Training and Decoding. These solution steps are solved by Forward-Backward algorithm, Baum-Welch algorithm, and Viterbi algorithm respectively [7]. HMM has three topologies: Fully Connected, Left Right Model and Left Right Banded (LRB) model. Fully Connected model is a model that any state can be reached from any other state. In Left Right Model, each state can go back to itself and the next states. In Left Right Banded Model, each state can go back to itself and next state. For hand gesture recognition, Left Right Banded Model is used in the literature [21] [48] [49]. Number of states in HMM is decided depending on the complexity of the gesture [48]. For each type of gestures, HMM is trained separately. Each HMM block is connected in parallel to construct classification block. After each HMM block outputs probability of recognition, gesture with maximum probability is selected. By the usage of parallel classification of each gesture, adding new hand gesture or deleting existing hand gesture is possible without retraining the whole system [7]. In the work conducted by Büyüksaraç et al [21], HMM is used to classify structural shape descriptors consisting of bounding eclipse, bounding box and center of mass coordinates. They achieved 94.19% success rate for 8 American Sign Language set. In [48], HMM, which has LRB topology with 5 states, is used to classify feature vectors that is combination of Hu invariant moments and hand orientation. HMM is trained with 10 gestures for each gesture type. They achieved recognition rate of 94.33% for recognizing 6 gestures.

Artificial Neural Network (ANN) is brain-inspired system that is developed to simulate the way human brain learn. It is widely used in sign language recognition works [50] [35]. In [35], authors developed a recognition system for ISL numerals (0-9). In

the classification step, they used ANN to classify HOG feature vectors. According to the experimental results, the system provided a recognition rate of 99%. In [50], Rahagiyanto et al. developed a SLR system that uses data acquired from sensor providing accelerometer, gyroscope, orientation information and classifies these data by ANN. They tested the system with 26 classes of static and dynamic hand gestures. According to the results, an accuracy of 93.08% is achieved.

Finite State Machine (FSM) is a state based method to use in gesture recognition. Each gesture is defined to be an ordered sequence of states in spatio-temporal space. For every gesture, one FSM is trained and defined. When feature vectors from the feature extraction are given to recognizer, FSM for each gesture makes a decision whether to jump to next state or to stay at current state by analyzing the spatial and temporal features. When a FSM traversed all of the states and reached its final state at any time, gesture of that FSM is considered as a recognized gesture [51]. Verma and Dev [52] proposed a FSM and fuzzy logic based method for hand gesture recognition. In their work, extracted features from the images consists of 2D hand positions. Hand positions within time are clustered by Fuzzy C-mean clustering. Resulting clusters represent states of FSM that will be used in recognition. In [51], the training data consists of head and hand locations. They used k-mean clustering to cluster hand and head positions. And then, they created the structure of the FSM by manually defining temporal sequence of states from gesture examples. Experimental results show that they can successfully classify hand gestures such as drawing circle and drawing figure eight.

Convolutional Neural Networks (CNN) are multi-layered neural networks specified for recognizing visual patterns from image pixels. Several computer vision problems are solved using CNN with the recent improvements of Graphics Processing Unit (GPU). CNN is preferred for feature extraction and classification in recent researches [53] [54] [55] [56]. The main advantage is that CNN extracts most meaningful information automatically by its convolution and pooling layers. Feature extraction methods are not required to construct training data as required by other classification methods. In [53], authors proposed a novel CNN architecture which classify hand gestures from raw videos. They used RGB, depth and body joint images as input to CNN. As a result, they show that CNN has better recognition rate than HMM in their

case. In [54], authors proposed CNN based sign language recognition system. In this work, training data was created directly from sampling different frames of demonstration videos. They achieved 86% recognition rate for 6 sign language actions.

Support Vector Machine (SVM), which is based on Vapnik's theory [6], is a machine learning algorithm which is widely used for sign language recognition in the literature [18] [24] [34] [35] [57] [58]. Aim of the SVM's is finding hyperplane that separates and classifies a set of data with maximum distance to nearest data points of either class. In [57], the authors proposed sign language recognition algorithm for 10 static American Sign Language (ASL) sign. SVM classifier is used in the classification part. In experimental results, they achieved 96.15% for one-against-all SVM classifier and 99.23% for one-against-one SVM classifier. In [35], the authors implemented automatic SLR system for statics signs which consists of 10 Indian Sign Language (ISL) numerals (0-9). SVM classifier is used with different feature extraction methods like SIFT, HOG. They found that recognition system provides a recognition rate of 93%. In [58], recognition accuracies of different kernel types of SVM are investigated. According to the results, it is shown that RBF kernel provides better recognition rate than those with linear kernel in their case.

KNN (K-Nearest Neighbor) is another classification techniques is used for sign language recognition in the literature [59] [18] [37]. In [59], KNN is used for 10 static posture recognition. According to experimental results, KNN provides a recognition rate of 99.00% for static posture recognition. They also implemented the classification part with SVM and showed that average execution time of KNN is higher than average execution time of SVM.

Table 2.1: SLR Systems in the Literature

Work	**Feature Descriptor**	**Classification**	**Dataset**	**Recognition Rate (%)**
[24]	Shape Descriptors	SVM	36 ISL	41.2
	HOG			78.52
[34]	Hu invariant moment	SVM	60 ISL	40.36
	Structural Shape Descriptors			80.98
[58]	Depth Feature	SVM Linear Kernel	8 ASL	83.2
	Motion Feature			84.5
	Color Feature			93.5
	Depth Feature	SVM RBF Kernel		83.3
	Motion Feature			87.1
	Color Feature			93.2
[37]	HOG	KNN	26 ISL	80
	SIFT			78.84
[22]	SURF	SVM	16 ASL	97.13
[42]	HOG	SVM	16 BSL	86.53
[45]	Local and Global Shape Descriptors	SVM Linear Kernel	14 TSL	86.40
		SVM Polynomial Kernel		80
		SVM RBF Kernel		91.20
		SVM Sigmoid Kernel		54.67
[35]	HOG	SVM	10 ISL	96.20
[60]	Low-level hand shape features	HMM	8 ASL	94.19
[48]	Hu invariant moments and hand orientation	HMM	6 gestures	94.33

Table 2.2: SLR Systems in the Literature

Work	**Feature Descriptor**	**Classification**	**Dataset**	**Recognition Rate (%)**
[25]	Hand shape, hand motion, hand position features	HMM	8 ASL	97.8
[51]	Positions of head and hands	FSM	User Defined Gestures	-
[53]	CNN		25 SL Vocabularies	94.2
[54]	CNN		6 SL Signs	86
[57]	HOG and EOH	OAA SVM	10 ASL	96.15
		OAO SVM		99.23
[59]	Position of palm, Fingertip positions Hand direction, Velocity	KNN	10 Static Postures	99
[15]	Acceleration, hand shape	HMM	5 ASL	90.48

CHAPTER 3

EVALUATED APPROACHES FOR SIGN LANGUAGE RECOGNITION

3.1 Introduction

Traditional machine learning techniques and deep learning networks are the major approaches used for sign language recognition in the literature. Machine learning is a field of artificial intelligence which is able to learn from data by using statistical techniques. Deep learning have been recently used by many works for sign language recognition and it is a subfield of machine learning that learns high level features from the input data by its network inspired by structure of human brain. There are main dissimilarities between traditional machine learning and deep learning [61]. They are as follows:

1. Dataset Dependency: The main difference between traditional machine learning and deep learning is data dependency. With small amount of training data, traditional machine learning algorithms provide better accuracy than deep learning algorithms since deep learning algorithms require large amount of training data to learn most discriminant features. On the other hand, deep learning algorithms result in better accuracy with large amount of training data.
2. Hardware Requirement: Deep learning algorithms requires machines that have high computational power and large memory, since algorithms consists of multiple matrix multiplication operations. Generally, GPUs are used for handling these matrix operations. On the other hand, traditional machine learning algorithms are not computationally expensive and perform well in ordinary CPU.
3. Problem Solving Method: In traditional machine learning, the problem is divided into multiple steps such as hand detection, hand feature extraction, and

hand shape recognition etc. Different algorithms of each step solve their problems separately and they are combined to work together. In contrast, deep learning algorithms solve the problem end-to-end.

4. Feature Extraction: Traditional machine learning usually require feature extraction. Feature extraction is used to extract meaningful information from input data before classification part. Deep learning algorithms automated the feature extraction step by its deep layers. Data can be directly passed to the deep network for training and testing. There is no need for additional feature extraction process.
5. Computation Time: Training time of deep learning algorithms is usually quite longer than traditional machine learning algorithms because of complex deep layers of the network. On the other hand, deep learning algorithms usually require much less time for testing.

Our main objective is to obtain the highest possible accuracy, while keeping the cost and complexity to a minimum. Also, large dataset is not available for sign language since it is not easy to create such a large dataset. In the light of this objective and limitation, traditional machine learning algorithms are suitable for our system. In recent works, deep learning approaches has gained attention for sign language recognition. Because of this, we also evaluated and compared a deep learning based approach in our work. In this chapter, two SLR systems, which use traditional machine learning and deep learning approaches, are presented. Block diagrams of the systems are shown in Figure 3.1 and 3.2. The first one is feature based SLR system which is composed of three steps; hand segmentation, feature extraction and classification. The second one is CNN based SLR system which is composed of hand segmentation and CNN. In the following sections, these algorithms are explained in details by explaining each individual block of the systems.

3.2 Hand Segmentation

Hand segmentation is first step in sign language recognition systems. The aim of the hand segmentation is to extract valuable information from sampled image of sign video. In this work, the signers wear gloves with different colors when performing the

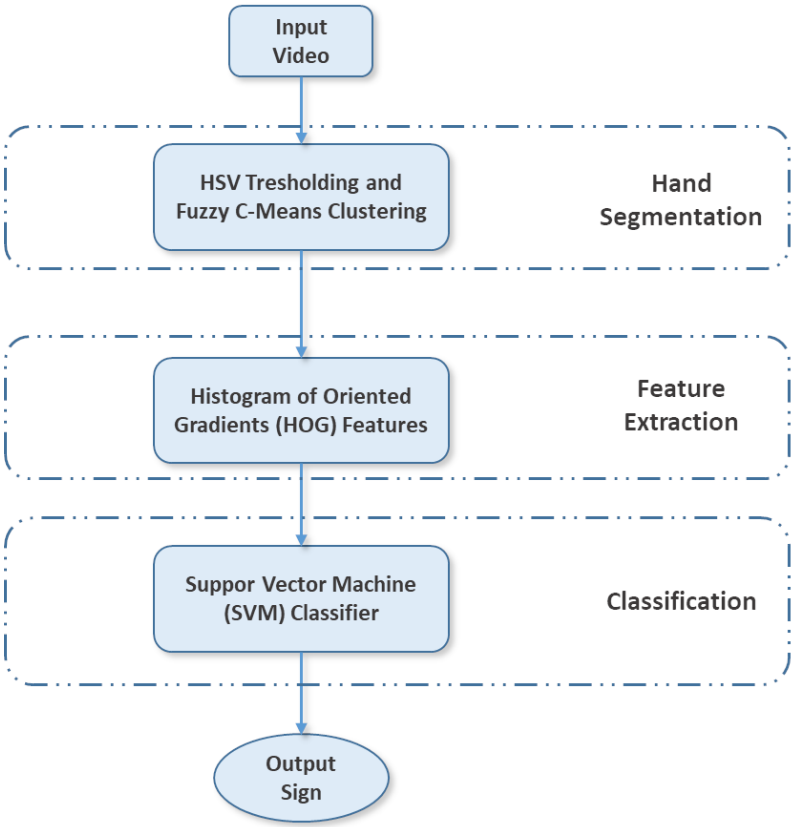

Figure 3.1: Block Diagram of Feature Based SLR System

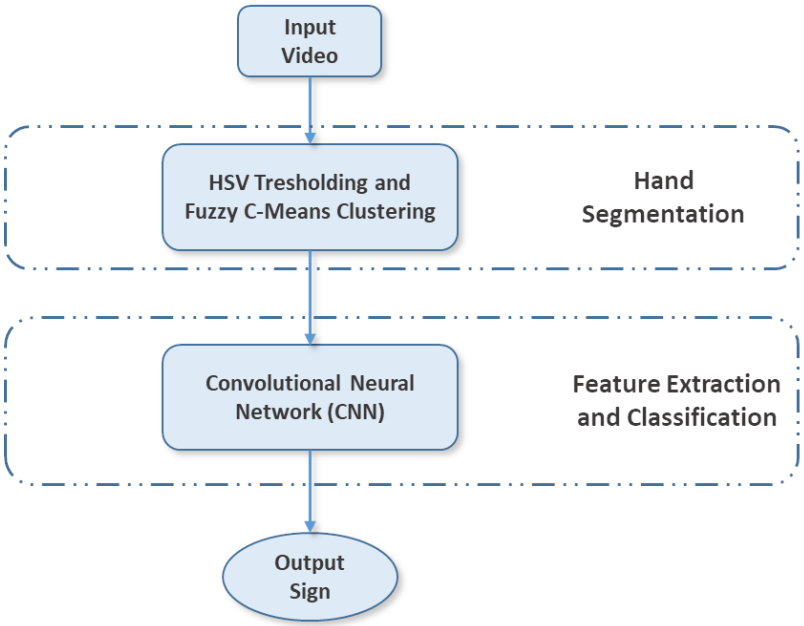

Figure 3.2: Block Diagram of CNN Based SLR System

signs. Thanks to the blue glove on the right hand and yellow glove on the left hand, segmentation problem is more simple compared to non-glove based segmentation. Because of this, simple hand segmentation algorithm, which has low computational cost, is used.

3.2.1 Hand Segmentation Algorithm

Hand segmentation algorithms, which are used by [62] and [60] are implemented and results achieved by using these segmentation algorithms are compared. Firstly, HSV color space based colored object segmentation is implemented as in Ganesan et al. [62]. In this implementation, histograms of hue, saturation and value components of colored hands are computed and plotted. By the help of the histograms, low and high thresholds for three color components are selected for sampled images from sign videos. The selected low and high thresholds are used in the threshold based segmentation of gloves. In the some of the cases, some part of the signer's arm are segmented as yellow glove since the color of the yellow glove and the color of the signer's arm fall in the same space in the HSV color space. In another implementation, Fuzzy C-Means Clustering is implemented on sampled images from sign video as Büyüksaraç [60] implemented. Gloves are clustered into the same groups with the t-shirts, arm and face of the signers in some sampled images. In these cases, the biggest contiguous part is not always the gloves. Because of these problems, a hand segmentation algorithm, which is combination of thresholding on HSV color space and Fuzzy C-Means Clustering, is implemented. The diagram for hand segmentation algorithm can be seen in Figure 3.3. Detailed algorithm is explained as follows.

RGB-HSV Color Space Conversion

There are several color spaces used in image processing such as RGB, HSV, YCbCr, YUV. According to the research [17], HSV color space is most suitable color space for segmentation problems. HSV color space is composed of three components; Hue, Saturation, Value. Figure 3.4 illustrates how hue, saturation and value components varies with the color. Hue corresponds to the color part. Saturation represents the amount of gray in the color. Value describes the brightness of the color. In this step, sampled image is converted into HSV color space.

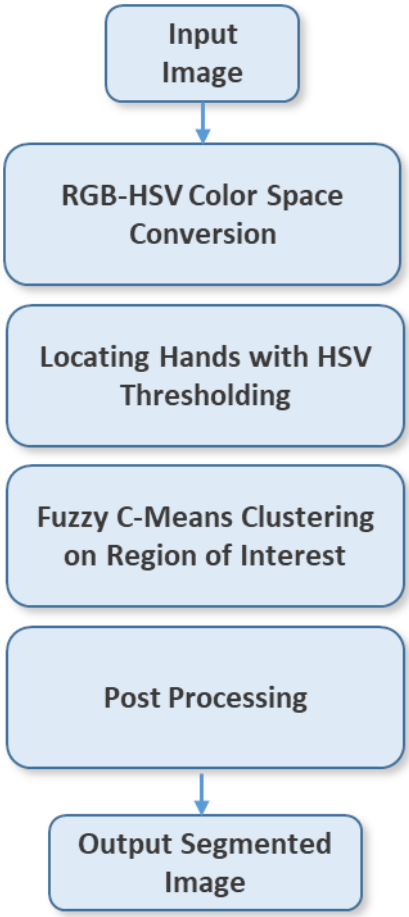

Figure 3.3: The Flow Diagram of the Hand Segmentation Algorithm

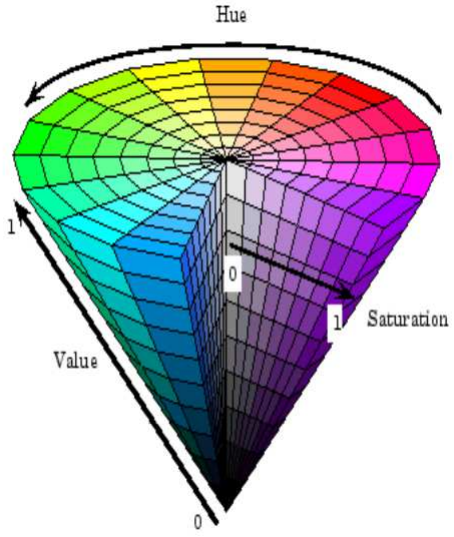

Figure 3.4: HSV Color Space [1]

Locating Hands with HSV Thresholding

In this step, yellow and blue gloves are located with HSV thresholding. Segmenting gloves from images with HSV thresholding does not provide good results, in some cases color of the yellow glove and the color of the signer's arm fall in the same region in HSV space. Because of that, only the region of interest is found by HSV thresholding in this step. In order to segment the gloves in the region of interest, Fuzzy C-Means Clustering is used in the next step. HSV threshold values are experimentally determined and listed in the Table 3.1 for each glove. After selecting the pixels that lie between the threshold values, connected components are found by using algorithm in [63]. Then, the largest connected component is selected and the center of the largest connected component is used as the center of region of interest in the next step. Example of this step can be seen in Figure 3.5.

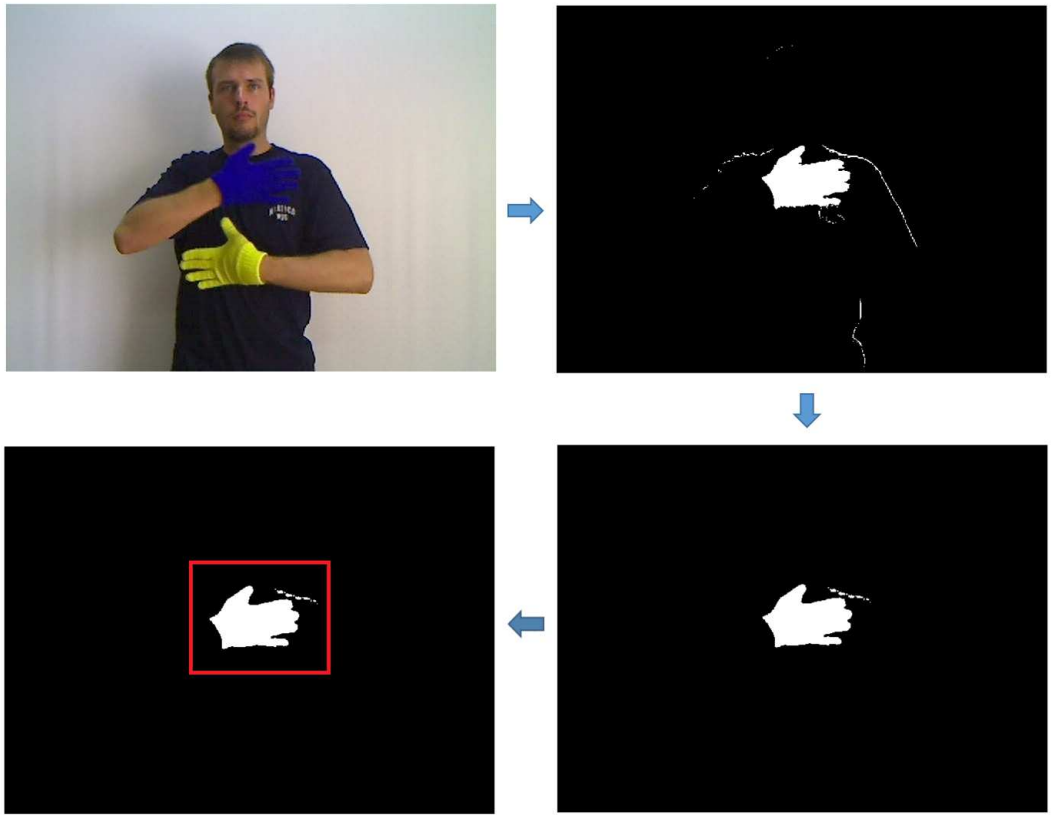

Figure 3.5: Example of Finding Region of Interest for Right Hand

Table 3.1: HSV Threshold Values for Gloves

	Threshold Values for Blue Glove	**Threshold Values for Yellow Glove**
Hue	0.56 to 0.76	0.04 to 0.32
Saturation	0.70 to 1.00	0.60 to 1.00
Value	-	0.40 to 1.00

Fuzzy C-Means Clustering on Region of Interest

In this step, in order to segment left and right hand from the background in the region of interest, Fuzzy C-Means algorithm is used as used by [60]. Fuzzy C-Means (FCM) is a clustering technique that finds degree of membership of each data points to the multiple clusters in an iterative manner [64]. FCM algorithm minimize the objective

function by (3.1).

$$J_m = \sum_{i=1}^{D} \sum_{j=1}^{N} \mu_{ij}^m \left\| x_i - c_j \right\|^2 \tag{3.1}$$

where

- D is the number of data points.
- N is the number of clusters.
- x_i is the ith data point.
- μ_{ij} is the degree of membership of x_i in the jth cluster.
- cj is the center of the jth clusters.
- m is fuzziness index.

In (3.1), cj and μ_{ij} are calculated by (3.2) and (3.3), respectively.

$$c_j = \frac{\sum_{i=1}^{D} \mu_{ij}^m x_i}{\sum_{i=1}^{D} \mu_{ij}^m} \tag{3.2}$$

$$\mu_{ij} = \frac{1}{\sum_{i\,=\,1}^{N} \left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}}} \tag{3.3}$$

Algorithm consists of the following steps:

1. The cluster membership values is randomly initilized.
2. The cluster centers c_j is calculated with (3.2)
3. The degree of memberships μ_{ij} is calculated with (3.3)
4. Objective function j_m is calculated according to (3.1)
5. Steps 2-4 is repeated until saddle point of j_m is achieved or the maximum number of iterations are done.

In our study, FCM algorithm is used for region of interest of each hand. Algorithm is used with the below parameters which are provided optimal results:

- Maximum number of iterations: 100
- Minimum improvement in objective function between two consecutive iteration: 0.001
- Number of clusters: 3

After the clusters are determined by the FCM algorithm, mean pixel values for each cluster is calculated. The clusters, which have mean pixel values in a certain HSV thresholds, is selected as hand pixel clusters. Threshold values are determined experimentally. Example FCM clustering for for left hand can be seen in Figure 3.6.

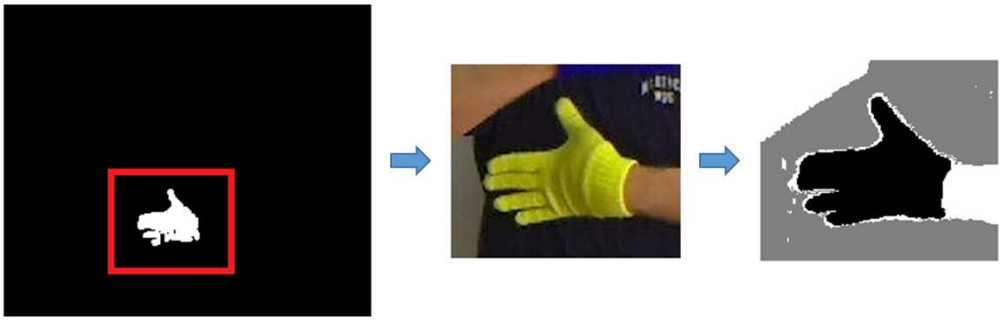

Figure 3.6: Example of Clustering for Left Hand

Post Processing

In order to correct the errors as a result of lighting conditions and clustering, two post processing operations are applied to the output of the clustering algorithm. The first one is removing pixels that do not belong to any hand. Connected components in the image are found by using algorithm in [63]. The largest one is selected as hand and the other ones are removed. The second post processing operation is filling the missing pixels in the hand region. Holes are eliminated by using the algorithm in [65]. Example post processing operation for removing pixels that do not belong to hand can be seen in Figure 3.7.

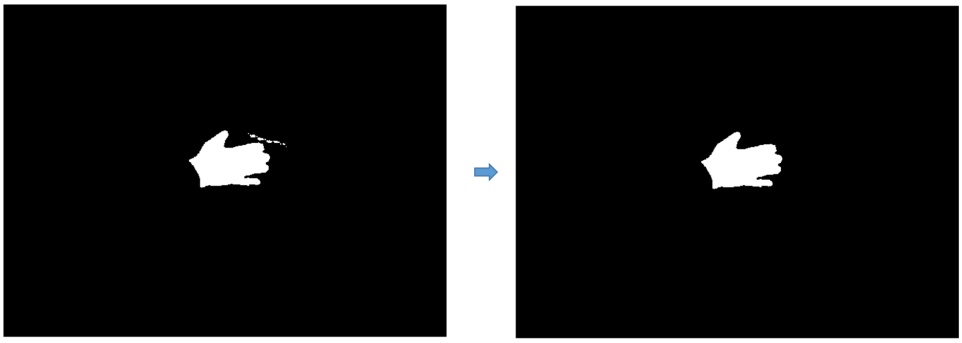

Figure 3.7: Example of Post Processing for Right Hand After Clustering

3.2.2 Results from the Dataset

Example hand segmentation process can be seen in Figure 3.8. Firstly, hands are located with low and high HSV thresholds. After region of interest is determined, each hand region is clustered with Fuzzy C-Means Clustering to segment hand pixels. Finally, post processing is applied to remove pixels that does not belong to any hand.

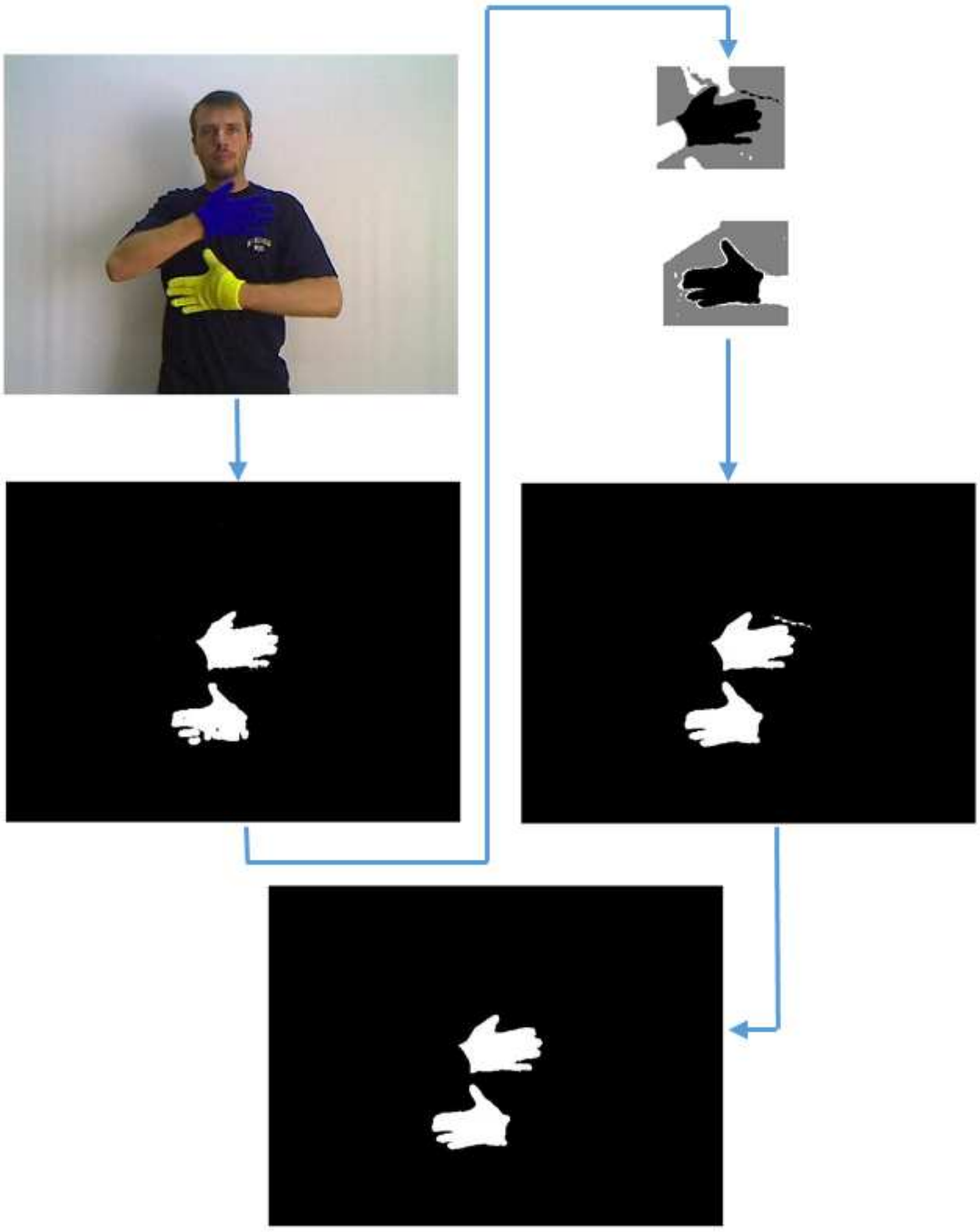

Figure 3.8: Example Hand Segmentation

3.3 Feature Extraction

3.3.1 Introduction

After image segmentation, feature extraction is used to remove unnecessary information from input data. Since body posture and face expressions are out of scope of this thesis, only hand features are extracted in this work. There are different feature extraction algorithms used in the literature. Methods used in the literature can be mainly grouped as appearance based methods and model based methods. In appearance based methods, color, shape and texture features of images are extracted using image processing techniques. On the other hand, in model based methods sensor devices providing skeletal tracking or wearable sensor devices for acquiring hand trajectory and hand position information are used to model hands in the three dimensional space. Our aim is to design an SLR system that works with a camera to capture video. Any sensor device that measures position, shape and orientation of the hands in three dimensional space is not used in our work due to cost and complexity of the system. Therefore, model based feature descriptors cannot be used in our system.

Performance of the recognition system is highly dependent on the features extracted. There are multiple criteria for a good feature descriptor. The first one is that feature vectors should be easily computable. Execution time is significantly important for real-time performance and better user experience. In order to be able to use the system in different platforms such as mobile phone etc. where the system has limited amount of computational power and memory, execution time should be low. The second one is that feature vectors must be scale, rotation and illumination invariant. In other words, feature vector must provide the same result for the same object with different rotation, scale, and illumination. In the sign language recognition, feature descriptor should produce the same output regardless of hand size, distance from the source, lighting condition and orientation of the hands. The last criterion for the feature vector is recognition accuracy which is demonstrated in the literature for the same problem.

By considering all criteria, Table 3.2 is prepared for most commonly used appearance based feature descriptors in the literature. Some of the works use region based

feature descriptors such as area, bounding box, center of mass etc.[21] [25] [34] [33] [35]. Since these feature vectors are extracted by using outer contour information, they do not provide inner hand shape details such as finger positions. These methods are computation efficient since they consist of simple calculations such as calculation of width, height, area of segmented hand images. On the other hand, they usually do not provide scale and orientation invariance. There are some works which use texture based feature descriptors to describe the hand shape in more details. SIFT, which finds key points of the image, is a feature extraction algorithm used to describe hand shape. It is used by some works since it is invariant to orientation, scale and illumination [24] [37] [35]. SURF, which is developed by [40] to replace SIFT algorithm with its low computational cost, is also used to describe hand shape in recent researches [38] [39]. SURF is invariant to scale, rotation and lightning changes. There are studies which demonstrated that SURF is more computational effienct and has slightly better recognition accuracy than SIFT [22] [66] [67]. HOG, which is proposed by [2], is one of the most commonly used feature desciptors for hand shape recogniton in recent researches [37] [41] [42] [31]. It is also invariant to scale and lightning changes. In some studies, experimental results showed that HOG provides more recognition accuracy than SIFT and region based feature desciptors for hand shape recognition [24] [37] [35]. In [68], comparative study is conducted on six feature extraction algorithms and experimental time measurements showed that HOG is less computational complex compared to SIFT and SURF. Hu invariant moments, which is known to be rotation and scale invariant, is also one of the most commonly used feature descriptors in the literature [34] [45]. In [45]. Although it has high recognition rate in some works, some recent experiments showed that region based features descriptors provides better recognition accuracy than hu invariant moments for hand shape recognition [34].

Table 3.2: Comparison of Feature Descriptors in the Literature

Feature Descriptor		**Computational Cost (levels represented by (●) according to the literature)**	**Scale Invariance (✓ / ×)**	**Rotation Invariance (✓ / ×)**	**Robustness to Illumination (✓ / × /)**	**Recognition Accuracy (levels represented by (●) according to the literature)**
Region Based Shape Desciptors	Ratio of hand pixels outside / Total hand pixels	●	✓	✓	n/a	●●
	Best fitting ellipse width		×	✓	n/a	
	Best fitting ellipse height		×	✓	n/a	
	Bounding box width		×	×	n/a	
	Bounding box height		×	×	n/a	
	Horizontal location of CoM		×	×	n/a	
	Vertical location of CoM		×	×	n/a	
HOG		●●	✓	×	✓	●●●●
SIFT		●●●	✓	✓	✓	●●
SURF		●●	✓	✓	✓	●●●
Hu Moment Invariants		●●	✓	✓	✓	●

Some of the works used multiple feature descriptors together [69] [24]. Multiple feature descriptors results in more recognition rate than single feature descriptor in most of the cases. However, computational cost of the multiple feature descriptors is also higher than single feature descriptor. Because of this, we decided to use a single feature descriptor for feature extraction step in our system. Appearance based feature descriptors in Table 3.2 have been taken into consideration for feature descriptor selection. We consider a mobile platform as our platform, therefore complexity is of high importance while maintaining a good level of accuracy. Combining these requirements, we decided to use the HOG feature descriptor, which is scale and illumination invariant, has low computational cost, and has high recognition accuracy for hand shape detection in the literature.

3.3.2 Histogram of Oriented Gradients (HOG) Features

Histogram of Oriented Gradients (HOG), which is proposed by Dalal and Triggs [2], is a feature descriptor used in computer vision and image processing. Distribution of directions of gradients are used as a feature in HOG feature descriptor. Since the magnitude of gradients is larger around the edges and corners than the other part of the image, HOG feature descriptor is useful for shape detection. HOG feature extraction algorithm is mainly composed of gradient computation, accumulating weighted votes for gradient orientation over spatial cells and normalization within block of cells as can be seen in Figure 3.9. In the first part, a filter with kernels given in Equation 3.4 is applied to image to calculate x and y derivatives. Then, these x and y derivatives are used in the computation of magnitude and orientation of gradients as given in Equation 3.5 and 3.6

$$D_x = \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} D_y = \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}^T \tag{3.4}$$

$$|G| = \sqrt{I_x^2 + I_y^2} \tag{3.5}$$

$$\theta = tan^{-1}\frac{I_x}{I_y} \tag{3.6}$$

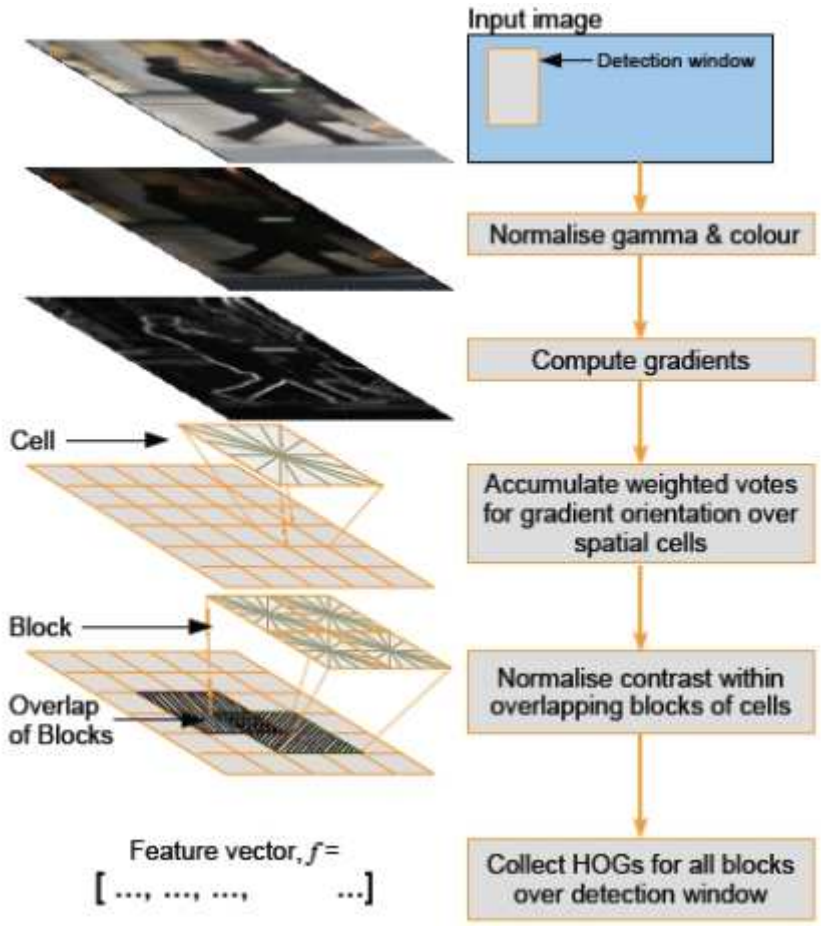

Figure 3.9: HOG Feature Extraction Process [2]

After gradient computation, image is divided into NxN pixel sub-images called cells which can either rectangular (R-HOG) or circular(C-HOG) as it seen in Figure 3.10. For each cell, histogram of oriented gradients are computed. Weighted votes, which are used according to the magnitude gradients calculated in the computation part, are accumulated into orientation bins for each direction to constitute a histogram.

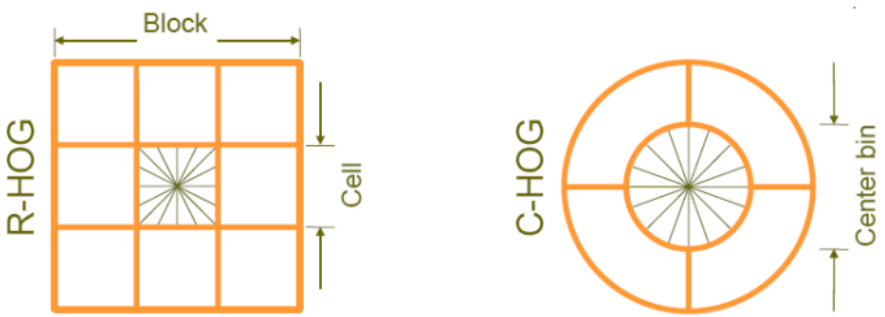

Figure 3.10: Cell and Block Geometries [2]

Due to the illumination changes, cells are required to be normalized. Adjacent cells are grouped to construct spatial region called blocks as in Figure 3.9. There are different normalization schemes available for block [57]:

- None: No Normalization used.
- $L_1 - norm = \dfrac{v}{\sqrt{\|v\|_1^2 + e^2}}$
- $L_2 - norm = \dfrac{v}{\sqrt{\|v\|_2^2 + e^2}}$

where v is non-normalized vector, $\|v\|_k$ is its k-norm for $k = 1, 2$, e is small constant.

When the all histograms of all cells in the image is calculated, all histograms are concatenated to construct descriptor vector of an image.

In Figure 3.11, example single hand HOG representation with cell size 2x2, 4x4, 8x8, 16x16, 32x32 can be seen. Arrows in the figure represents the gradient orientation. In our study, HOG feature descriptor extracted using below parameters which are provided optimal results:

- Number of pixels in cell: 4x4
- Number of cells in block: 2x2
- Number of bins: 9
- Number of overlapping cells between adjacent block: Block size/2
- Normalization scheme: L2-Norm

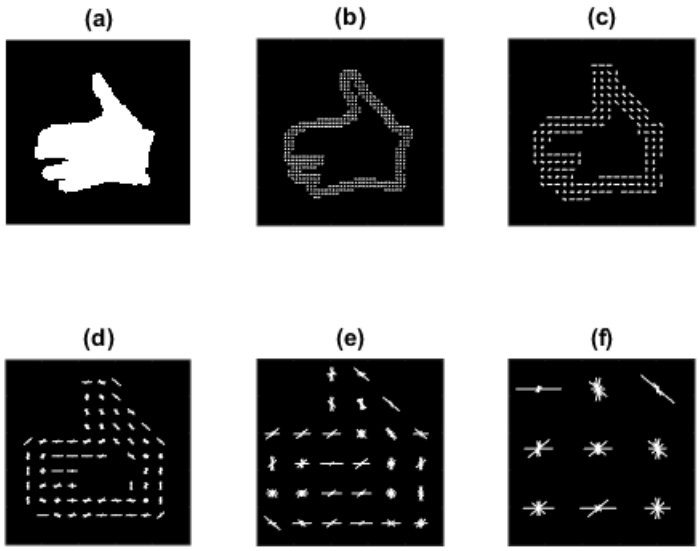

Figure 3.11: HOG Representations for Different Cell Size.

Since this study is dealing with dynamic hand gestures, HOG feature vectors are extracted for N frames of sign video. Then, HOG features for N frames are concatenated in order to constitute one feature vector for a dynamic gesture. Example HOG feature for dynamic gesture is shown in Figure 3.12. After building feature descriptor for a dynamic gesture, this feature vector will be fed to the recognition part.

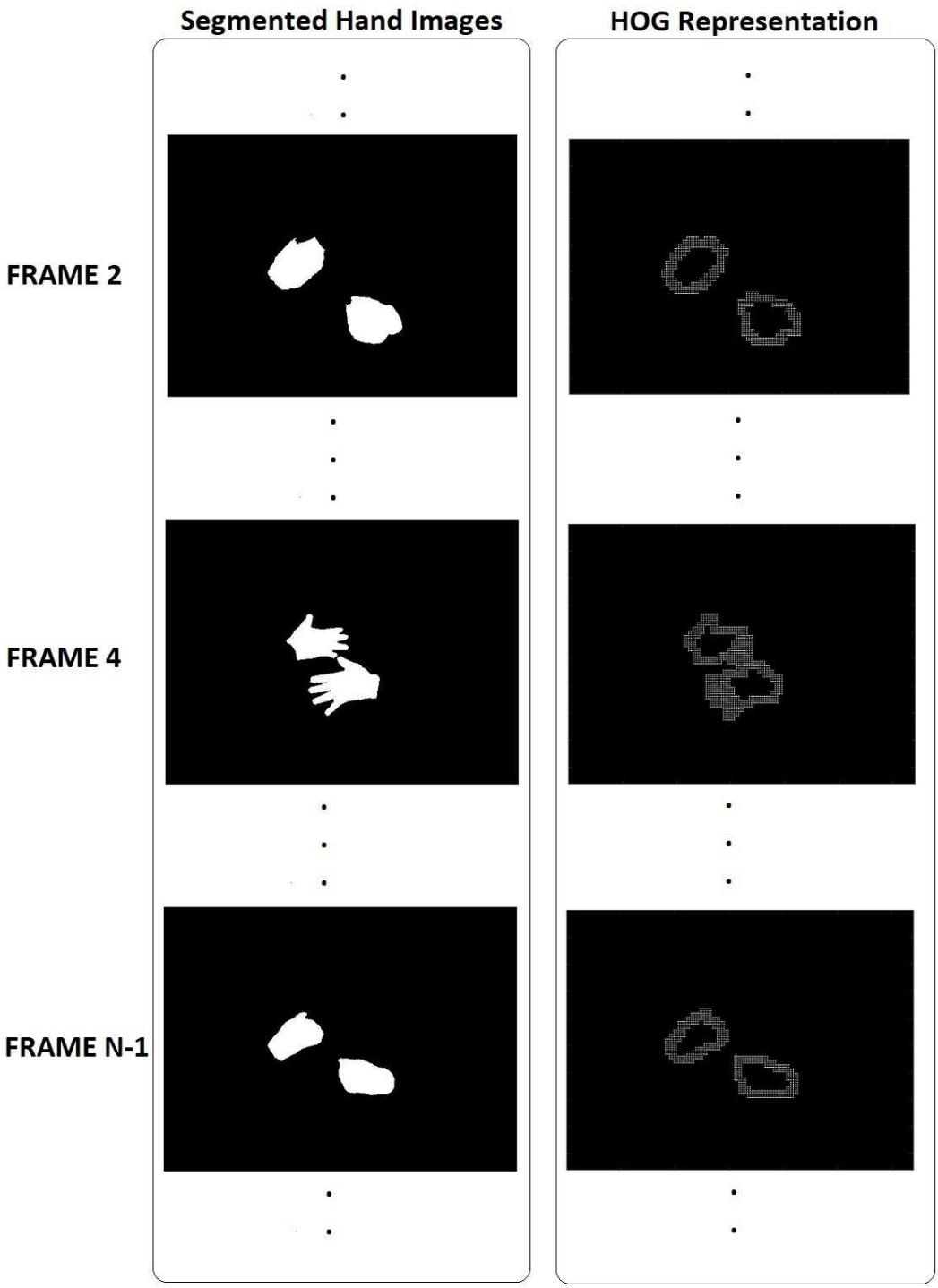

Figure 3.12: Segmented Hand Images and HOG Representations for N Frames.

3.4 Classification

3.4.1 Introduction

In the literature, there are several machine learning classification algorithms used for sign language recognition such as Hidden Markov Model (HMM), Support Vector Machines (SVM), Artificial Neural Networks (ANN), and K-Nearest Neighbor (KNN) Algorithm.

For choosing between classification algorithms, it is essential to compare the algorithms across multiple criteria. The first one is computation time. Time required to train and predict differ between algorithms. For some applications such as real-time applications, training and prediction speed are significantly important. The second one is number of parameter tuning needed for optimization. There are several parameters that affect the algorithm's behavior such as number of iteration, error tolerance etc. Finding a good combination of parameters is harder with large number of parameters since the time required to train algorithm increases exponentially with the number of parameters. The third one is training dataset size. Classifiers should achieve high accuracy with the number of training data that is available for the problem. The last criterion for classification algorithm is overall accuracy which is demonstrated in the literature the same problem.

By considering all criteria, we reviewed the most frequently used classification algorithms for sign language recognition in the literature. Some of the works uses KNN algorithm to classify sign language signs [59] [18] [37]. KNN is a simple algorithm that stores all available data and classifies new data based on a distance function. It requires large memory to use and it is computationally expensive because it stores almost all of the training data to use in prediction. The prediction time is dependent on the number of training data. As the number of training data increases, prediction gets slower [59]. Parameter optimization of KNN is easy since there is one parameter to optimize [70]. K value is usually tuned with cross validation techniques. In some studies, ANN is used for classification step of SLR systems [50] [35]. It is known for its high accuracy with large number of training data. It is not suitable for the classification problem with small datasets. There are lots of parameters of ANN

to optimize such as number of hidden layers, number of neurons in each layer etc. There is no specific method for finding optimum combinations of these parameters [71]. Although Neural Networks can take a long time to train since it calibrates linkage weights several times for each training data, prediction speed is significantly fast [72]. SVM is another classification technique used for SLR in some works [18] [24] [34] [35] [57] [58]. It results in high recognition rate with small datasets. It is not suitable to use with larger datasets since training time can be high [73] [74]. Kernel type, gamma and cost parameters must be optimized for better performance [58]. HMM, which is a statistical model to model spatio-temporal time series, is known for having high recognition rate with small datasets in dynamic gesture recognition [11] [48]. There are lots of unknown parameters of an HMM. In order to find the optimal values of an HMM with a set of feature vectors, the Baum Welch algorithm is used [60].

A large dataset is not available for sign language recognition and it is not easy to create such a large dataset. Also, we consider a mobile platform as our platform, therefore it is important to keep training and prediction time and computational complexity to a minimum while having the highest possible accuracy. Classification algorithms that we reviewed have been taken into consideration for classifier selection. With mentioned requirements, we decided to use SVM which provides high accuracy with small dataset, requires acceptable training time and provides fast prediction.

3.4.2 Support Vector Machines (SVM)

Support Vector Machine (SVM), which is based on Vapnik's theory [75], is a machine learning algorithm which is mostly used in classification problems. Aim of SVM is finding hyperplane that separates and classifies a set of data with maximum distance to nearest data points of either class. Finding the right hyperplane is crucial, because hyperplane with larger margin reduce the change of miss-classification. In Figure 3.13 (a), it can be seen that hyperplane has lower margin to red circle class, has higher margin to clue square class. Because of the unbalanced separation, it is not the optimum hyperplane. In Figure 3.13 (b), right hyperplane, which is separating classes with largest margin, for these two classes can be seen.

In some problems, two classes have linearly non-separable datasets. For this case, SVM has a solution called kernel trick. This technique converts linearly non-separable problem to linearly separable problem by transferring low dimensional input space to higher dimensional space until a separating hyperplane can be found [73] [74].

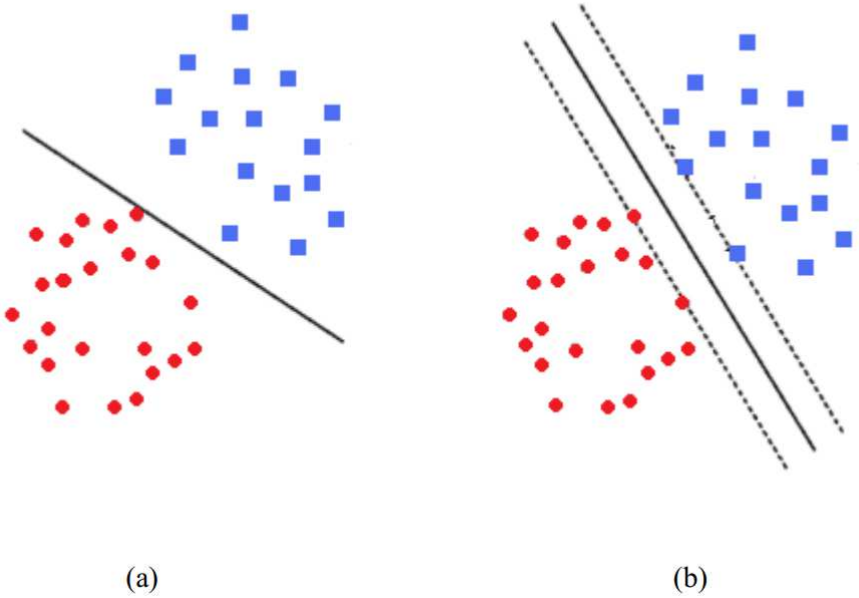

Figure 3.13: Non-optimal and Optimal Hyperplane for Classes [3]

Hyperplane can be formulated as in Equation 3.7, where β is weight vector and β_0 is the bias. The aim is to obtain β and β_0.

$$f(x) = \beta_0 + \beta^T x \tag{3.7}$$

As formulated in Equation 3.8, if hyperplane function is greater than or equal to to 1 it is decided that it belongs to one class. If hyperplane function is less than or equal to -1, it is decided that it belongs to other class.

$$\begin{aligned} y_i &= +1 \text{ if } f(x_i) \geq +1 \\ y_i &= -1 \text{ if } f(x_i) \leq +1 \; i = 1, ..., n \end{aligned} \tag{3.8}$$

The distance between a data point x and a hyperplane(β,β_0) is formulated as follows:

$$\text{distance} = \frac{\beta_0 + \beta^T}{\|\beta\|} \tag{3.9}$$

For the nearest point called support vector, distance equation in Equation 3.9 is given as Equation 3.10.

$$\text{distance}_{\text{support vector}} = \frac{\beta_0 + \beta^T}{\|\beta\|} = \frac{1}{\|\beta\|} \tag{3.10}$$

Weight vector β and bias β_0 for the optimal hyperplane can be obtained by solving optimization problem in Equation 3.11 by Lagrangian method. As a result, β can be recovered as given in Equation 3.12 where α_i is Lagrange multiplier.

$$\text{minimize } \frac{1}{2}\|\beta\|^2 \text{ subject to } y_i(\beta^T x_i + \beta_0) \geq 1 \; \forall i \tag{3.11}$$

$$\beta = \sum_{i=1}^{n} \alpha_i y_i x_i \tag{3.12}$$

Commonly used kernel functions are listed below:

- Linear kernel function
- Polynomial kernel function
- Radial basis kernel function

SVM is initially designed for binary classification. For multi-class classification, different algorithms are developed and proposed. Two of the mostly used methods are one-against-all (OAA) method and one-against-one (OAO) method [57]. Number of binary SVM classifier in one-against-all (OAA) method is equal to number of class in training set. For each class, corresponding binary classifier finds hyperplane between data points of corresponding class and the data points of the remaining classes. In

the decision step, decision is made only when testing data is recognized by only one SVM classifier. In Figure 3.14, example hyperplanes of OAA method can be seen. There are regions that are recognized by multiple classes. Recognizer cannot able to decide which class the input data belongs to. So, this results in poor classification performance. In one-against-one (OAO) method, there is a SVM classifier available for each possible pair of classes. Total number of SVM classifiers is equal to n*(n-1)/2. In algorithm, final recognition decision is made by selecting the class which is output of the majority of the pairwise classifiers. In Figure 3.15, it can be seen that uncovered and common region in input space is very small. Thus, this algorithm provides more accurate results compared to one-against-all (OAA) algorithm. However, it requires more training time than OAA algorithm since number of SVM classifiers increases exponentially as the number of classes increases.

In our feature based SLR system, C-Support Vector Classification (C-SVC), in which multiclass support is handled according to the one-against-one (OAO) method, is used for classification. SVM classifier is used with the below parameters:

- Type of SVM: C-SVC
- Kernel: Radial basis function $K(x^i, x^j) = e^{-\gamma \left\| x^{(i)} - x^{(j)} \right\|^2}, \gamma > 0$
- The parameter C of C-SVC: 10

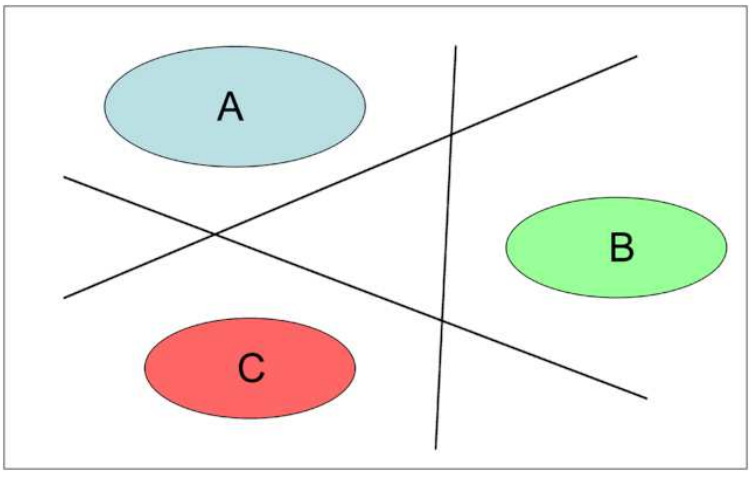

Figure 3.14: OAA Hyperplanes on an Example Problem [4]

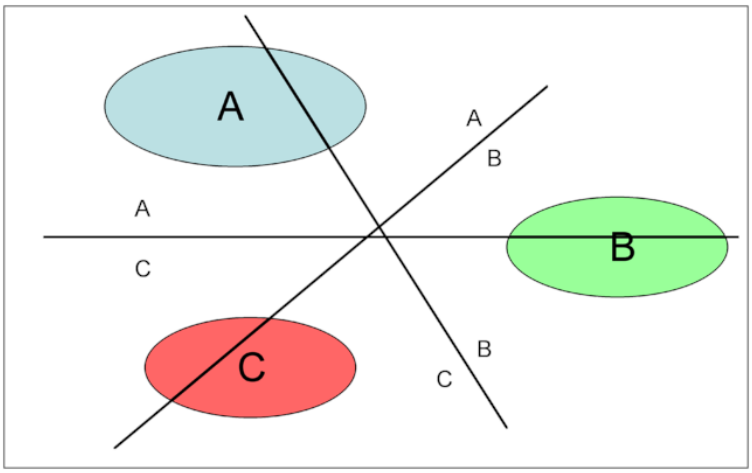

Figure 3.15: OAO Hyperplanes on an Example Problem [4]

3.5 Feature Extraction and Classification Based on CNN

In the second SLR system, Convolutional Neural Networks (CNN) is used for feature extraction and classification because CNN demonstrated its success for sign language recognition in recent researches [53] [54] [55] [56]. The main advantage is that CNN is able to extract most discriminant features of input data automatically by its convolution and pooling layers. Feature extraction methods are not needed before training as needed by other classification methods. On the other hand, CNN have several drawbacks. The first one is that it requires high computational power and large memory for training since the algorithm consists of multiple matrix multiplication operations. The second one is that training time of CNN is usually quite longer than the traditional machine learning algorithms. Parameter optimization requires too much effort due to this long training time. The last disadvantage is that it requires large amount of training data to learn most discriminant features accurately. In the following section, CNN is explained in details.

3.5.1 Convolutional Neural Networks (CNN)

Convolutional Neural Networks(CNN) is deep artificial neural network which is most commonly used to classify images and to recognize objects. CNN has changed the way pattern recognition works. Before CNN, meaningful informations were extracted from images by feature extraction algorithms separately. CNN is developed to not require any pre-processing of input images. CNN extracts most meaningful features by training data automatically. CNN consists of several steps as can be seen in Figure 3.16 [76].

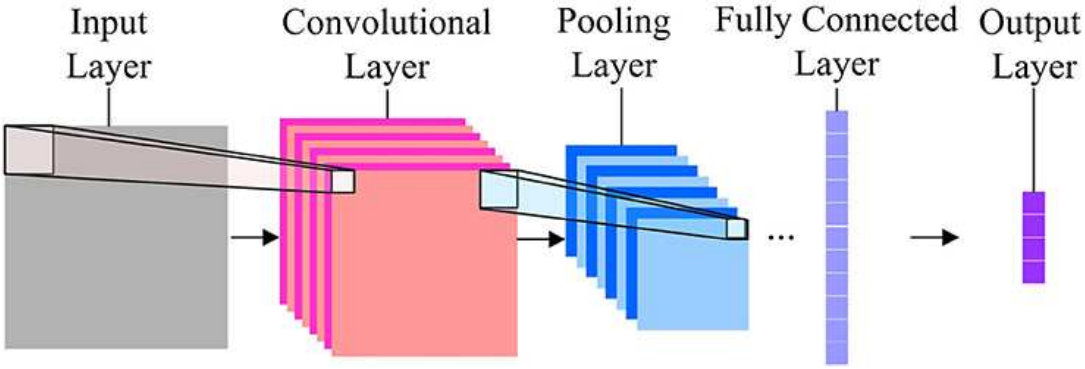

Figure 3.16: Architecture of a CNN

Input Layer: Input layer holds the 3D input volume consisting of pixel values of image having red, green and blue channels. For grayscale image, input volume consists of 2 dimensional pixel data with one channel.

Convolution Layer: Convolution layer is one of the main layers of a CNN network. The purpose of the convolution layer is to extract features from the input data. Convolution operation is applied on the input data by using filters to construct feature map. This layer takes the input data of size W_1xH_1xD_1 and produces output data of size W_2xH_2xD_2 where W_2, H_2, D_2 is given in Equation 3.13.

$$\begin{aligned} W_2 &= (W_1 - F + 2P)/S + 1 \\ H_2 &= (H_1 - F + 2P)/S + 1 \\ D_2 &= K \end{aligned} \tag{3.13}$$

where K is the number of filters, F is the spatial extend, S is the stride, and P is the amount of the zero padding.

Pooling Layer: Pooling layer is used to reduce the dimension of the feature map while preserving the important features. The main benefit of this layer is that it shortens the training time and reduces the memory utilization by reducing the number of parameters in the network. There are different types of pooling method: max pooling, average pooling, sum pooling. This layer takes the input data of size W_1xH_1xD_1 and produces output data of size W_2xH_2xD_2 where W_2, H_2, D_2 is given in Equation 3.14. In Figure 3.17, max pooling example can be seen.

$$\begin{aligned} W_2 &= (W_1 - F)/S + 1 \\ H_2 &= (H_1 - F)/S + 1 \\ D_2 &= D_1 \end{aligned} \tag{3.14}$$

where F is the spatial extend, S is the stride.

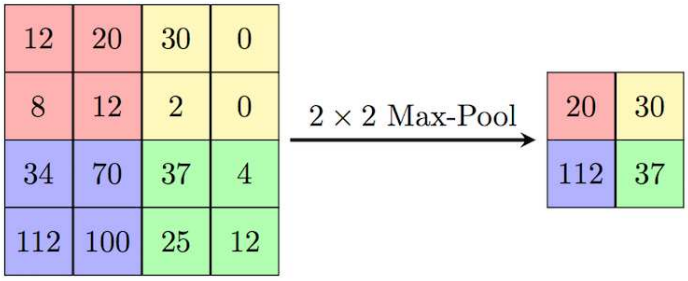

Figure 3.17: Max Pooling Example [5]

Fully Connected Layer: Fully Connected Layer is final learning layer which maps extracted feature maps into output classes. It outputs one dimensional array of size equal to the number of classes.

For dynamic SLR classification, CNN network to classify images cannot be used to classify videos because videos have temporal informations besides spatial informations. In literature, in order to classify videos with CNN, there are some methods used. In [54], they sampled 9 frames from sign videos and concatenate these 9

frames to construct one image for a sign video. In [53], they sampled 9 frames from sign video and stacked these 9 frames to form cube formed data for a sign video. By the help of these methods, they are able to extract features on both spatial and temporal dimensions. In our work two different CNN architectures, which have input layer similar to input layers used by [54] and [53], are implemented for dynamic sign language recognition. The architectures are given in the following subsections.

In order to find out the best performing architecture that results in higher recognition rate, CNN architectures that using different parameters are implemented and tested as in [55]. Firstly, CNN architectures were tested by different input layer resolutions. Optimal input layer resolution values are determined by several iterations. Secondly, effect of different pooling techniques on the performance is investigated. We decided to use max pooling technique for its better performance. Finally, different filter sizes for convolution operations were tested to find a best performing one and filter size of 5x5 is chosen for our systems.

3.5.1.1 CNN with Input Layer Consisting of Stacked Frames

In this CNN architecture, input layer consists of cube formed data which is formed by stacking 10 frames of sign video. The CNN architecture used is given in Figure 3.18. Different architectures with different convolution and max pooling layers are used and tested on the dataset to find out the best architecture. This architecture consists of six layers. Firstly, convolution operation with 20 different kernels size of 5x5x10(5x5: spatial dimension) is applied on cube formed input data to produce C1 layer. Then, in order to decrease the image size in spatial dimension and increase the robustness to spatial noise, 2x2 max pooling operation is applied. After the first max pooling, second convolution operation with 50 different filters size of 5x5x20(5x5: spatial dimension) and second 4x4 max pooling are applied. Last convolution layer C3 is produced by applying convolution operation with 100 filters size of 5x5x50 on S2 layer. By the help of multiple convolution and max pooling operation, CNN extracted features of input composed of 10 frames. Finally, fully connected layer multiplies the input feature vector by a weight matrix and maps input vectors into a class probability distribution. The class with high probability is chosen for output

sign.

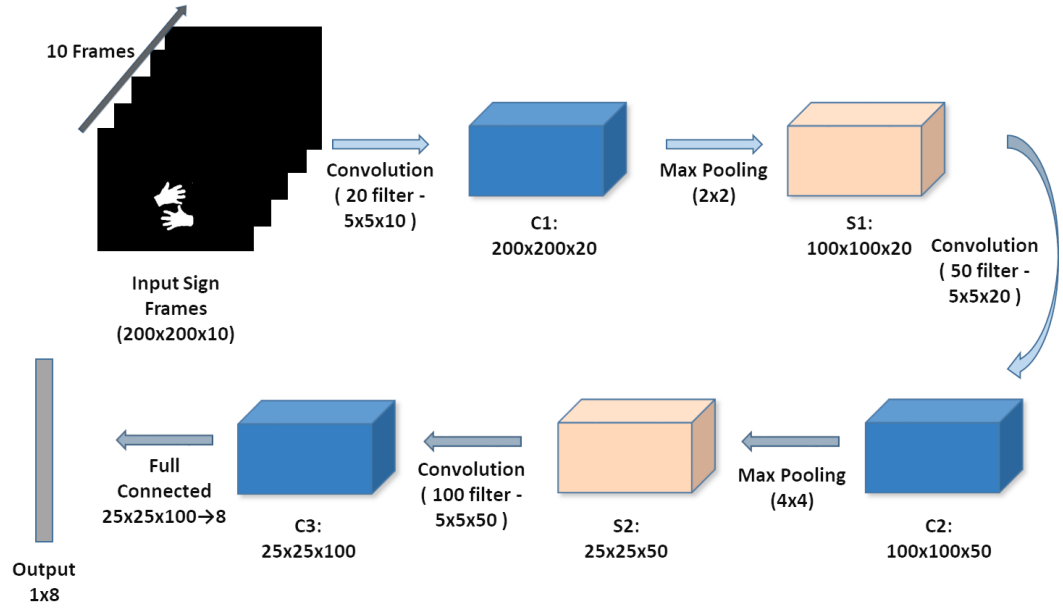

Figure 3.18: CNN with Input Layer Consisting of Stacked Frames

3.5.1.2 CNN with Input Layer Consisting of Concatenated Frames

In this CNN architecture, input layer consists of 2 dimensional image which is formed by concatenating 10 frames of sign video. The CNN architecture used is given in Figure 3.19. Different architectures with different convolution and max pooling layers are used and tested on the dataset to find out the best architecture. This architecture consists of six layers. Firstly, convolution operation with 20 different kernels size of 5x5x1(5x5: spatial dimension) is applied on input data to produce C1 layer. The remaining convolution and max pooling operations are the same as CNN architecture with input layer of stacked frames. Finally, fully connected layer multiplies the input feature vector by a weight matrix and maps input vectors into a class probability distribution.

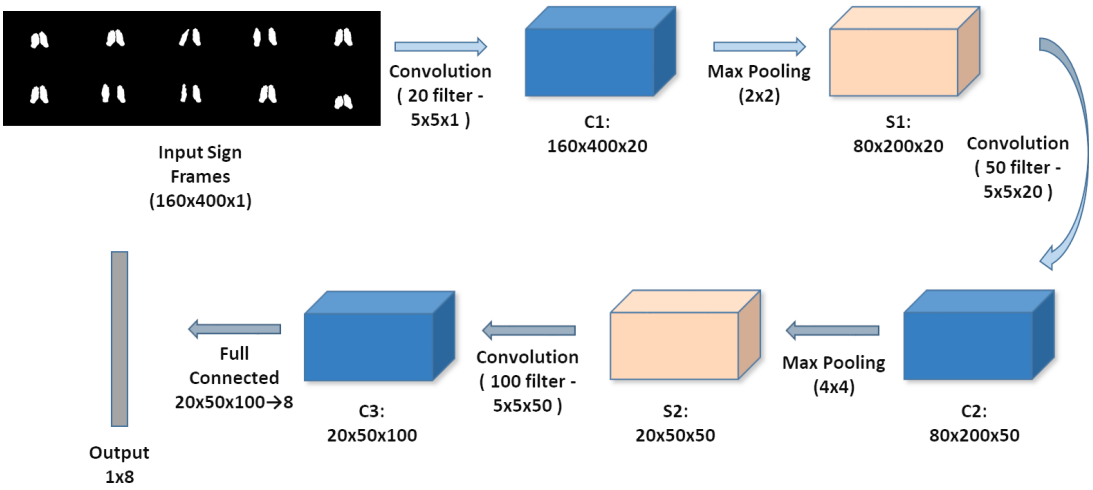

Figure 3.19: CNN with Input Layer Consisting of Concatenated Frames

CHAPTER 4

EXPERIMENTAL RESULTS

In this chapter, implemented sign language recognition systems are evaluated by conducting experiments on eNTERFACE dataset [25]. Generally, there are two types of tests to evaluate sign language recognition systems in the literature. The first one is signer-dependent test. In this type of test, training dataset and test dataset consist of sign videos that are performed by the same performer. The second one is signer-independent test, in which training videos and test videos are performed by different performers. In the literature, many of the works are focusing on signer-dependent tests for evaluating their proposed algorithms. However, proposed sign language recognition system must be usable by different people without the necessity of retraining for new people. Because of that, signer-independent tests are reported in details throughout this thesis.

In the first part of this chapter, dataset and gestures are explained in details. In the second part, cross-validation method which is used throughout this chapter is described. In the third part, optimization result of number of sampled frames per sign videos are reported. On the forth part, signer-independent test results for feature based SLR system and CNN based SLR system and comparison between them are given. Finally, signer-dependent test results are given.

4.1 Dataset

In order to evaluate the performance of the implemented SLR systems, eNTERFACE'06 American Sign Language (ASL) Dataset [25] is used. This dataset is prepared for SignTutor project of eNTERFACE'06 workshop. Videos are recorded with

a single web camera with 640x480 resolution and 25 frames per second frame rate. Dataset consists of 8 base ASL signs and 11 variation of base signs which contains different head movements and face expressions. In our work, face expressions and head movements are out of the scope of this thesis. In Table 4.1, descriptions of the ASL signs in terms of the hand expression can be found.

Table 4.1: Sign Names and Descriptions

Sign Name	**Hand Expression Description**
Afraid	Hands move from the sides to the middle to meet in the front of the body.
Clean	Right hand is slided over the left hand while right palm facing down and left palm facing up.
Door (noun)	One hand repetitively moves as if the door is opened.
Drink (noun)	Right hand repetitively moves like drinking from cup.
Fast	Hands move towards to body from in front of the body while fingers are partially closed and thumb is open.
Here	Right hand moves circularly parallel to the ground.
Look at	Hands move forward together starting from eyes.
Study	Fingers of right hand are open and move while left hand palm is facing upwards.

For each sign, five repetitions are recorded by eight subjects. In total, there are 320 videos used in our experimental tests. The signers wear a yellow glove on their left hands and a blue glove on their right hands.

4.2 Cross-Validation Method

Cross-validation is statistical technique for performance evaluation in machine learning systems. Is is useful method to estimate accuracy of the recognition algorithm in practice and to compare accuracy of different algorithms for available dataset. In machine learning systems, dataset is generally divided into two parts; training dataset and testing dataset. Classifiers are trained by training dataset and tested by testing dataset in an iterative way defined by cross-validation. There are different cross-validation methods proposed in the literature. Some of them is as follows:

- Random Subsampling
- K-fold Cross-Validation
- Leave-one-out Cross-Validation

Random Subsampling

In this method, training and testing experiments are performed in k iteration. In each iteration, testing set is randomly selected from entire dataset without replacement and remaining unselected part of the entire dataset is used for the training dataset as can be seen in Figure 4.1. After calculating the error rate of the each iteration, total error estimate is calculated as average of the separate estimates (Equation 4.1).

$$E = \frac{1}{K} \sum_{i=1}^{K} E_i \tag{4.1}$$

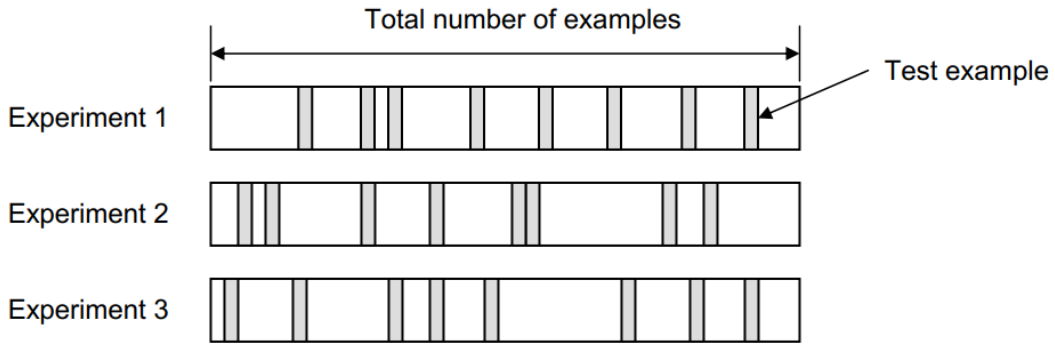

Figure 4.1: Random Supsampling [6]

K-Fold Cross-Validation

In this method, dataset is firstly divided into k equal folds. Training and testing experiments are performed in k iteration as can be seen in Figure 4.2. In each operation, different fold is used in the testing and remaining k-1 folds is used in training. Total error estimate is calculated as random subsampling method by Equation 4.1.

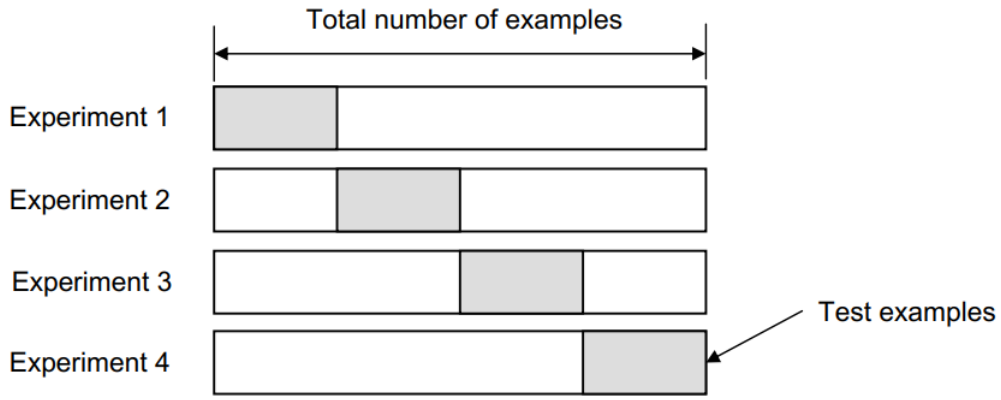

Figure 4.2: K-Fold Cross-Validation [6]

Leave-one-out Cross-Validation

Leave-one-out cross-validation is a one case of k-fold cross-validation, in which number of experiments is equal to the number of elements in the dataset. In each experiment, one data is used for testing and remaining data is used for training. The error estimate is calculated as k-fold cross-validation. This method is generally used when dataset is composed of small number of example.

In our work, k-fold cross-validation is used to estimate the performance of the implemented SLR systems since it is preferred mostly for accurate performance evaluation. In k-fold cross-validation, it is important to select appropriate k value for dataset. At first glance, the larger k may look better, because number of iteration and number of elements in training dataset increases with k. However, increase of k causes more overlapping dataset between trainings. Moreover, as k increases the size of the test dataset decreases. Thus, this will lead to less accurate performance measurements [77]. In literature, the most preferred k value is 10. However, we choose the value of k to be eight in signer-independent tests because it is equal to the number of signers in our dataset. The value of k greater than that can lead to over fitting problems. In other words, the videos of one signer are used as testing dataset and the other videos of seven signers are used as training dataset. K is chosen as to be 5 in signer-dependent test because number of videos per sign is equal to 5.

4.3 Frame Selection Method from Videos

In order to recognize the gesture performed by signer, sufficient number of sampled frame from sign video must be used in the feature extraction and recognition steps. Number of frames that must be sampled from sign video is dependent on gesture and dataset itself. In [78], authors proposed an algorithm to select key frames from sign videos according to temporal differences. Only key frames are used in recognition part. Key frame selection algorithms are not preferred in this work because different sign videos of same sign could have different number of key frames. As a result, this may result in challenges in classification step. In [54], it is claimed that 9 screen shots from sign video is sufficient for classifiers to recognize the sign language performed. In the proposed algorithm, 9 images are sampled from sign videos at equal intervals to construct training data as can be seen in Figure 4.3. In our work, the same approach is used, but the number of frames that will be sampled is defined according to the experiment shown in Figure 4.4.

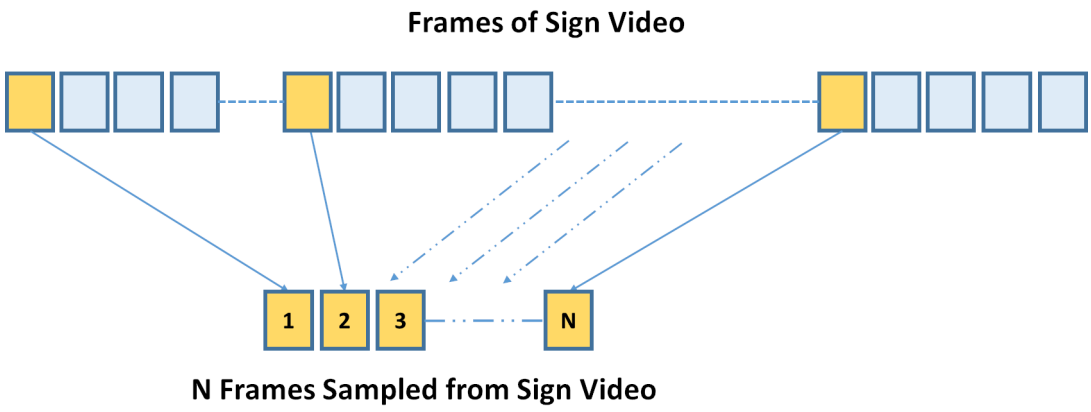

Figure 4.3: Sampling of N Frames from Video

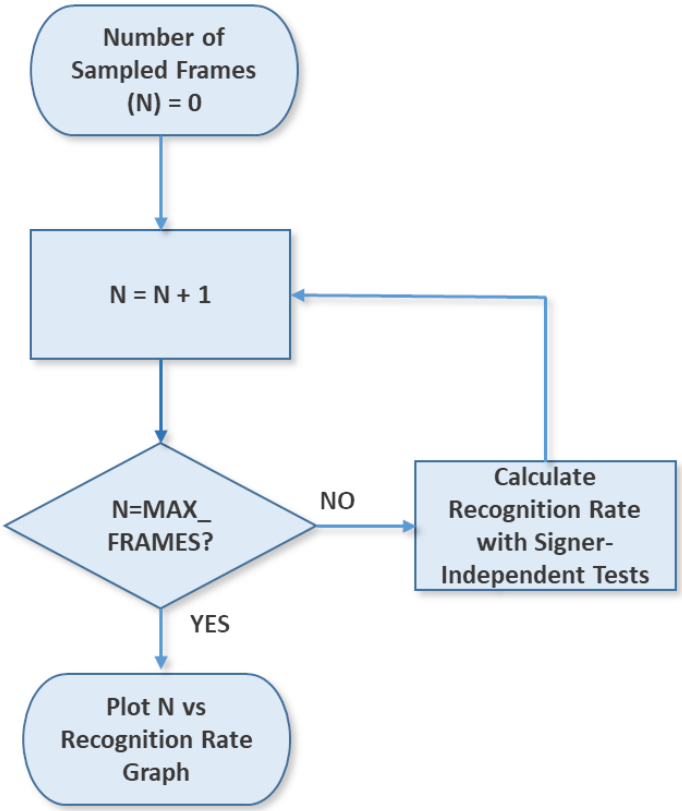

Figure 4.4: Experiment to Find Best Number of Sampled Frames Value

In each iteration of experiment, recognition rate of feature based SLR system is calculated by conducting signer-independent tests and number of sampled frames is incremented by one. At the end of the last iteration, number of sampled frames vs recognition rate graph is plotted as can be seen in Figure 4.5. As the number of sampled frames increases, training time, ram utilization and CPU utilization also increases. Because of that, number of sampled frames which is small as possible and results in high recognition rate is chosen for rest of our work because. According to results of the signer-independent tests for feature based SLR system, number of sampled frames is chosen as 10.

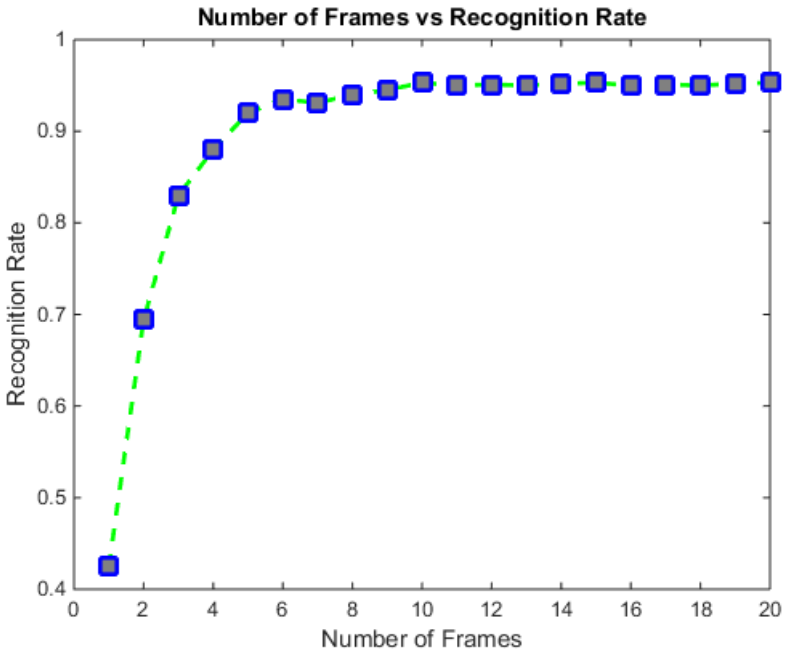

Figure 4.5: Number of Frames vs Recognition Rate

4.4 Signer-Independent Tests

Proposed sign language recognition systems must be usable by different people without retraining them for new people. In order to evaluate the system performance for new signers, signer-independent tests are conducted in this part. As mentioned earlier, k-fold cross-validation with k=8 is used for validation method in this work. In total, eight training and testing iterations are performed. In each iteration, gesture videos of different signers are used for testing and sign videos of remaining seven signer is used for training. Performance results of feature based SLR system and CNN based SLR system are given in following subsections.

4.4.1 Test results of Feature Based SLR System

In this subsection, performance results of feature based SLR system are given. Firstly, Trial count, correct classification count, and recognition rate of each fold is given in Table 4.2. Average recognition rate of 8 fold is equal to 95.31%.

Recognition rate of each sign in each fold is given in Table 4.3. Finally, confusion matrix which is constructed to summarize the performance of feature based SLR system is given in Table 4.4.

Table 4.2: Test Results of Each Fold for Feature Based SLR System

Subject	**Alex**	**Ana**	**Fx**	**Ismail**	**Jakov**	**Levacic**	**Oya**	**Pavel**	**Total**
Trial Count	40	40	40	40	40	40	40	40	320
Correct Recognition Count	39	39	40	40	38	40	36	33	305
Recognition Rate(%)	97.5	97.5	100	100	95	100	90	82.5	95.31

Table 4.3: Sign Based Recognition Rate for Feature Based SLR System

	Afraid	**Clean**	**Door**	**Drink**	**Fast**	**Here**	**Look**	**Study**	**Total**
Alex	100	100	100	100	100	100	80	100	97.5
Ana	100	100	80	100	100	100	100	100	97.5
Fx	100	100	100	100	100	100	100	100	100
İsmail	100	100	100	100	100	100	100	100	100
Jakov	60	100	100	100	100	100	100	100	95
Levacic	100	100	100	100	100	100	100	100	100
Oya	100	100	100	100	100	100	20	100	90
Pavel	0	100	100	100	100	100	100	60	82.5

Table 4.4: Confusion Matrix for Feature Based SLR System

	Afraid	**Clean**	**Door**	**Drink**	**Fast**	**Here**	**Look**	**Study**
Afraid	33	1			4			2
Clean		40						
Door			39		1			
Drink				40				
Fast					40			
Here						40		
Look			3		2		35	
Study		2						38

4.4.2 Test Results of CNN Based SLR System

In this subsection, performance results of CNN with input layer consisting of stacked frames and CNN with input layer consisting of concatenated frames are given. These two CNN architectures are evaluated by three different input data types given below:

- Input data consisting of grayscale original frames.
- Input data consisting of grayscale segmented frames
- Input data consisting of boundary based segmented frames.

Examples of grayscale original frame, grayscale segmented frame and boundary based segmented frame can be seen in Figure 4.6.

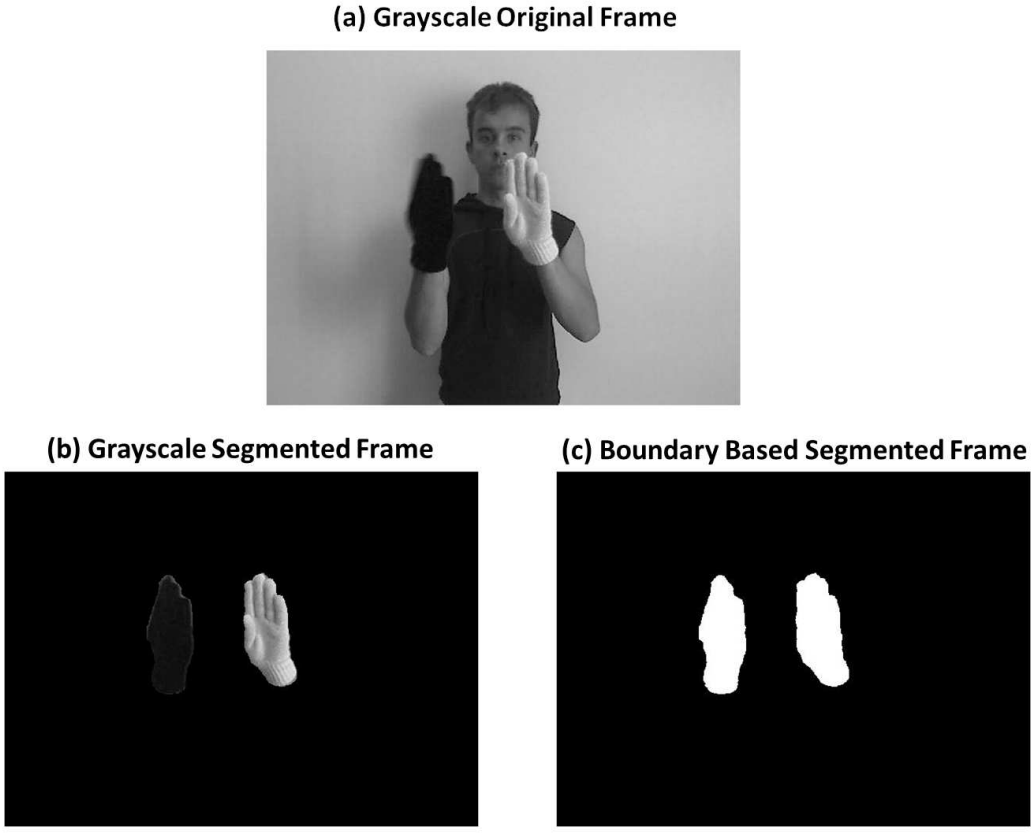

Figure 4.6: Example of Frame Types

For each CNN architectures and each input data types, 8-fold cross-validation is performed. In total, 48 training operations are performed. Average recognition rates of

CNN architectures with different input data types are given in Table 4.5. According to the results, CNN with input layer consisting of stacked frames has higher recognition rate then CNN with input layer consisting of concatenated frames for all input data types. Another results shows that CNN with input layer consisting of original frames has poor performance. The main reason for low accuracy is that the number of training data is not enough for CNN to classify unsegmented frames. Also, CNN works better with boundary based segmented frames. CNN with input layer consisting of stacked boundary based segmented frames has the highest accuracy with 93.12% recognition rate.

Table 4.5: Test Results of CNN Based SLR Systems

CNN and Input Layer Type	**Trial Number**	**Correct Classification Number**	**Success Rate(%)**
CNN with Input Consisting of Stacked Grayscale Original Frames	320	187	58.43
CNN with Input Consisting of Stacked Grayscale Segmented Frames	320	291	90.93
CNN with Input Consisting of Stacked Boundary Based Segmented Frames	320	298	93.12
CNN with Input Consisting of Concatenated Grayscale Original Frames	320	149	46.56
CNN with Input Consisting of Concatenated Grayscale Segmented Frames	320	287	89.68
CNN with Input Consisting of Concatenated Boundary Based Segmented Frames	320	291	90.93

In the following tables, the results of CNN with input layer consisting of stacked boundary based segmented images are given. Firstly, trial count, correct classification count, and recognition rate of each fold is given in Table 4.6. Average recognition rate of 8 fold is equal to 93.12%.

Recognition rate of each sign in each fold is given in Table 4.7. Finally, confusion matrix which is constructed to summarize the performance of CNN based SLR system is given in Table 4.8.

Table 4.6: Test Results of Each Fold for CNN Based SLR System

Subject	**Alex**	**Ana**	**Fx**	**Ismail**	**Jakov**	**Levacic**	**Oya**	**Pavel**	**Total**
Trial Count	40	40	40	40	40	40	40	40	320
Correct Recognition Count	36	36	37	38	38	39	37	37	298
Recognition Rate(%)	90	90	92.5	95	95	97.5	92.5	92.5	93.12

Table 4.7: Sign Based Recognition Rate for CNN Based SLR System

	Afraid	**Clean**	**Door**	**Drink**	**Fast**	**Here**	**Look**	**Study**	**Total**
Alex	100	100	100	100	40	100	80	100	90
Ana	100	100	80	100	100	80	100	60	90
Fx	100	80	100	100	60	100	100	100	92.5
İsmail	100	100	100	100	100	100	60	100	95
Jakov	100	80	100	100	100	100	100	80	95
Levacic	100	100	100	80	100	100	100	100	97.5
Oya	100	80	100	100	100	100	60	100	92.5
Pavel	60	100	100	100	100	80	100	100	92.5

Table 4.8: Confusion Matrix for CNN Based SLR System

	Afraid	**Clean**	**Door**	**Drink**	**Fast**	**Here**	**Look**	**Study**
Afraid	38							2
Clean		37		2		1		
Door			39		1			
Drink				39	1			
Fast			5		35			
Here		1		1		38		
Look			1		4		35	
Study		2		1				37

4.4.3 Test Results of CNN Based SLR System with Data Augmentation

The main disadvantage of CNN classifier is that it generally works best with large number of training data [79]. In dataset we use, there is 40 videos for each sign. Experimental results show that CNN provides decent performance with small number of training samples. In machine learning systems, in order to increase the system robustness to conditions in which illumination, rotation, size of the objects slightly differs, datasets which have limited number of elements can be expanded with data augmentation techniques. Dataset diversity of eNTERFACE dataset is sufficient, since dataset consists of sign videos of eight subjects. However, number of training samples is quite low for CNN classifier. Because of this, data augmentation is used to increase the performance of CNN classifier. In this work, dataset is expanded by using the method used by [54]. In dataset, sign generally lasts for 50 frames in videos, but only 10 frames are sampled with specific rate to construct training data. In order to expand training data, remaining unused frames are sampled to construct further training data as can be seen in Figure 4.7. By doing this, total number of training data is increased by 1 times and 3 times for two experiments.

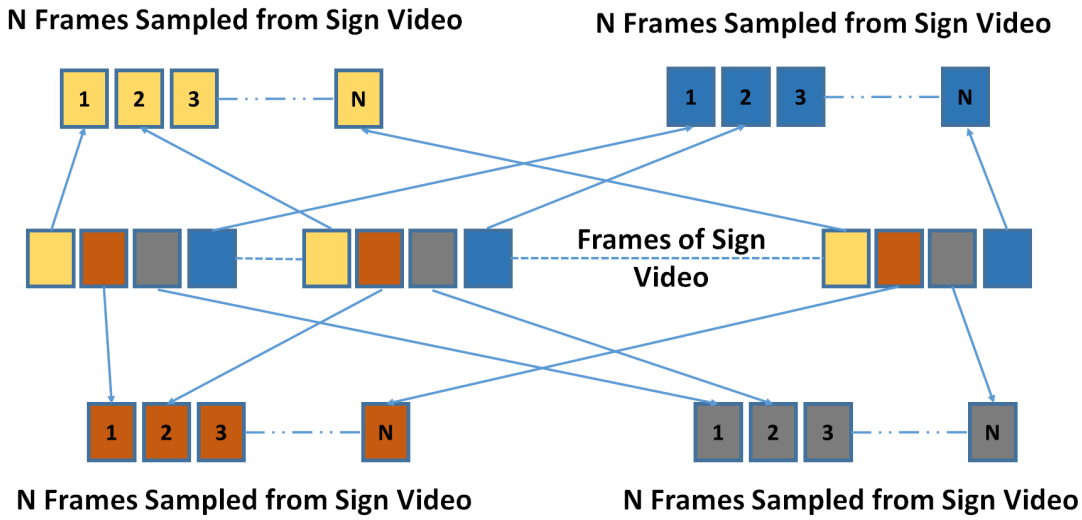

Figure 4.7: Frame Sampling for Training Dataset Augmentation

Performance results of CNN based SLR system with data augmentation are given in Table 4.9 and Table 4.10. It can be seen that CNN based SLR system achieved

recognition rate of 93.43% with doubled dataset and 94.29% with quadrupled dataset.

Table 4.9: Test Results of Each Fold for CNN Based SLR System with Doubled Dataset

Subject	**Alex**	**Ana**	**Fx**	**Ismail**	**Jakov**	**Levacic**	**Oya**	**Pavel**	**Total**
Trial Count	80	80	80	80	80	80	80	80	640
Correct Recognition Count	71	73	74	76	77	78	75	74	598
Recognition Rate(%)	88.75	91.25	92.5	95	96.25	97.5	93.75	92.5	93.43

Table 4.10: Test Results of Each Fold for CNN Based SLR System with Quadrupled Dataset

Subject	**Alex**	**Ana**	**Fx**	**Ismail**	**Jakov**	**Levacic**	**Oya**	**Pavel**	**Total**
Trial Count	160	160	160	160	160	160	160	160	1280
Correct Recognition Count	144	147	153	153	156	159	147	150	1207
Recognition Rate(%)	90	91.87	95.62	95.62	97.5	99.37	91.87	93.75	94.29

4.5 Signer-Dependent Tests

As mentioned earlier, k-fold cross-validation with k=5 is used for validation method in signer-dependent tests. In total, five training and testing iterations are performed for each signer. In each iteration, 1 different video of each sign for signer is used for testing and remaining sign videos of signer are used for training. Signer-dependent tests are conducted for only feature based SLR system, because number of sign videos per signer is not enough to train CNN Classifier. As can be seen from Table 4.11, all recognition rate for each signer is equal to 100%.

Table 4.11: Signer-Dependent Test Results for Feature Based SLR System

Subject	**Alex**	**Ana**	**Fx**	**Ismail**	**Jakov**	**Levacic**	**Oya**	**Pavel**	**Total**
Trial Count	40	40	40	40	40	40	40	40	320
Correct Recognition Count	40	40	40	40	40	40	40	40	320
Recognition Rate(%)	100	100	100	100	100	100	100	100	100

4.6 Time Measurements and Discussion

We implemented and tested the SLR systems on MATLAB R2014b running on desktop computer which has Intel Core i5-2410M 2.30 GHz processor and 4GB RAM. We used LibSVM [80] library for multi-class SVM and MatConvNet [81] library for CNN implementation, and other MATLAB functions for segmentation and feature extraction parts. By considering all performance measurements, which are segmentation time, feature extraction time, training time, prediction time, and recognition rate, Table 4.12 is prepared for the implemented SLR systems. Time measurements of each algorithms are taken several times and mean values are calculated.

Table 4.12: Comparison of the Implemented SLR Systems

Measurement	**Feature Based SLR System**	**CNN Based SLR System**
Average Segmentation Time for One Dynamic Gesture(ms)	613	613
Average Feature Extraction Time for One Dynamic Gesture(ms)	68	n/a
Average Training Time of Classifier	24.3 seconds	6-7 hours
Average Prediction Time of Classifier(ms)	257	121
Recognition Rate(%)	95.31	93.12

Experimental results demonstrated that feature based SLR system provides better

recognition rate than CNN based SLR system when they are trained with a limited number of training data. The main reason for this is that SVM is able to find the right hyperplane with between extracted feature vectors even if the training dataset has small number of samples.

The main disadvantage of CNN based SLR system is the training time. It takes approximately 6-7 hours to train CNN in our development environment. Also, there are several parameters that affect CNN's behavior such as filter size, number of layers. Finding a good combination of parameters of a CNN is harder when the training takes too much time. Due to the long training times and parameter tuning, it is difficult to develop scalable application that can be easily adopted to work with new datasets.

Another important time measurement to discuss is prediction time. According to the measurements, average prediction time of feature based SLR system for one sign is equal 938(613+257+68) milliseconds which is the sum of average segmentation, feature extraction and SVM prediction time. Also, average prediction time of CNN based SLR system is equal 734(613+121) milliseconds which is the sum of segmentation time and CNN prediction time. Both of the systems can meet the real-time requirements with their short average recognition time and low prediction time variance. Furthermore, CNN based SLR system is also trained and tested by original sign images without hand segmentation step. In this case, although the average prediction time decreases to 121 milliseconds it provides a poor recognition rate due to the fact that CNN cannot extract discriminant features of original images with a small dataset.

We also investigated the effect of training dataset size on recognition rate of CNN based SLR system. Dataset is expanded to 4 times by sampling unused frames of sign videos. As a result, recognition rate increased from 93.12% to 94.29%. As we expected, average training time also increased significantly while average prediction time stayed almost the same as before.

Although, almost most of the systems in the literature review chapter and the systems that we implemented are not evaluated by the same dataset, by looking at the recognition rates and computation times of the implemented systems, one can say that the implemented systems achieved a high performance to recognize dynamic hand gestures with a limited training dataset. There is a work that uses the same dataset

for performance evaluation. In [60], authors used low-level hand feature descriptors, which are bounding eclipse, bounding box, and center of mass coordinates, and HMM classifier to recognize hand gestures. According to the results, the only result that can be compared to our results is recognition rate. The system provides a recognition rate of 94.19% which is relatively low compared to our feature based SLR system.

CHAPTER 5

CONCLUSIONS AND FUTURE WORKS

5.1 Summary and Conclusion

Traditional machine learning and deep learning algorithms are the major approaches used for SLR in the literature. When we considered the dissimilarities between traditional machine learning and deep learning algorithms, we decided to use traditional machine learning algorithms in our system for having high accuracy and low computational cost. However, we also implemented deep learning based SLR system because deep learning algorithms have gained popularity and attention in recent works for SLR.

The first SLR system, which is based on traditional machine learning, consists of three main step: hand segmentation, feature extraction and classification. System starts with hand segmentation algorithm. In this algorithm, firstly region of interest is found by using HSV thresholding. In order to segment hands in the region of interest, Fuzzy C-Means Clustering algorithm is used. Then, post processing operations are used to correct errors caused by clustering algorithm and lighting conditions. For feature extraction step, we considered and compared the most frequently used appearance based feature descriptors across multiple criteria which are computational cost, scale and rotation invariance, and accuracy. We decided to use HOG feature descriptor for describing the hand shape because it is scale and illumination invariant, has low computational cost, and has high recognition accuracy for hand shape detection in the literature. In the classification step, multi-class SVM is decided to be used to classify HOG feature vectors by considering dataset dependency, computation time, number of parameter tuning needed for optimization, and accuracy. The second system, which

is based on deep learning, consists of two steps: hand segmentation, feature extraction and classification based on CNN. System starts with the same hand segmentation algorithm as in the first system. Then, CNN, which is most popular deep learning algorithm, is used to classify series of segmented hand images.

In order to justify the implemented SLR systems, they are evaluated by conducting signer independent tests on eNTERFACE dataset consisting of 8 ASL signs of 8 signers. We focused on the signer independent tests, because the systems must also result in good performance when they are used by different people. Signer independent tests are conducted by using k-fold cross validation technique. K is chosen as to be 8 in signer-independent tests, because it is equal to the number of signers in the dataset. Experimental results demonstrated that feature based SLR system and CNN based SLR system achieved recognition rate of 95.31% and 93.12%, respectively. The effect of training dataset size on the recognition rate of CNN is investigated by using data augmentation. The number of samples in the training dataset is increased to 4 times by sampling unused frames of sign videos. Recognition rate increased to 94.29% with expanded dataset. As we expected, feature based SLR system provides better recognition rate than CNN based SLR system for our case where a limited number of training data is available. The main disadvantage of CNN based SLR system is long training time. It is difficult to develop a scalable CNN based SLR application that can be easily usable with new set of hand gestures. Moreover, both of the systems can be used in real time with their short average recognition time and low prediction time variance. With the experimental results, it can be concluded that both of the SLR systems can meet the goal of obtaining high accuracy while keeping the cost to a minimum with their short average recognition time and high recognition rate.

5.2 Future Works

In this work, the implemented systems can only work with the isolated signs, in other words systems recognize one sign at a time. In the future, this work could be extended to the system, which detects and classifies all words in a sentence. Also, more signs could be introduced to systems by training systems with corresponding datasets. Moreover, the implemented systems could be used in mobile application in the future

since systems are designed by considering the computational power constraints of mobile platforms.

REFERENCES

[1] Poorani, Prathiba, and Ravindran, "Integrated feature extraction for image," 2013.

[2] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, pp. 886–893 vol. 1, June 2005.

[3] D. Duman, "Moving vehicle classification," Master's thesis, Middle East Technical University, 2013.

[4] "A comparison of multiclass svm methods." http://courses.media.mit.edu/2006fall/mas622j/Projects/aisen-project/. Accessed: 29 June, 2018.

[5] "Max-pooling / pooling." https://computersciencewiki.org/index.php/Max-pooling_/_Pooling. Accessed: 5 August, 2018.

[6] "Cross-validation." https://www.cs.tau.ac.il/ nin/Courses/NC05/pr_l13.pdf. Accessed: 29 June, 2018.

[7] S. Mitra and T. Acharya, "Gesture recognition: A survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, pp. 311–324, May 2007.

[8] J. S. Sonkusare, N. B. Chopade, R. Sor, and S. L. Tade, "A review on hand gesture recognition system," in *2015 International Conference on Computing Communication Control and Automation*, pp. 790–794, Feb 2015.

[9] A. Er-Rady, R. Faizi, R. O. H. Thami, and H. Housni, "Automatic sign language recognition: A survey," in *2017 International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*, pp. 1–7, May 2017.

[10] H. Kaur and J. Rani, "A review: Study of various techniques of hand gesture recognition," in *2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES)*, pp. 1–5, July 2016.

[11] 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*, pp. 411–417, Sept 2012.

[12] M. B. Waldron and S. Kim, "Isolated asl sign recognition system for deaf persons," *IEEE Transactions on Rehabilitation Engineering*, vol. 3, pp. 261–271, Sept 1995.

[13] M. W. Kadous, "Machine recognition of auslan signs using powergloves: Towards large-lexicon recognition of sign language," 1996.

[14] C. Vogler and D. Metaxas, "Adapting hidden markov models for asl recognition by using three-dimensional computer vision methods," in *1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*, vol. 1, pp. 156–161 vol.1, Oct 1997.

[15] H. Brashear, T. Starner, P. Lukowicz, and H. Junker, "Using multiple sensors for mobile sign language recognition," in *Seventh IEEE International Symposium on Wearable Computers, 2003. Proceedings.*, pp. 45–52, Oct 2003.

[16] W. Wang and J. Pan, "Hand segmentation using skin color and background information," in *2012 International Conference on Machine Learning and Cybernetics*, vol. 4, pp. 1487–1492, July 2012.

[17] W. Tan, C. Wu, S. Zhao, and S. Chen, "Hand extraction using geometric moments based on active skin color model," in *2009 IEEE International Conference on Intelligent Computing and Intelligent Systems*, vol. 4, pp. 468–471, Nov 2009.

[18] J. Rekha, J. Bhattacharya, and S. Majumder, "Shape, texture and local movement hand gesture features for indian sign language recognition," in *3rd International Conference on Trendz in Information Sciences Computing (TISC2011)*, pp. 30–35, Dec 2011.

[19] A. Y. Dawod, J. Abdullah, and M. J. Alam, "Adaptive skin color model for hand segmentation," in *2010 International Conference on Computer Applications and Industrial Electronics*, pp. 486–489, Dec 2010.

[20] S. Mo, S. Cheng, and X. Xing, "Hand gesture segmentation based on improved kalman filter and tsl skin color model," in *2011 International Conference on Multimedia Technology*, pp. 3543–3546, July 2011.

[21] B. Büyüksaraç, M. M. Bulut, and G. B. Akar, "Sign language recognition by image analysis," in *2016 24th Signal Processing and Communication Application Conference (SIU)*, pp. 417–420, May 2016.

[22] C. M. Jin, Z. Omar, and M. H. Jaward, "A mobile application of american sign language translation via image processing algorithms," in *2016 IEEE Region 10 Symposium (TENSYMP)*, pp. 104–109, May 2016.

[23] M. A. Uddin and S. A. Chowdhury, "Hand sign language recognition for bangla alphabet using support vector machine," in *2016 International Conference on Innovations in Science, Engineering and Technology (ICISET)*, pp. 1–4, Oct 2016.

[24] S. C. Agrawal, A. S. Jalal, and C. Bhatnagar, "Recognition of indian sign language using feature fusion," in *2012 4th International Conference on Intelligent Human Computer Interaction (IHCI)*, pp. 1–5, Dec 2012.

[25] "enterface'07 the summer workshop on multimodal interface." http://www.enterface.net/enterface06/docs/results/eNTERFACE06_proceedings.pdf. Accessed: 29 June, 2018.

[26] L.-G. Zhang, Y. Chen, G. Fang, X. Chen, and W. Gao, "A vision-based sign language recognition system using tied-mixture density hmm," in *ICMI*, 2004.

[27] Z. Mo and U. Neumann, "Real-time hand pose recognition using low-resolution depth images," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2, pp. 1499–1505, 2006.

[28] X. Liu and K. Fujimura, "Hand gesture recognition using depth data," in *Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004. Proceedings.*, pp. 529–534, May 2004.

[29] D. Uebersax, J. Gall, M. V. den Bergh, and L. V. Gool, "Real-time sign language letter and word recognition from depth data," in *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pp. 383–390, Nov 2011.

[30] Z. Li and R. Jarvis, "Real time hand gesture recognition using a range camera," 01 2009.

[31] Y. Chen and W. Zhang, "Research and implementation of sign language recognition method based on kinect," in *2016 2nd IEEE International Conference on Computer and Communications (ICCC)*, pp. 1947–1951, Oct 2016.

[32] J. Zhang, W. Zhou, and H. Li, "A new system for chinese sign language recognition," in *2015 IEEE China Summit and International Conference on Signal and Information Processing (ChinaSIP)*, pp. 534–538, July 2015.

[33] M. Oszust and M. Wysocki, "Polish sign language words recognition with kinect," in *2013 6th International Conference on Human System Interactions (HSI)*, pp. 219–226, June 2013.

[34] K. Dixit and A. S. Jalal, "Automatic indian sign language recognition system," in *2013 3rd IEEE International Advance Computing Conference (IACC)*, pp. 883–887, Feb 2013.

[35] J. Ekbote and M. Joshi, "Indian sign language recognition using ann and svm classifiers," in *2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, pp. 1–5, March 2017.

[36] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91–, 11 2004.

[37] 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)*, pp. 1–5, Jan 2016.

[38] D. Kim and R. Dahyot, "Face components detection using surf descriptors and svms," in *2008 International Machine Vision and Image Processing Conference*, pp. 51–56, Sept 2008.

[39] K. Singh and S. Chander, "Content based image retrival using surf , svm and color histogram – a review," 2014.

[40] H. Bay, A. Ess, T. Tuytelaars, and L. V. 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.

[41] J. He, Z. Liu, and J. Zhang, "Chinese sign language recognition based on trajectory and hand shape features," in *2016 Visual Communications and Image Processing (VCIP)*, pp. 1–4, Nov 2016.

[42] M. Hasan, T. H. Sajib, and M. Dey, "A machine learning based approach for the detection and recognition of bangla sign language," in *2016 International Conference on Medical Engineering, Health Informatics and Technology (MediTec)*, pp. 1–5, Dec 2016.

[43] A. Sharma, S. Singh, and A. Sharma, "Implementation of single precision conventional and fused floating point add-sub unit using verilog," in *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, pp. 169–171, March 2017.

[44] S. Lahoti, S. Kayal, S. Kumbhare, I. Suradkar, and V. Pawar, "Android based american sign language recognition system with skin segmentation and svm," in *2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pp. 1–6, July 2018.

[45] T. Pariwat and P. Seresangtakul, "Thai finger-spelling sign language recognition using global and local features with svm," in *2017 9th International Conference on Knowledge and Smart Technology (KST)*, pp. 116–120, Feb 2017.

[46] E. M. P. S. Edirisinghe, P. W. G. D. Shaminda, I. D. T. Prabash, N. S. Hettiarachchige, L. Seneviratne, and U. A. A. Niroshika, "Enhanced feature extraction method for hand gesture recognition using support vector machine," in *2013 IEEE 8th International Conference on Industrial and Information Systems*, pp. 139–143, Dec 2013.

[47] F.-S. Chen, C.-M. Fu, and C.-L. Huang, "Hand gesture recognition using a real-

time tracking method and hidden markov models," *Image and Vision Computing*, vol. 21, no. 8, pp. 745 – 758, 2003.

[48] R. Shrivastava, "A hidden markov model based dynamic hand gesture recognition system using opencv," in *2013 3rd IEEE International Advance Computing Conference (IACC)*, pp. 947–950, Feb 2013.

[49] M. Elmezain, A. Al-hamadi, and B. Michaelis, "Hand gesture recognition based on combined features extraction."

[50] A. Rahagiyanto, A. Basuki, R. Sigit, A. Anwar, and M. Zikky, "Hand gesture classification for sign language using artificial neural network," in *2017 21st International Computer Science and Engineering Conference (ICSEC)*, pp. 1–5, Nov 2017.

[51] P. Hong, M. Turk, and T. S. Huang, "Gesture modeling and recognition using finite state machines," in *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580)*, pp. 410–415, 2000.

[52] R. Verma and A. Dev, "Vision based hand gesture recognition using finite state machines and fuzzy logic," in *2009 International Conference on Ultra Modern Telecommunications Workshops*, pp. 1–6, Oct 2009.

[53] J. Huang, W. Zhou, H. Li, and W. Li, "Sign language recognition using 3d convolutional neural networks," in *2015 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6, June 2015.

[54] Y. Ji, S. Kim, and K. B. Lee, "Sign language learning system with image sampling and convolutional neural network," in *2017 First IEEE International Conference on Robotic Computing (IRC)*, pp. 371–375, April 2017.

[55] G. A. Rao, K. Syamala, P. V. V. Kishore, and A. S. C. S. Sastry, "Deep convolutional neural networks for sign language recognition," in *2018 Conference on Signal Processing And Communication Engineering Systems (SPACES)*, pp. 194–197, Jan 2018.

[56] M. Taskiran, M. Killioglu, and N. Kahraman, "A real-time system for recognition of american sign language by using deep learning," in *2018 41st International Conference on Telecommunications and Signal Processing (TSP)*, pp. 1–5, July 2018.

[57] R. A. Elsayed, M. I. Abdalla, and M. S. Sayed, "Hybrid method based on multi-feature descriptor for static sign language recognition," in *2017 Eighth International Conference on Intelligent Computing and Information Systems (ICICIS)*, pp. 98–105, Dec 2017.

[58] P. Usachokcharoen, Y. Washizawa, and K. Pasupa, "Sign language recognition with microsoft kinect's depth and colour sensors," in *2015 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, pp. 186–190, Oct 2015.

[59] D. Zhi, T. E. A. de Oliveira, V. P. d. Fonseca, and E. M. Petriu, "Teaching a robot sign language using vision-based hand gesture recognition," in *2018 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*, pp. 1–6, June 2018.

[60] B. Buyuksarac, "Sign language recogniton by image analysis," Master's thesis, Middle East Technical University, 2015.

[61] "Machine learning vs. deep learning." http://www.dataversity.net/machine-learning-vs-deep-learning/. Accessed: 1 November, 2018.

[62] P. Ganesan, V. Rajini, B. S. Sathish, and K. B. Shaik, "Hsv color space based segmentation of region of interest in satellite images," in *2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*, pp. 101–105, July 2014.

[63] P. Soille, *Morphological image analysis: principles and applications*, pp. 173–174. Springer Science & Business Media, 2013.

[64] J. Bezdec, *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York, 1981.

[65] L. G. Shapiro and G. Linda, "stockman, george c," *Computer Vision, Prentice hall. ISBN 0-13-030796-3*, pp. 69–73, 2002.

[66] Suharjito, R. Anderson, F. Wiryana, M. C. Ariesta, and G. P. Kusuma, "Sign language recognition application systems for deaf-mute people: A review based on input-process-output," *Procedia Computer Science*, vol. 116, pp. 441 – 448, 2017. Discovery and innovation of computer science technology in artificial intelligence era: The 2nd International Conference on Computer Science and Computational Intelligence (ICCSCI 2017).

[67] 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.

[68] Y. Kortli, M. Jridi, A. A. Falou, and M. Atri, "A comparative study of cfs, lbp, hog, sift, surf, and brief techniques for face recognition," 2018.

[69] R. Bora, A. Bisht, A. Saini, T. Gupta, and A. Mittal, "Isl gesture recognition using multiple feature fusion," in *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, pp. 196–199, March 2017.

[70] A. B. Hassanat, M. A. Abbadi, G. A. Altarawneh, and A. A. Alhasanat, "Solving the problem of the k parameter in the knn classifier using an ensemble learning approach," *CoRR*, vol. abs/1409.0919, 2014.

[71] M. Bashiri and A. Farshbaf Geranmayeh, "Tuning the parameters of an artificial neural network using central composite design and genetic algorithm," *Scientia Iranica*, vol. 18, pp. 1600–1608, 08 2011.

[72] "How does artificial neural network (ann) algorithm work?." https://www.analyticsvidhya.com/blog/2014/10/ann-work-simplified/. Accessed: 1 November, 2018.

[73] "Understanding support vector machine algorithm from examples." https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/. Accessed: 29 June, 2018.

[74] "Support vector machines: A simple explanation." https://www.kdnuggets.com/2016/07/support-vector-machines-simple-explanation.html. Accessed: 29 June, 2018.

[75] V. Vapnik, *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, 1995.

[76] M. Peng, C. Wang, T. Chen, G. Liu, and X. Fu, "Dual temporal scale convolutional neural network for micro-expression recognition," *Frontiers in Psychology*, vol. 8, p. 1745, 2017.

[77] "Cross-validation." http://leitang.net/papers/ency-cross-validation.pdf. Accessed: 29 June, 2018.

[78] A. S. Jalal, "Automatic recognition of dynamic isolated sign in video for indian sign language," 2016.

[79] A. Oliveira, S. Pereira, and C. A. Silva, "Augmenting data when training a cnn for retinal vessel segmentation: How to warp?," in *2017 IEEE 5th Portuguese Meeting on Bioengineering (ENBENG)*, pp. 1–4, Feb 2017.

[80] C.-C. Chang and C.-J. Lin, "Libsvm: A library for support vector machines," *ACM TIST*, vol. 2, pp. 27:1–27:27, 2011.

[81] A. Vedaldi and K. Lenc, "Matconvnet - convolutional neural networks for matlab," in *ACM Multimedia*, 2015.

APPENDIX A

SIGN IMAGES

A.1 Afraid

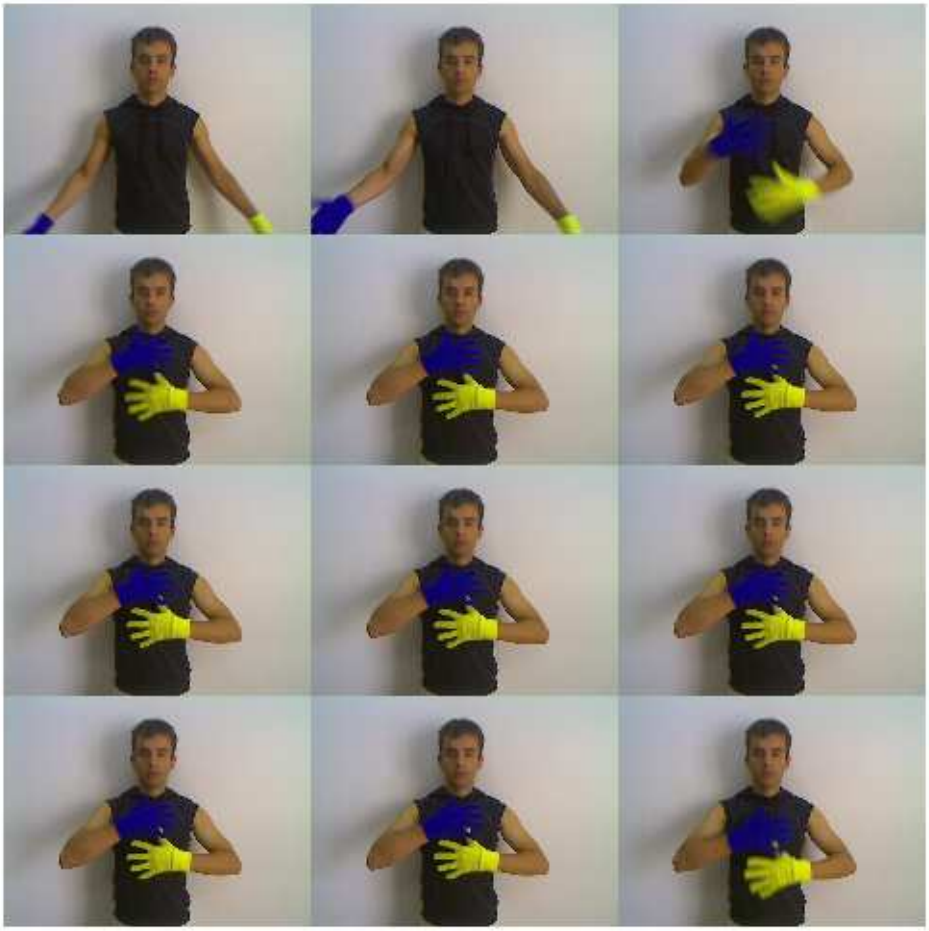

Figure A.1: Afraid Sign, Signer: Alex

A.2 Clean

Figure A.2: Clean Sign, Signer: Ana

A.3 Door (noun)

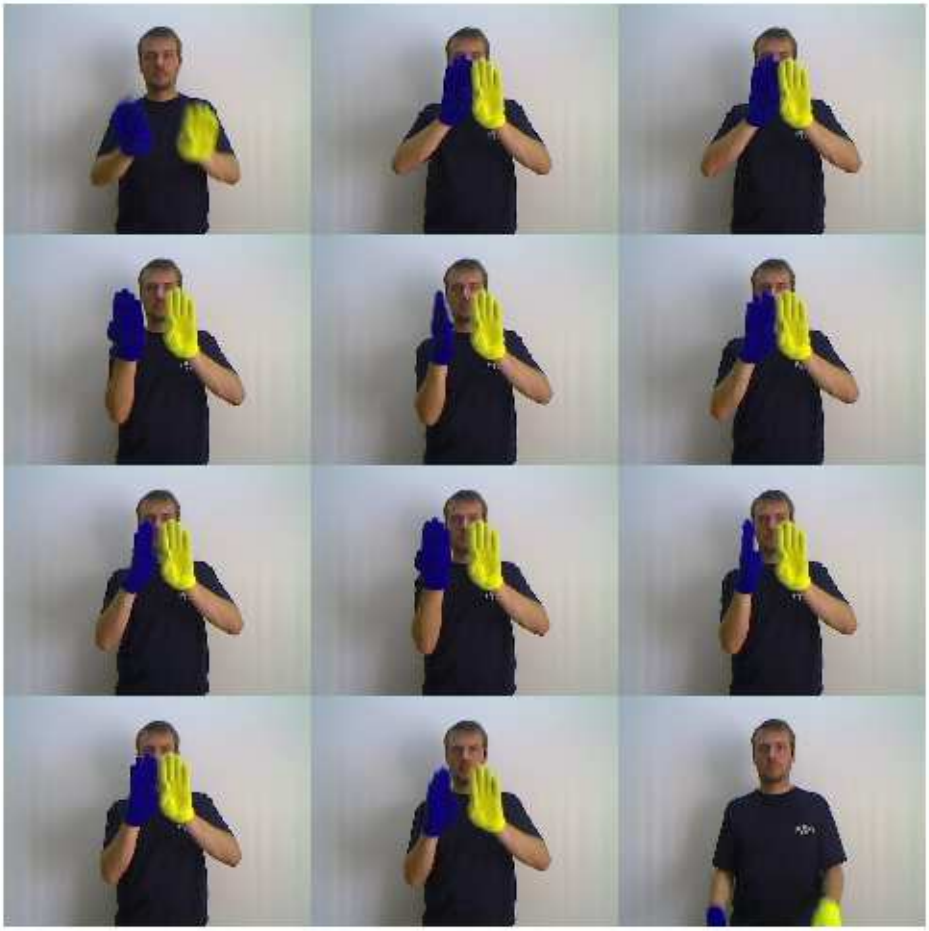

Figure A.3: Door (noun) Sign, Signer: FX

A.4 Drink (noun)

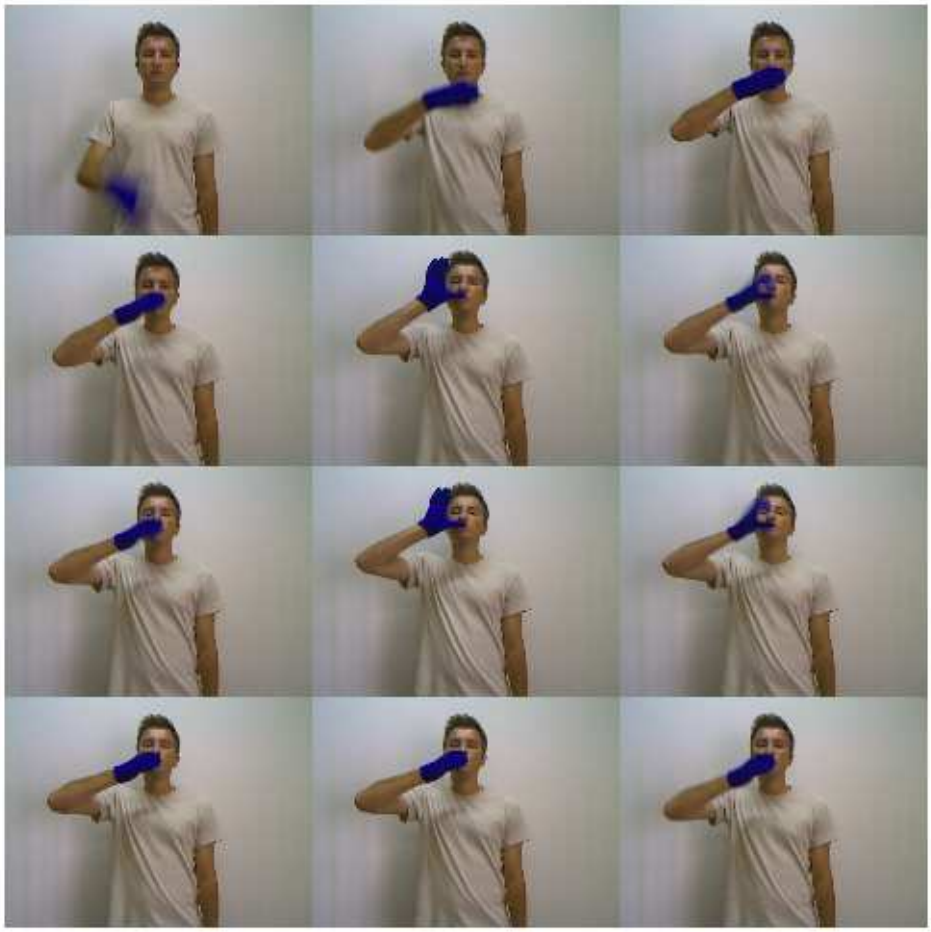

Figure A.4: Drink (noun) Sign, Signer: Ismail

A.5 Fast

Figure A.5: Fast Sign, Signer: Jakov

A.6 Here

Figure A.6: Here Sign, Signer: Levacic

A.7 Look at

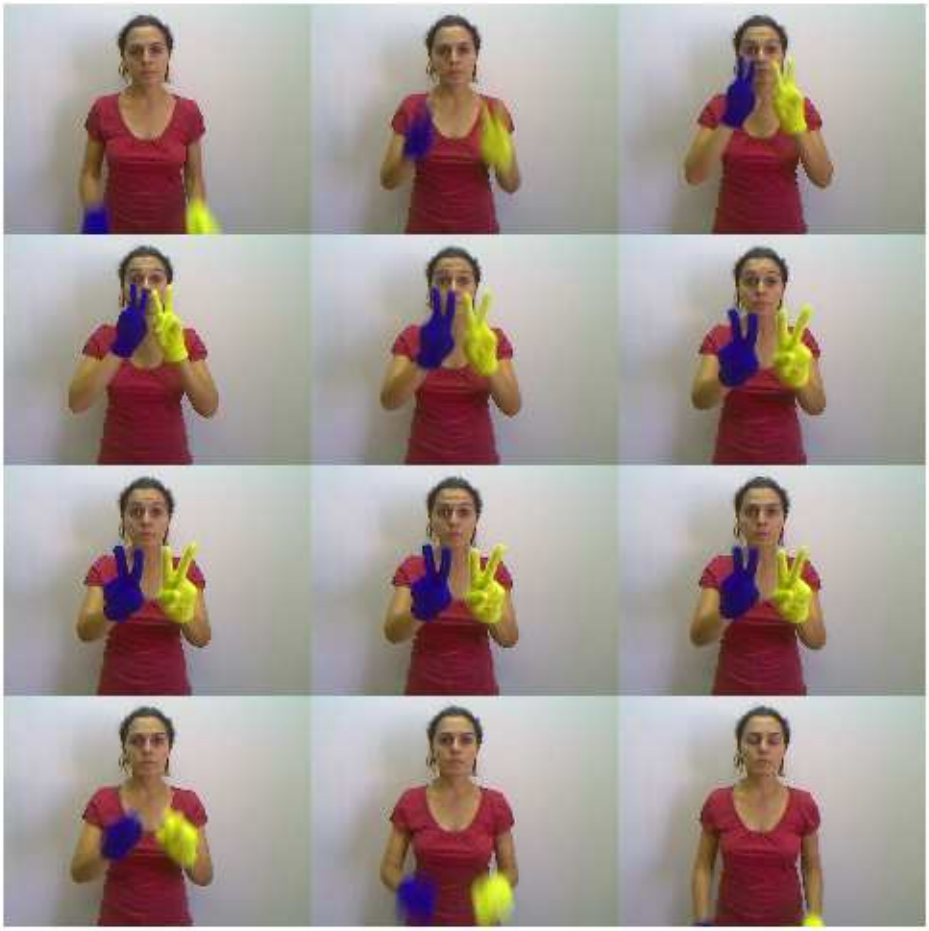

Figure A.7: Look at Sign, Signer: Oya

A.8 Study

Figure A.8: Study Sign, Signer: Pavel