ISOLATED SIGN LANGUAGE CLASSIFICATION USING HAND DESCRIPTORS AND TRAJECTORY BASED METHODS

by

Oğulcan Özdemir

B.S., Computer Engineering, Yıldız Technical University, 2013

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computer Engineering

Boğaziçi University

2018

ISOLATED SIGN LANGUAGE CLASSIFICATION USING HAND DESCRIPTORS AND TRAJECTORY BASED METHODS

APPROVED BY:

Prof. Lale Akarun (Thesis Supervisor)	.....
Prof. Murat Saraçlar	.....
Assist. Prof. Berk Gökberk	.....

DATE OF APPROVAL: 24.07.2018

ACKNOWLEDGEMENTS

First, and most of all, I would like to thank my thesis advisor Prof. Lale Akarun for her expertise, guidance, assistance and patience during my masters education. I would also like to thank Prof. Murat Saraçlar and Berk Görberk for participating in my thesis jury and their valuable comments and suggestions on my thesis.

I would like to express my deepest gratitude to my family for their endless support and encouragement throughout my life. Without them, I wouldn't be the person who I am today. I would especially like to thank my friend Necati Cihan Camgöz for believing in me and motivating me to pursue an academic career, and for supporting me for the last nine years.

I would like to thank my friends and colleagues in Boğaziçi University and Perceptual Intelligence Laboratory for their endless support and valuable friendship, namely, Ahmet Alp Kındıroğlu, Alper Kamil Bozkurt, Alptekin Orbay, Barış Evrim Demiröz, Barış Kurt, Berkant Kepez, Çağatay Yıldız, Doğa Siyli, Gaye Genç, Gizem Esra Ünlü, Mehmet Burak Kurutmaz, Orhan Sönmez, Yusuf Taha Ceritli, Ufuk Can Biçici, Uras Mutlu, Yiğit Yıldırım and Yunus Emre Kara. I would also like thank my friend Metehan Doyran for the sleepless nights that we were working before deadlines, and for all the fun we have had in the last three years.

This thesis has been supported by the Turkish Ministry of Development under the TAM Project, number 2007K120610.

ABSTRACT

ISOLATED SIGN LANGUAGE CLASSIFICATION USING HAND DESCRIPTORS AND TRAJECTORY BASED METHODS

In this thesis, we propose a sign language recognition system in which we have adapted and simplified the Improved Dense Trajectory (IDT) approach which was originally proposed for large-scale human action recognition problem. Since the sign language recognition problem mostly focuses on hand gestures, body posture and facial expressions, we have extracted IDT features and filtered the trajectories around the hand region by matching the trajectory coordinates with hand coordinates obtained by pose extraction. In addition to trajectory filtering, we also propose Hand Descriptors, a spatio-temporal feature extraction method, for sign language recognition. In our proposed method, we extract spatio-temporal descriptors around left and right hands. After descriptor extraction, we encoded each sign video as Fisher Vectors which were derived from a Gaussian Mixture Model which was estimated from the training descriptors. Then, we have trained Support Vector Machines to perform sign language classification using the Fisher Vectors as its inputs. We have conducted experiments on two subsets of the BosphorusSign dataset and evaluated the performance of the system in terms of feature extraction speed, computational complexity and memory requirement. In our experiments, the combination of all descriptors yields the best recognition performance on both subsets for both features. We have found that trajectory filtering approach yields a similar recognition performance to the baseline approach while the number of trajectories are drastically reduced. Moreover, we have analysed the effects of using different parameters and video resolutions on the performance of the Hand Descriptors. Our experiments have shown that hand region produces the most important features in our sign language classification system.

ÖZET

EL BETİMLEYİCİLERİ VE GEZİNGE TABANLI YÖNTEMLER KULLANARAK YALITILMIŞ İŞARET DİLİ SINIFLANDIRMA

Bu tezde, büyük ölçekli insan eylemi tanıma problemi için önerilmiş olan bir yöntem olan Geliştirilmiş Yoğun Gezingeler yöntemini uyarlayıp basitleştirdiğimiz bir işaret dili tanıma sistemi önerilmektedir. İşaret dili tanıma problemi çoğunlukla el hareketleri, vücut duruşu ve yüz ifadesine odaklandığı için, Geliştirilmiş Yoğun Gezingeler özniteliklerini çıkartıldıktan sonra elde edilen gezinge koordinatları poz çıkartma ile elde edilmiş olan el koordinatları ile eşleştirerek el çevresindeki gezingeler ayrılmıştır. Gezinge filtrelemesine ek olarak, bu çalışmada bir uzam-zamansal öznitelik çıkartma yöntemi olan El Betimleyicileri işaret dili tanıma için önerilmiştir. Önerdiğimiz bu yöntemde, sol ve sağ ellerin etrafından uzam-zamansal betimleyiciler elde edilmektedir. Betimleyiciler elde edildikten sonra, her bir işaret videosu, eğitim betimleyicilerinden elde edilen Gauss Karışım Modelinden türetilen Fisher Vektörler olarak tanımlanmıştır. Sonrasında, bu Fisher Vektörler işaret dili sınıflandırması yapmak için eğitilecek bir Destek Vektör Makinesine girdi olarak verilmektedir. Sistemi öznitelik çıkartma hızı, hesaplama karmaşıklığı ve bellek gereksinimi açısından değerlendirmek için BosphorusSign veri setinin alt kümeleri üzerinde deneyler yapılmıştır. Yapılan bu deneylerde, bütün öznitelik türlerinde, bütün betimleyicilerin beraber kullanıldığı durumda sistem en iyi tanıma performasını elde etmiştir. Yörünge filtreleme yöntemi temel yönteme yakın bir tanıma performansı verdiği gibi aynı zamanda yörüngelerin sayısını büyük bir ölçüde azaltmıştır. Ayrıca, farklı parametreler ve video çözünürlükleri kullanmanın El Betimleyicilerinin performansı üzerindeki etkisini analiz ettik. Bu çalışmada yaptığımız deneyler, el bölgesinden elde edilen betimleyicilerin işaret dili sınıflandıran sistemimiz için en önemli öznitelikleri barındırdığını göstermiştir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

a	Lagrange multiplier
C	Regularization parameter of SVM
D	Number of dimensions
f_t	Dense optical flow field at time t
g	Image intensity gradient
G_ϕ	Normalized gradient vector with respect to weights
G_μ	Normalized gradient vector with respect to mean
G_σ	Normalized gradient vector with respect to variance
k_M	Median filtering kernel
K	Total number of Gaussian components
$K(x, v)$	Kernel function given x and v
L	Trajectory length
$\mathcal{L}$	Log-likelihood
n_σ	Spatial dimension of the spatio-temporal grid
n_τ	Temporal dimension of the spatio-temporal grid
n_{cell}	Cell size of the descriptor
n_{block}	Block size of the descriptor
n_{bins}	Number of bins of the descriptor
n_t	Temporal cell size of the descriptor
$\mathcal{N}(x, \mu, \Sigma)$	Gaussian component
$p(X\|\Phi)$	Probability of X given parameters Φ
P_t	Sampled point at time t
S	Trajectory shape descriptor
T_L	Trajectory with the length L
$\|w\|$	Margin
X_D	D-dimensional local image descriptor
γ_{ik}	Posterior probability of component of x_i for k

ΔP_t	Displacement vector of the tracked point at time t
∇_Φ	Gradient vector with respect to Φ
θ	Orientation angle of the intensity gradient
μ	Mean
ξ	Soft margin error
σ_k	Variance of the Gaussian component k
Σ_k	Covariance Matrix
$\phi(x)$	Mapping function given x
ϕ	Mixture component weights
Φ	Estimated GMM parameters weights, mean and covariance

LIST OF ACRONYMS/ABBREVIATIONS

1D	One Dimensional
2D	Two Dimensional
3D	Three Dimensional
3D CNN	Three Dimensional Convolutional Neural Network
ASL	American Sign Language
Auslan	Australian Sign Language
BoF	Bag of Features
BoW	Bag of Words
CNN	Convolutional Neural Network
ConGD	Continuous Gesture Dataset
CSL	Chinese Sign Language
DT	Dense Trajectories
DTW	Dynamic Time Warping
ED	Encoder-Decoder
EM	Expectation Minimization
EU	Excluded User
FMMNN	Fuzzy Min-Max Neural Network
FV	Fisher Vector
GD	General Dataset
GMM	Gaussian Mixture Models
GSL	German Sign Language
HCRF	Hidden Conditional Random Fields
HD	Hand Descriptors
HMM	Hidden Markov Model
HOF	Histogram of Optical Flow
HOG	Histogram of Oriented Gradients
IDT	Improved Dense Trajectories
ILSVRC	ImageNet Large-Scale Visual Recognition Challenge

IsoGD	Isolated Gesture Dataset
k-NN	k-Nearest Neighbors
KSL	Korean Sign Language
LAP	Looking at People
LSTM	Long-short Term Memory
MBH	Motion Boundary Histograms
MEI	Motion Energy Image
MGO	Motion Gradient Orientation
MHI	Motion History Image
NMT	Neural Machine Translation
N/A	Not Available
OSD	Overlapping Subsequences Dataset
PCA	Principle Component Analysis
RANSAC	Random Sample Consensus
RBF	Radial Basis Function
RDF	Random Decision Forest
RGB	Red Green Blue
RGB-D	Red Green Blue Depth
RNN	Recurrent Neural Network
SIFT	Scale Invariant Feature Transform
SL	Sign Language
SLR	Sign Language Recognition
STIP	Space-time Interest Points
SURF	Speeded Up Robust Features
SVM	Support Vector Machine
TaSL	Taiwanese Sign Language
TSL	Turkish Sign Language
TT	Temporal Templates
VLAD	Vectors of Locally Aggregated Features

1. INTRODUCTION

Sign languages are the main mode of communication of the hearing impaired. However, lack of sign language knowledge in our society and low number of sign language translators and tutors create communication problems for deaf people. These communication problems hamper the inclusion of hearing impaired with the society in their daily lives. Each society has its own sign language and there are as many sign languages as there are deaf societies in the world. To eliminate these communication problems, researchers have been developing automatized real-time sign language recognition systems. In order to recognize signs in these systems, attributes such as hand shape, hand movements and facial expressions of the user making a sign captured using glove-like sensors or vision-based cameras.

In the studies on sign language recognition, features that have been used in action recognition have been employed to represent hand motion and appearance. Since these action recognition methods have been proposed for more diverse set of human movements performed in a general background, most of them cannot be used in real-time recognition systems. Therefore, the development of fast alternative methods for real-time systems are needed.

Sign language recognition and human action recognition problems involve fundamental differences even if they are seen as similar problems by researchers because they involve finding the movement of the human in videos. While the human action recognition problem relies on human movements, human interactions, camera movement and background knowledge, short and consecutive movements of the person making the sign are conveyed by the signer's hand movements and facial expressions in sign language recognition. Also, in sign language videos, the user is mostly stationary and the most important information is derived from the shape and the position of the hand of the user. Due to these reasons, it is expected that a method designed specifically for the

sign language recognition problem for real-time recognition systems will be successful in terms of performance and speed.

Sign language recognition research is nearly 40 years old. Earlier studies have focused on recognizing isolated signs. Recent studies employ news for the hearing impaired data to train continuous sign language systems. However, currently, the success of these systems are limited. Recent studies have employed RGB-D data and have applied deep learning techniques.

1.1. Contributions

In this thesis, we have made the following contributions;

(i) Adaptation of the Improved Dense Trajectories (IDT) approach into sign language recognition domain
(ii) Introduction of a computationally inexpensive spatio-temporal feature extraction method, the Hand Descriptors
(iii) In-depth complexity analysis of both Improved Dense Trajectories and Hand Descriptors on sign language classification

First, we have adapted Improved Dense Trajectories (IDT) [14], which were first proposed for recognizing actions of humans in videos, into the sign language recognition domain. Improved Dense Trajectories approach, proposed by Wang *et al.* [14], represent the videos as trajectories by sampling the dense optical flow. After dense sampling, spatio-temporal descriptors such as Histogram of Oriented Gradients (HOG), Histogram of Optical Flow (HOF) and Motion Boundary Histograms (MBH) are extracted from each trajectory tracked.

The motivation of using the IDT features in sign language recognition is that IDT can characterize upper body movements and hand shapes which are commonly

used traits in sign language recognition. Although the IDTs were not popular when recognizing the sign languages, they were used in gesture recognition [15].

Improved Dense Trajectory (IDT) features are computationally complex and high dimensional representations. Hence, obtaining these features in a real-time system will make the system more difficult to train and test. In this study, we presented an alternative solution to Improved Dense Trajectories for sign language recognition domain. In our experiments, we have seen that IDT method extracts more trajectory than needed for a problem like sign language recognition where information is present in a part of the frame instead of the entire frame.

In the proposed method, we first filtered the IDT features by matching the trajectory tracking coordinates with the pixel coordinates of both hands which were obtained via Microsoft Kinect. Even though filtering the trajectories does not speed up the extraction process, we have seen that we can still reach a similar prediction accuracy with less trajectories. Secondly, we extracted HOG, HOF and MBH descriptors from the crops obtained from the hand regions of signer performing the sign.

In order to train these descriptors, we first apply Principle Component Analysis (PCA) normalization and estimate Gaussian Mixture Models (GMM) which is then used to extract Fisher Vector (FV) from each video in the training and the test set. After extracting FVs, we train linear Support Vector Machines (SVMs) and test our methodology. In our experiments, we used two different datasets which are subsets of BosphorusSign Turkish Sign Language Dataset. While one of these subsets is much smaller than the other, it contains 399 sign videos of 10 classes in which each video has a RGB video, a depth map and a skeleton information which was recorded via Microsoft Kinect. Second subset of the BosphorusSign consists of nearly $6,000$ videos of 154 sign classes which has the same recording setup as the first subset.

In our experiments, we have seen that extracting these descriptors around hand regions has sped up the feature extraction process while yielding similar accuracy on the

small subset. However, increasing the size of dataset has resulted decrease in accuracy. In addition, extraction speeds of the descriptors, training-testing times, feature sizes and thus the model complexity were reduced as expected.

1.2. Organization of the Thesis

The rest of the thesis is organized as follows: In Chapter 2, we review the literature on both sign language and action recognition. We also investigate the popular datasets for both domains in the same chapter. In Chapter 3, we explain the background of the spatio-temporal descriptors, normalization, clustering and encoding methods, and SVM classifiers. After we explain their background information, we describe usage of these methods in our proposed methodology in Chapter 4. In Chapter 5, we present the experiments that we have performed to recognize sign language videos, and discuss their results. Lastly, we share our final thoughts about our work and give some future directions in Chapter 6.

2. RELATED WORK

2.1. Sign Language Recognition

Sign language recognition research started in the 1990's and different approaches have been applied since then. While some researchers have treated the problem as a gesture recognition problem in which signs are assumed as well-defined gestures, others have tried to adapt known speech or action recognition techniques because recognizing a sign is a sequence prediction problem and very similar to these problems. These approaches can mainly be divided into three different categories which can be listed as wearable sensor based, feature based and deep learning based approaches. Although wearable sensor based approaches were popular in the earlier work on sign language and gesture recognition, they lost their importance over time when vision based sensors such as RGB and depth cameras have become more popular and available to researchers.

Earlier research on sign language recognition presented solutions in which wearable sensors like data gloves and accelerometers were used. With these sensors, researchers have collected sign or gesture datasets which contain important measurements such as hand coordinates and velocity. Usage of these sensors allowed researchers to collect discriminative information about a sign or a gesture. In the work of Kadous [1], instance-based and decision tree learning were performed on features extracted from Australian Sign Language (Auslan) dataset, which was collected using a PowerGlove (see Figure 2.1). Similarly, Vogler and Metaxas [16] used a magnetic sensor system to collect 3D wrist position and orientation information. In the same work authors proposed a Hidden Markov Model (HMM) based sign language recognition on a American Sign Language (ASL) dataset that they have collected using magnetic sensors. In another work, Hernández-Rebollar *et al.* [2] presented a hierarchical finger spelling translation system which uses a glove sensor called Accele Glove (see Figure 2.1) to measure finger positions of the hand shapes of American Sign Language (ASL).

Figure 2.1. Sensors used in earlier research on sign language recognition. Left: PowerGlove [1], Right: Accele Glove [2]

DataGloves have also been used for research related to sign language and gesture recognition. Liang and Ouhyoung [17] used these sensors for acquiring posture information on Taiwanese Sign Language (TaSL). In their work, they tried to recognize sequences of posture data in real-time using Hidden Markov Models. In [18], Kim *et al.* have used a pair of DataGloves to collect their dataset from a small sign set of Korean Sign Language (KSL). They used the dataset in a system which was able to recognize and translate a small sign set of Korean Sign Language (KSL) signs using Fuzzy Min-Max Neural Networks (FMMNNs) [19].

Although wearable sensor-based sign language recognition was popular in the earlier years, sensors used in these studies were expensive and needed calibration for each user. Because of these problems and technological developments on RGB and depth cameras, computer vision based methods have become more popular among researchers. These methods have been based on several different stages which can be listed as feature extraction, temporal modelling and classification.

In the systems using color-based cameras, finding the signer in the frame and estimating the upper body pose of the signer are still hard problems because of the color ambiguity in the frames. In addition, color-based visual sensors are heavily influenced

by the illumination of the environment. In order to find the signer and the signer's pose in the frame, several researchers have proposed methods [3,20,21] in which signers wear coloured gloves which helped the researchers to retreieve real-time information about the hand shape and its position from the performed sign. An example of colored glove used in the work of Zhang *et al.* [3] can be seen in Figure 2.2. Beside these studies, researchers have also worked on skin color based segmentation [22–26] for finding the hand and the face of the signer in the frame.

Figure 2.2. Colored gloves used in Zhang *et al.* [3] for sign language recognition

Researchers have started using depth cameras like Microsoft Kinect [27] in the past several years. The ability to estimate the pose of the signer with these depth cameras has opened a new research area for researchers because depth maps provided by these cameras made it easier to separate signers from the background. Furthermore, Shotton *et al.*'s work on Random Decision Forest (RDF) [28] based upper body pose estimation has enabled researchers to use skeletal information extracted from the signer in the frame in real-time [29].

In every sign language, users use hand shapes, hand movements, position of their hands, pose of their upper bodies and facial expressions in order to communicate with each other. In a similar way, sign language recognition systems can extract descriptive information from performed signs and can use them to recognize new signs. Considering feature extraction, researchers have published many different methods in the

literature on sign language and gesture recognition. To detect hands and faces in a video, Kadir *et al.* [30] have trained weak classifiers with AdaBoost algorithm [31] using Haar wavelet-like features proposed by Viola and Jones [32] as its inputs. In another work, Liu *et al.* used depth and grayscale images to find hand trajectories and orientation for gesture recognition [33]. Wong and Cipolla [34] have used Motion Gradient Orientation (MGO) [35] method which was proposed by Bradski and Davis as an extension of Motion History Image (MHI) [36] and Motion Energy Image (MEI) [37].

To acquire more descriptive information about the performed sign or gesture, more complex visual features have been extracted by researchers. For example, Nandakumar *et al.* [38] extracted Space-time Interest Points (STIP) [39] features from RGB gesture videos in a multi-modal framework where audio and skeleton information were used. Liwicki *et al.* [26] and Camgoz *et al.* [40] similarly used Histogram of Oriented Gradients (HOG) [41] features to retrieve hand shape descriptors after obtaining hand crops. More recently, Peng *et al.* [15] proposed a gesture recognition framework where Improved Dense Trajectory (IDT) [14] features are extracted for temporal spotting in the Chalearn LAP challenge [6].

In sign language recognition, modelling temporal changes of extracted features is important for recognizing the sign because these languages, like every other spoken language, can be modeled as time series. Hidden Markov Models (HMMs) [42] and Dynamic Time Warping (DTW) [43] are often used in sign language and gesture recognition problems. Starner and Pentland [44] used HMMs in real-time continuous sign language recognition where signs from American Sign Language are recognized at the sentence-level. Vogler and Metaxas [45] has also showed that HMMs are suitable for word-level American Sign Language recognition. In addition, DTW based methods have been used for temporal modelling and alignment of gesture and sign language in the works of Chai *et al.* [46], Keskin *et al.* [47] and Camgoz *et al.* [48].

As for the classification step, template matching [23,33,49], k-Nearest Neighbours (k-NN) [50] and Bayesian classifiers [34, 51] are often used in earlier research on sign

language and gesture recognition problem. More recently, Camgoz *et al.* [52] used template based Random Decision Forest (RDF) [53] models to predict skeleton based features extracted from continuous human gestures. In a different work of the same authors [40], temporally modelled upper body pose, hand shape and hand position features of isolated sign phrases are predicted using k-Nearest Neighbours (k-NN) and Random Decision Forests (RDF).

Recently, researchers have started to use deep learning learning approaches which are successful on problems such as image classification and segmentation and recently used in temporal problems such as action and speech recognition. Pigou *et al.* [54] have investigated the idea of using Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) on gesture recognition by examining different CNN architectures which contain temporal pooling, temporal convolutions and RNNs. Kang *et al.* [55] have proposed an approach where they tried to recognize static finger spelling in American Sign Language (ASL). In their work, they have trained a CNN using depth images of fingerspelling signs of the fingerspelling alphabet. In [56], Koller *et al.* proposed an approach (DeepHand) to train frame-based classifiers by using CNN embeddings within Hidden Markov Models (HMMs). In the same work, they were also able to train a fine-grained CNN model for single frame hand shape recognition using nearly one million hand images.

3D Convolutional Neural Networks (3D CNNs) were also used in sign and gesture recognition problems in the recent studies. Molchanov *et al.* [57] used RGB and depth images to train 3D CNNs for hand gesture recognition where they trained two subnetworks; a high-resolution and a low resolution networks which are then fused into a single layer for classification. In another work, Camgöz *et al.* [58] proposed an end-to-end deep learning framework that was used in continuous gesture recognition in 2017 Chalearn LAP Challenge. In their work, they obtained class probabilities for each video by applying a sliding windows approach to 3D CNNs.

In the work of Neverova *et al.* [59], a multi-modal approach was proposed for detecting and recognizing gestures. In the same study, authors trained an individual classifier for each different modality; depth map, skeleton information and audio features, which are then merged as long gesture sequences using Recurrent Neural Networks (RNNs). In another work of Neverova *et al.* [60], a similar multi-modal approach has been proposed for gesture recognition where they extracted features from four different individual classifiers that are specifically trained for different modalities; intensity, depth, audio and pose information. These features are then fused using two fully connected layers.

Aside from sign language recognition, researches have also focused on the sign language translation problem which is based on the idea of generating spoken language translations from sign language videos. Recently, Camgöz *et al.* [61] followed this idea and proposed a sign language translation framework based on Neural Machine Translation (NMT) [62]. In their work, they have tried to translate from German Sign Language (GSL) videos of weather broadcasts to German sentences using their NMT based framework. Summary of the sign language literature mentioned in this section is listed in Table 2.1.

2.1.1. Sign Language Datasets

In the context of gesture and sign language recognition, several datasets exist in the literature. These datasets mainly focused on having large vocabularies and multiple repetitions of the performed sign. Most of them have multiple signers for each sign sample to acquire signer-independent features. With the handiness of depth cameras like Microsoft Kinect v2 [27], researchers have collected datasets containing upper body pose information which are extracted using algorithms such as in [29].

One of the most extensive is the RWTH-PHOENIX [4] dataset, which has been collected from weather forecasts and daily news of the German television channel PHOENIX. In the dataset, there are $6,231$ sentence-level continuous sequences of

Table 2.1. Summary of the literature review on Gesture and Sign Language recognition that mentioned in Section 2.1.

Author	Year	Category	Goal	Method	Features	Scope	Dataset	Language	Sensor Input/Type
Kadous *et al.* [1]	1996	Sensor	SLR	Instance-based Decision Trees	Hand and finger positions	Isolated	N/A	Auslan	PowerGlove signal
Kim *et al.* [18]	1996	Sensor	SLR	FMMNN	Hand and finger positions	Isolated	N/A	KSL	DataGlove signal
Vogler and Metaxas [16]	1997	Sensor	SLR	Hidden Markov Models	3D wrist pose	Isolated	N/A	ASL	Magnetic sensor signal
Starner and Pentland [44]	1998	Vision	SLR	HMMs	Hand Shape	Continuous	N/A	ASL	RGB
Hernández-Rebollar *et al.* [2]	2002	Sensor	FSR	Hierarchical Tranlation	Hand and finger positions	Isolated	N/A	ASL	Accele Glove signal
Zhang *et al.* [3]	2004	Vision	SLR	HMMs	Hand shape	Isolated	N/A	CSL	RGB frame and colored glove
Kadir *et al.* [30]	2004	Vision	SLR	Adaboost	Haar wevelet-like features	Isolated	N/A	BSL	RGB
Liu *et al.* [33]	2004	Vision	GR	Template Matching	Hand trajectories	Isolated	N/A	-	RGB-D
Wong and Cipolla [34]	2005	Vision	SLR	Sparse Bayesian Classifier	MGO images	Continuous	N/A	ASL	RGB
Liwicki *et al.* [26]	2009	Vision	FSR	HMMs	Histogram of Oriented Gradients	Isolated	N/A	BSL	RGB
Nandakumar *et al.* [38]	2013	Vision	GR	HMMs	STIP, audio and skeleton	Continuous	Montalbano v1	-	RGB-D,audio and skeleton
Neverova *et al.* [59]	2013	Deep learning	GR	RNNs	Depth, skeleton and audio	Isolated	Montalbano v2	-	RGB-D, skeleton and audio
Peng *et al.* [15]	2014	Vision	GS	Multi-class Linear SVMs	Super Vectors	Continuous	Montalbano v2	-	RGB-D and skeleton
Camgoz *et al.* [52]	2014	Vision	GR	TT based RDFs	Upper body joint coordinates	Continuous	Montalbano v2	-	Skeleton
Kang *et al.* [55]	2015	Deep learning	FSR	CNN	Depth images	Isolated	N/A	ASL	RGB-D
Molchanov *et al.* [57]	2015	Deep learning	GR	3D CNN	RGB and depth images	Isolated	VIVA	-	RGB-D
Koller *et al.* [56]	2016	Deep learning	SLR	CNN + HMM	RGB images	Continuous	RWTH-PHOENIX	GSL	RGB
Camgöz *et al.* [40]	2016	Vision	SLR	TT based RDFs	HOG	Isolated	BosphorusSign	TSL	RGB and skeleton
Camgöz *et al.* [58]	2016	Deep learning	GR	3D CNN	RGB images	Continuous	Montalbano v2	-	RGB
Neverova *et al.* [60]	2016	Deep learning	GR	CNN	Depth, skeleton and audio	Continuous	Montalbano v2	-	RGB-D,audio and skeleton
Pigou *et al.* [54]	2018	Deep learning	GR	CNN + RNN	RGB images	Continuous	Montalbano v2	-	RGB
Camgöz *et al.* [61]	2018	Deep learning	SLT	CNN + ED Networks	RGB images	Continuous	RWTH-PHOENIX	GSL	RGB

1,231 RGB sign videos which were collected from 7 different native signers of German Sign Language. Each video in the dataset has been recorded at 210×260 pixels and 25 frames per second. Examples from the RWTH-PHOENIX dataset can be seen in Figure 2.3. As one of the datasets with a large vocabulary, DEVISIGN dataset [5]

Figure 2.3. Example signs from RWTH-PHOENIX [4] German Sign Language dataset.

has been recorded by Kinect Sign Language Working Group. This publicly available dataset contains 24,000 isolated sign videos (3 to 6 seconds long) of 2,000 Chinese Sign Language (CSL) words, which were collected from 8 different users (repeated 4 users). Similar to most datasets, this dataset also collected using Microsoft Kinect v1 [27] and contains body pose information, RGB video and depth mask for each video (see Figure 2.4).

Figure 2.4. An example sign video from DEVISIGN dataset [5]. Left to right; RGB video, visualized depth map and body pose information

More recently, BosphorusSign dataset has been published by Boğaziçi University's Perceptual Intelligence Laboratory [63]. This dataset is divided by three different categories of sign in terms of their content. These categories are health, finance and commonly used words. Each sign sample in this dataset, recorded using Microsoft Kinect v2 [27], contains RGB video, depth map, user mask and skeletal upper body pose information. In the dataset, there are nearly 24,000 isolated recordings of 636 sign classes which were collected from 6 different signers (see Figure 2.5). In the thesis, two separate subsets are used to perform the proposed sign language recognition approach.

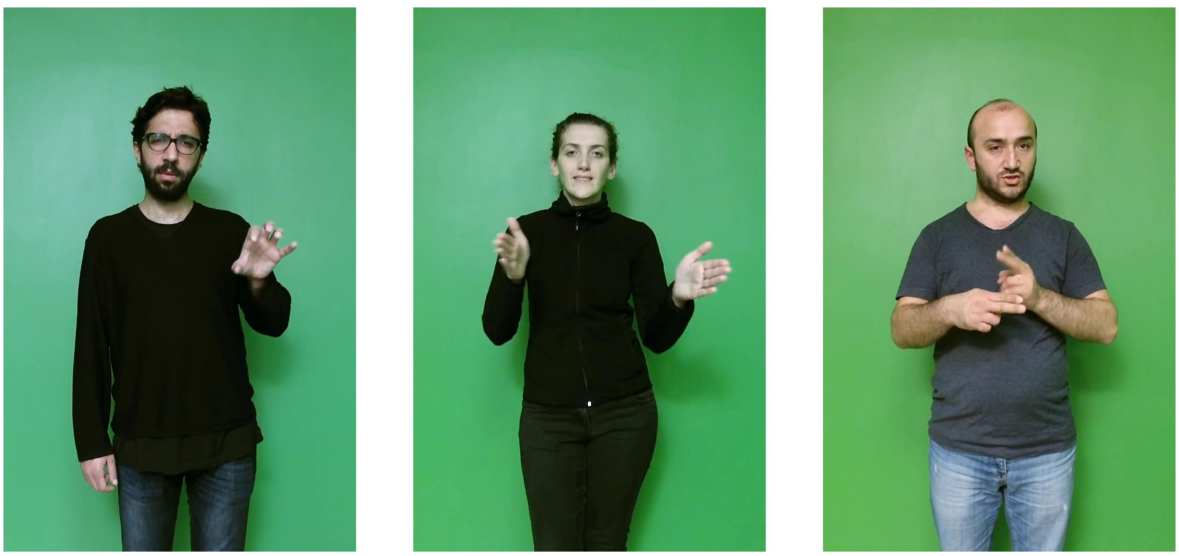

Figure 2.5. Examples signs from BosphorusSign Dataset. Left to right: Pain, Address and Urgent.

Apart from sign language datasets, there are gesture recognition dataset e which are used in the literature. Montalbano v2 dataset which was first published in Chalearn Looking at People (LAP) challenge for the problem of user independent gesture spotting [6]. This dataset contains 13,858 sentence-level gesture videos of 20 gesture classes which were selected from popular Italian cultural/anthropological signs (see examples in Figure 2.6). In the dataset, each gesture class is performed by 27 different signers. In addition, each video in the dataset includes RGB-D and upper body pose information extracted via Microsoft Kinect v1 [27].

Figure 2.6. Example Italian cultural/anthropological gestures from the Montalbano v2 dataset [6].

Moreover, for the large-scale user-independent gesture recognition, Chalearn LAP RGB-D Gesture Dataset has been reproduced from videos of the Montalbono v2 dataset (Chalearn Gesture Dataset) for researchers in 2016 Chalearn LAP challenge [64]. The dataset were published with isolated (IsoGD - Isolated Gesture Dataset) and continuous (ConGD - Continuous Gesture Dataset) subsets which can used be used in different subtasks. In ConGD, there are $22,535$ continuous RGB-D gesture videos of 249 gesture classes. Each video in this dataset includes more than one gestures which are performed by 21 different users. Differently, as a reuse of ConGD, IsoGD has been built by trimming and manually labelling videos from the ConGD, and contains $47,933$ gesture videos performed by the same 21 individuals. Table 2.2 summarizes the discussed datasets in terms of number of classes, number of users, and other details.

2.2. Human Activity Recognition

Human activity recognition has been a popular research area among computer vision researchers since the 1980s. In earlier work on this problem, researchers have aimed to recognize simple activities such as walking, running, jumping which are activities that involve one or more person. Recognizing such activities from videos or static images is important for human-computer interaction applications because it enables us to learn information about the person who is performing the activity in the frame.

Table 2.2. Dataset used in the literature for Gesture and Sign Language Recognition

Dataset	Year	Context	# Classes	# Samples	# Users	Type	Modality
RWTH-Phoenix [4]	2012	German SL	1,231	6,931 Sentences	7	Continuous	RGB
Montalbano v2 [6]	2014	Italian cultural signs	20	13,858 Sentences	27	Continuous	RGB-D + Skeleton
DEVISIGN [5]	2015	Chinese SL	2,000	24,000 Samples	8	Isolated	RGB-D + Skeleton
Chalearn LAP GD [64]	2016	General gestures	249	47,933 Samples	21	Isolated + Continuous	RGB-D
BosphorusSign [63]	2016	Turkish SL	636	24,161 Samples	6	Continuous	RGB-D + Skeleton

This extracted information could be the identity of the person, their personality or even their behavioural state. Moreover, this information can be used in many applications including human-computer interaction, video surveillance or video retrieval systems.

In the work on human activity recognition, researchers have followed similar tasks in order recognize activities in videos or still images. These tasks include capturing the frame, segmentation, feature extraction, tracking and classification. These tasks are challenging due to difficulties such as cluttered background, occlusions, changes in appearance or different lightning effects. Because of these difficulties, researchers have tried different techniques for human detection, human tracking and background subtraction to overcome these problems in order to recognize human activities in videos or still images.

Since the development of vision-based sensors, most of the human activity recognition approaches have been using RGB video cameras. These approaches often use spatio-temporal features and trajectory matching methods which can give us low-level features and 3D body representations to analyse human motion in videos.

Bobick *et al.* [37] have proposed Motion Energy Images (MEI) which represent the action as a cumulative distribution of the motion energy extracted from the spatial regions in activity videos. In a subsequent work, Bobick and Davis [36] have introduced Motion History Images (MHI) for representing human actions as functions of motion history where the actions are performed. More recently, Dalal *et al.* [41] proposed Histogram of Oriented Gradients (HOG) as a descriptor for detecting humans or objects in the frame by describing them as distributions of intensity gradients.

Furthermore, most action recognition approaches rely on optical flow, which is a method for calculating the apparent motion difference between two frames [65]. Laptev *et al.* [66] have proposed Histograms of Optical Flow (HOF) method which uses the optical flow to recognize human activities by representing them as a distribution of the motion difference between frames. Moreover, Dalal *et al.* used Motion Boundary

Histograms (MBH) where actions are represented by histograms of the derivatives of horizontal and vertical components of the optical flow [67].

In the literature, these descriptors were not used alone, but their fusions were also used in the past. In the work of Wang *et al.* [14], Improved Dense Trajectory approach has been proposed for understanding human actions in videos by extracting HOG, HOF and MBH descriptors and concatenating them together as a single feature descriptor of trajectories obtained from the dense optical flow. Then, these feature descriptors are encoded as Fisher Vectors (FV) and classified using Support Vector Machine (SVM) classifier. Fisher Vectors (FV) method has been proposed by Sánchez *et al.* [68] to classify images and videos. This method has also been proposed as an alternative feature pooling technique for well-known Bag-of-Words (BoW) [69] pooling technique.

Gaidon *et al.* have proposed an unsupervised approach which is based on performing hierarchical clustering on spatio-temporal features extracted from videos [70]. In another work, Vrigkas *et al.* [71] have used an approach where they cluster the motion trajectories using Gaussian Mixture Models (GMM) and label them using a Nearest Neighbor Classifier.

Stochastic methods have also been used in human activity recognition problem where any action can be considered as a collectiong of predictable sequences. These type of methods provide researchers with an understanding of more complex activities and non-periodic actions by applying statistical models. Most of the research uses stochastic methods frequently uses Markov Models (HMM) [42] and Hidden Conditional Random Fields (HCRF) [72]. Hidden Conditional Random Fields were first proposed by Wang and Mori for action recognition task [73]. In their work, they have modelled the motion of human action by using the observations of the image parts which are then represented as BoW features.

Estimating the pose of a human in a frame is also an important for extracting more discriminative features. To estimate the pose of the person in the frame, researchers have proposed shape-based approaches to describe human activities by modelling the motion of the human body parts. These methods are based on the importance of pose and appearance which can be extracted from human silhouettes. In the work of Maji *et al.* [74], pose estimation model based on head and torso orientation has been proposed to extract representations of human pose. Rahmani *et al.* [75] have proposed a pose estimation model for online human action by training Random Decision Forests (RDF) using the joint information of human body which are extracted from a 3D depth sensor.

In addition to these earlier studies on human activity recognition, recent studies have focused on deep learning based methods. After the success of Krizhevsky *et al.* [76]'s work on image classification in ImageNet Large-Scale Visual Recognition Challenge 2012 (ILSVRC) [77], Convolutional Neural Networks (CNNs) [78] have become more popular among computer vision researchers and used in many different tasks such as image classification and segmentation.

Deep learning methods have gained interest with the development of faster hardware and GPU implementations. Many researchers have been using deep neural networks to detect and recognize complex activities on videos or static images since these developments. In the area of human activity recognition, Karpathy *et al.* [79] used Convolutional Neural Networks (CNNs) to classify activities from large video datasets. In another work, Donahue *et al.* [80] used a combined version of CNNs and Recurrent Neural Networks with LSTM units (Long-short Term Memory) [81] to describe complex human actions in long videos.

More recently, Simonyan and Zisserman [82] have proposed Two-Stream Convolutional Networks which is a deep neural network model based on the two-stream hypothesis [83] which supports the idea that human visual cortex contains two separate pathways; dorsal (motion) and ventral (appearance) streams. In their implementation

of this hypothesis, they used still frames for modeling the spatial (ventral) stream and stacked optical flow images for modeling the motion (dorsal) stream which are extracted randomly from videos of UCF-101 [10] and HMDB-51 [84] action recognition datasets. Followed by this work, researchers have applied the idea of fusing multiple networks which have different modalities. Fusing audio spectrogram with spatial and temporal stream [85], pooling hand-crafted IDT with CNN features [86] or temporal pooling of CNN features with Long-Term Short Networks (LSTM) [87] can be seen as some of these methods which are used in multi-modal fusion.

Alternatively, Tran *et al.* [88] have extended 2D Convolutional Network with 3D convolutions in order to train a model which can learn spatio-temporal dependencies on large scale supervised video datasets. In their work, they have trained the model using small volumes cropped from videos as inputs for mini-batches. In one of the most recent work, Carreira *et al.* [89] have proposed Two-Stream Inflated 3D Convolutional Networks model which uses both Two-Stream and 3D CNN ideas. In their work, they have changed the convolution and pooling layers of Google's Inception-v1 [90] model, which was pre-trained on ImageNet dataset, and then trained two separate models for spatial and temporal streams on the Kinetics dataset. Summary of the literature review of the human activity recognition is in Table 2.3.

2.2.1. Human Activity Recognition Datasets

In the literature, datasets used for the purpose of recognizing human activities have various characteristics in terms of size, resolution, action type and environment. A summary of frequently used datasets for human activity recognition studies are given in Table 2.4.

One of the earliest datasets, the KTH dataset [7] contains six different actions (walking, running, boxing, hand waving, hand clapping and jogging) which are collected from 25 different people with repetitions (see Figure 2.7). In the dataset, actions are

Table 2.3. Summary of the literature review on Human Activity Recognition that mentioned in Section 2.2.

Author	Year	Representation Type	Goal	Method	Features	Dataset	Input Type
Dalal *et al.* [41]	2005	Handcrafted	Human Detection	Linear SVMs	Histogram of Oriented Gradients (HOG)	INRIA and MIT Pedestrian	RGB
Dalal *et al.* [67]	2006	Handcrafted	Human Detection	Linear SVMs	Motion Boundary Histograms (MBH)	INRIA	RGB
Laptev *et al.* [66]	2008	Handcrafted	Action Recognition	Non-linear SVMs	Histogram of Optical Flow (HOF)	Hollywood	RGB
Wang and Mori [73]	2009	Handcrafted	Action Recognition	Max-Margin HCRF	Optical flow fields	Weizmann and KTH	Grayscale
Maji *et al.* [74]	2011	Handcrafted	Human Pose Estimation	Linear SVMs	Poselet Activation Vector	PASCAL VOC 2010	RGB
Wang *et al.* [14]	2013	Handcrafted	Action Recognition	Linear SVMs	FVs from HOG, HOF and MBH	UCF-101 and HMDB51	RGB
Gaidon *et al.* [70]	2014	Handcrafted	Action Recognition	Hierarchical clustering + SVMs	BOF trees of dense tracklets	Hollywood 2 and HMDB51	RGB
Vrigkas *et al.* [71]	2014	Handcrafted	Action Recognition	Nearest Neigbor Classifier	Motion trajectories	UCF Sports and Youtube	RGB
Rahmani *et al.* [75]	2014	Handcrafted	Human Pose Estimation	Random Decision Forests	Histograms of Depth Gradients	MSR Action 3D	RGB-D
Karpathy *et al.* [79]	2014	Deep	Action Recognition	CNN	-	UCF-101	RGB
Simonyan *et al.* [82]	2014	Deep	Action Recognition	Two-stream CNN + SVM	Optical flow and RGB frames	UCF-101 and HMDB51	RGB
Donahue *et al.* [80]	2015	Deep	Action Recognition	CNN + LSTM	-	UCF-101 and HMDB51	RGB
Wang *et al.* [86]	2015	Deep	Action Recognition	CNN	IDT features as CNN inputs	UCF-101 and HMDB51	RGB
Tran *et al.* [88]	2015	Deep	Action Recognition	3D CNN + SVM	-	UCF-101	RGB
Carreira *et al.* [89]	2017	Deep	Action Recognition	Two-stream Inflated 3D CNN	Optical flow and RGB frames	UCF-101, HMDB51 and Kinetics	RGB

Table 2.4. A summary table of commonly used human activity recognition datasets.

Dataset	# Classes	# Videos	Resolution (Dimensionality)	Modality	Year
KTH [7]	6	600	160×120	Grayscale	2004
UCF-Sports [8]	10	150	720×480	RGB	2008
HMDB51 [9]	51	$6,849$	320×240	RGB	2011
UCF-101 [10]	101	$13,320$	320×240	RGB + Audio	2012
Youtube 8M [11]	$3,862$	61 million	1024-dim rgb 128-dim audio	RGB + Audio	2016
Kinetics [12]	600	$500,000$	320×240 640×360	RGB + Audio	2017

collected as grayscale videos in open and closed environments at the resolution of 160×120 and 25 frames per second.

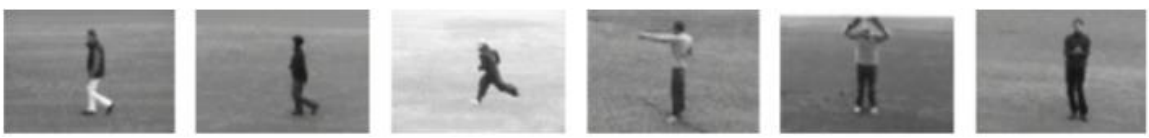

Figure 2.7. Example action classes from the KTH dataset [7]. Left to right; walking, jogging, running, boxing, hand waving, hand clapping

The UCF-Sports dataset contains various sports related videos which were collected from television channels such as BBC and ESPN [8]. In this dataset, there are 150 different sport videos with 10 sport action at the resolution of 720×480 and 10 frames per second (see Figure 2.8). This dataset is often used for human action recognition, action localization and saliency detection problems.

Figure 2.8. Example action classes from the UCF-Sports dataset [8]. Left to right; diving side, golf swing, lifting, riding horse

The HMDB51 dataset (Human Motion Database) [9] contains 6,849 videos of 51 different action classes of five categories which can be listed as facial actions, facial action with object manipulations, body movements, body movements with object interactions and body movement for human interactions. In addition, action videos in the dataset were collected from digitized popular movies, Youtube and Google videos (see Figure 2.9).

Figure 2.9. Example action classes from each categories of the HMDB51 dataset [9]. Left to right: climb, kick ball, shake hands, smile and smoke.

The UCF-101 dataset [10] has been collected as an extension of UCF-50 dataset containing 101 different actions which are collected from Youtube. Examples from this dataset are shown in Figure 2.10. This dataset has challenges such as camera movement, changes in pose and appearance and changes in light and viewpoint. Due to their characteristics of adding more realistic traits to the actions, these challenges have an important impact on solving the problem of human action recognition. Therefore, the UCF-101 and the HMDB datasets are often used in challenges related to human activity recognition.

Figure 2.10. Example action classes from the UCF-101 dataset [10]. Left to right: apply eye makeup, biking, ice dancing and writing on board.

The Youtube 8M dataset [11] has published as a benchmark dataset which can be used for video understanding. The dataset has diverse classes such as sports, food, animals and products (see Figure 2.11). It has around 6.1 million videos of $3,862$ classes with an average of 3 labels per video. Due to the size of the dataset which is around $350,000$ hours of video, authors have also published extracted features of RGB and audio modalities for the researchers.

Figure 2.11. Example action classes from the Youtube 8M dataset [11]. Left to right: basketball, cartoon, concert and cooking.

Recently, Kinetics dataset [12] has been proposed by Google's DeepMind team for large-scale human activity recognition. The dataset contains nearly $500,000$ video of 600 action classes which are collected from Youtube (see examples in Figure 2.12). Kinetics dataset also includes action classes such as hand shake, riding a bike and playing trumpet actions which can be considered as human-object and human-human interactions. This dataset has also been used in ActivityNet challenge since 2017.

Figure 2.12. Example action classes from the Kinetics dataset [12]. Left to right: abseiling, mowing lawn, biking through snow and kayaking.

3. DESCRIPTORS FOR SHAPE AND TRAJECTORY

In this chapter, we review the background of the methods which were used to train and evaluate our system. We first explore the spatio-temporal descriptor extraction techniques which were used in feature extraction step. Secondly, we review the PCA which was used to reduce the dimensionality of the extracted features. Then, we briefly explain GMMs and Fisher Vectors which were used for encoding the video samples. Finally, we describe the Support Vector Machines (SVMs) which were used to perform sign language classification using the Fisher Vectors that were extracted from each video.

3.1. Spatio-temporal Descriptors

3.1.1. Histogram of Oriented Gradients

Detecting and recognizing humans or objects in still frames and videos is a challenging problems due to pose changes, lightning conditions and cluttered background. Considering these problems, Histogram of Oriented Gradients (HOG) descriptors were first proposed by Dalal *et al.* [41] for detecting and recognizing humans in images. As these descriptors are seen very similar to edge orientation histograms and SIFT descriptors, they are extracted from a dense grid of cells and locally normalized blocks. In this approach, appearance and shape of humans or objects in the frame are described as a distribution of intensity gradients.

To compute HOG descriptors from a still frame, vertical (g_x) and horizontal (g_y) image gradient values are first computed by applying 1D centered derivative masks over the image. This computation often uses the following derivative masks;

$$[-1, 0, 1] \quad \text{and} \quad [-1, 0, 1]^T$$

After computing g_x and g_y values, the frame is divided into small spatial fragments called *cells*. 1D orientation histograms are then computed for each of these cells using g_x and g_y. To compute these 1D orientation histograms, calculating the magnitude and the orientation (direction) for each pixel is needed. The following formulas can be used to calculate the magnitude and the orientation using the image gradients;

$$g = \sqrt{g_x^2 + g_y^2}$$

$$\theta = \arctan \frac{g_y}{g_x}$$

where g represents the magnitude of the changes in gradient and θ represents the orientation angle of the gradient.

The process is completed by measuring the energy of the histograms in the regions called *blocks* which contain the cells, followed by the normalization of these blocks. Finally, normalized block of histograms vectorized into a one dimensional feature vector to be used in classification.

HOG descriptors represent the local appearance and structure information because they use the gradient changes in spatial regions. Therefore, they are often used in sign language recognition because they can be extracted from the hands and the face of the signers which have valuable discriminative information for recognition.

3.1.2. Histogram of Optical Flow

Histogram of Optical Flow (HOF) descriptors were first used in the work of Laptev *et al.* [66] for recognizing human action in videos. Although these descriptors are very similar to HOG descriptors in terms of their computation, they are calculated using optical flow which are obtained from consecutive frames of a video.

To compute HOF descriptors, same methodology for computing the HOG descriptors is followed except the gradient computation step. In this method, magnitude and orientation informations are computed from optical flow channels instead of intensity gradients. These optical flow channels represent vertical and horizontal local motion changes in pixels of consecutive frames. Using optical flow channels enables these descriptors to characterize local motion changes as a distribution of gradients of optical flow.

Due to their ability to represent the local motion between consecutive frames, these descriptors are often used in human action recognition. They can also be used in sign language recognition because they are able to capture the smaller motion changes of the signer in video. These smaller motion changes can be used to recognize movements such as finger spelling or facial expressions performed by the signer.

3.1.3. Motion Boundary Histograms

Motion Boundary Histograms (MBH) were proposed by Dalal v.d. [67] for detecting and recognizing movements of humans in videos. These descriptors were developed as an improvement to HOF descriptors and they are more robust to problems such as camera movement, background clutter and changes in lightning.

To compute MBH descriptors, optical flow and HOG methods are used together. After calculating optical flow from two consecutive frames in a video, each channel of the optical flow is assumed as grayscale image channel. These grayscale channel images are then used to extract HOG descriptors.

MBH descriptors are often used in human action recognition because they can cancel the background movement which occurs as a result of optical flow, and can focus on moving objects or humans in the frame. This may be important for human action recognition where videos contain too much background movement which dominates the person who is performing the action.

3.2. Improved Dense Trajectories

Although deep learning techniques have commonly used in large scale video recognition recently, local feature based trajectory extraction techniques are still popular for efficient human action recognition, and are used in comparison with popular deep learning approaches. As one of these methods Improved Dense Trajectories (IDT), proposed by Wang *et al.* [13, 14], is a state-of-the-art hand crafted trajectory extraction method which is based on dense optical flow.

3.2.1. Dense Trajectories

In the first work of Wang *et al.* [13], Dense Trajectories (DT) were proposed as an improvement of KLT tracker and SIFT descriptor matching which were successful approaches in the earlier work on action recognition domain. Their intention was to increase the quality and the quantity of the trajectories by sampling dense tracking points from dense optical flow fields extracted using subsequent frames.

To perform dense sampling, dense optical flow is sampled at 8 different spatial scales with an interval of $1/\sqrt{2}$. Then, median filtering to the dense optical flow field $f = (u_t, v_t)$ is performed for each sampled point $P_t = (x_t, y_t)$ for frame t to consecutive frame $t + 1$;

$$P_{t+1} = (x_{t+1}, y_{t+1}) = (x_t, y_t) + (k_M * f)|_{(\bar{x}_t, \bar{y}_t)} \tag{3.1}$$

where the rounded position of (x_t, y_t) is $(\bar{x}_t, \bar{y}_t)$ and the median filtering kernel is k_M. After median filtering is completed, tracked points are concatenated to form the trajectory $T_L = (P_t, P_{t+1}, \ldots, P_{t+L})$, where T is the trajectory and L is the length of trajectory. Once a trajectory reaches the length of L, it is removed from the tracking process. In addition to trajectory length, dynamic information of trajectories are also examined during the tracking process. Therefore, static trajectories and trajectories with large displacements are seen as noise and are removed from the tracking process.

After the tracking process, the trajectory shape descriptor is computed for each trajectory by encoding displacement vectors between tracked points of two subsequent frames. For the trajectory with length of L, it's shape is described by a sequence $S = (\Delta P_t, \ldots \Delta P_{t+L-1})$ where $\Delta P_t = (P_{t+1} - P_t)$ is displacement vector of tracked point P between time t and $t+1$. After defining the displacement sequences, trajectory shape descriptors are calculated by normalizing them using the equation as follows;

$$S' = \frac{(\Delta P_t, \ldots, \Delta P_{t+L-1})}{\sum_{j=t}^{t+L-1} \|\Delta P_j\|} \tag{3.2}$$

Beside the trajectory shape descriptor, local appearance and motion descriptors are also computed from tracked trajectories. These descriptors are computed within a space-time volume with the size of $N \times N$ pixels and L frames. Then, this space-time volume is divided into small spatio-temporal grids with the size of $n_\sigma \times n_\sigma \times n_\tau$ where n_σ is the size of spatial dimension and n_τ is the size of temporal dimension. Visualization of the tracking and the descriptor extraction can be seen in Figure 3.1.

HOG, HOF and MBH feature descriptors are computed for each spatio-temporal grid to represent appearance and motion. While the HOG descriptors are extracted to describe static appearance information, HOF descriptors are extracted to capture local motion information. Because of their ability to capture absolute motion, optical flow automatically captures the camera motion which can be seen as noise when recognizing human actions. Therefore, MBH descriptors were also used with HOG and HOF descriptors to suppress camera motion in action videos. While HOG and MBH descriptors are quantized into 8-bin histograms, HOF descriptors are quantized into 9-bin histograms because they need an additional non-motion bin. In addition, these features are also normalized by their l_2 norm.

3.2.2. Improving the Dense Trajectories

In their later work, Wang *et al.* [14] improved the dense trajectory approach by taking into account unnecessary camera motion in frames. One of the biggest

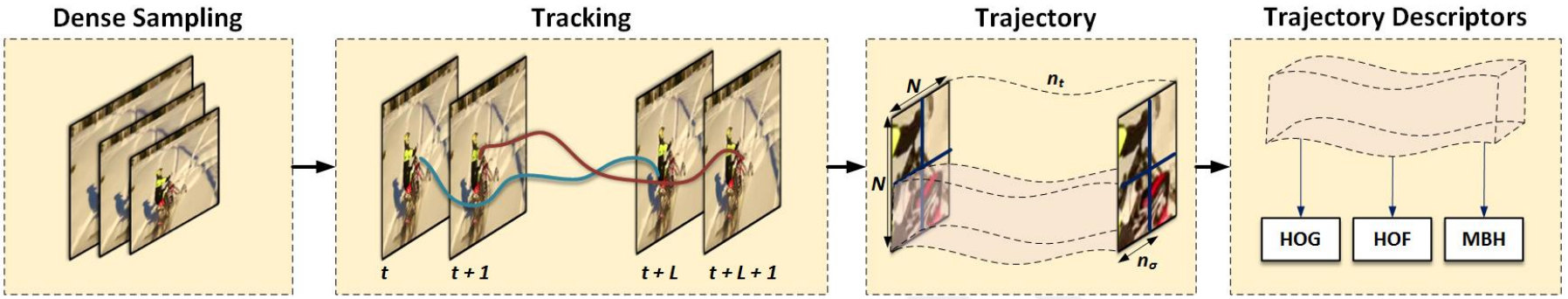

Figure 3.1. Illustration of the pipeline for extracting Dense Trajectories. Original illustration can be found in the work of Wang *et al.* [13].

problems authors have encountered in the action recognition using Dense Trajectories is unnecessary trajectories which describe the movement of the camera. Visualization of IDT on RGB videos can be found in Figure 3.2.

Figure 3.2. Visualization of Improved Dense Trajectory features on action and sign videos.

To overcome this problem, Wang *et al.* proposed Improved Dense Trajectories approach which can remove trajectories related to camera motion by taking into account camera movement. In order to remove unnecessary trajectories, they first estimated the camera motion. To estimate camera motion, they assumed that two consecutive frames are related by homography. Then, they have computed the homography between two frames by using Speeded Up Robust Features (SURF) [91] and Random Sample Consensus (RANSAC) [92]. After calculating the homography between two frames, optical flow field of the second frame is warped. Then, trajectories related to camera motion are removed if the displacement vectors in the warped optical flow field is under a threshold.

However, after this process, author have also seen that the camera motion estimation is not effective when humans dominate the frame. Therefore, they applied a state-of-the-art human detector just before the homography estimation, and removed

the bounding box regions where humans are present. Then, they estimated an another homography from obtained frames, and removed trajectories that are related to homography as they are considered as camera motion.

3.3. Dimensionality Reduction

In a machine learning algorithm, normalization or dimensionality reduction is needed in most cases where every dimension of the input for the algorithm may be unnecessary. Therefore projecting the data to a lower dimensional space may enable us to investigate the algorithm or the data in a much better way in terms of memory, computational complexity and robustness.

In this work, as proposed in [14], Principal Component Analysis (PCA) is used to reduce dimensionality of extracted spatio-temporal and trajectory features. The idea behind the PCA is projecting the d dimensional original data to a k dimensional space where $k < d$, without losing too much information.

We apply PCA to each descriptor of extracted features HOG, HOF and MBH separately. Instead of reducing the dimensions of these descriptors by half as proposed in [14], we reduce dimensions of our data by preserving %99 variance.

3.4. Clustering and Feature Encoding

Although local spatio-temporal descriptors that extracted from videos are very effective for representing appearance and motion, they are insufficient to be used in classification because their size may be different for each sample in the dataset. In order to use these features in a classifier, local feature encoding methods are used in the literature.

Most of the feature encoding methods are based on the idea of assigning local descriptors to visual vocabularies which are the codebooks learned by clustering using

a large set of descriptors. After assigning to a proper codebook, extracted descriptors are then encoded as a compact feature vector with a fixed size.

Most popular feature encoding methods used in the literature are Bag of Features (BoF) [93], Vectors of Locally Aggregated Features (VLAD) [94] and Fisher Vectors (FV) [68]. In this thesis we use Fisher Vectors to encode the descriptors for each video.

3.4.1. Gaussian Mixture Models

To encode Fisher Vectors for each video in the dataset, we use Gaussian Mixture Models (GMMs) [95] for clustering as proposed in [14, 68]. Gaussian Mixture Models are generative probabilistic models which can fit multiple Gaussian distributions on data samples. Therefore, GMMs are very practical when they are used to cluster data with variable number of samples.

When used in computer vision, Gaussian Mixture Models can be used as a probabilistic clustering method which forms universal visual vocabularies from the local features extracted from the data. Generated GMMs are then used to attain probabilities for each feature sample to visual vocabularies (clusters).

To generate a multivariate Gaussian Mixture Model for d-dimensional input $\vec{x}$, three parameters need to be considered; mixture component weights ϕ_k, mean vector $\vec{\mu_k}$ and covariance matrix Σ_k. For a D-dimensional sample set, a multivariate GMM can be defined as follows;

$$p(\vec{x}) = \sum_{k=1}^{K} \phi_k \mathcal{N}(\vec{x} \mid \vec{\mu_k}, \Sigma_k) \tag{3.3}$$

with the requirement of $\sum_{k=1}^{K} \phi_k = 1$ so that the probability distribution normalizes to 1. K is the total number of Gaussian components and $\mathcal{N}(\vec{x} \mid \vec{\mu_k}, \Sigma_k)$ is one of K

normal distribution;

$$\mathcal{N}(\vec{x} \mid \vec{\mu_k}, \Sigma_k) = \frac{1}{\sqrt{(2\pi)^D |\Sigma_k|}} \exp\left(-\frac{1}{2}(\vec{x} - \vec{\mu_k})^\top \Sigma_k^{-1} (\vec{x} - \vec{\mu_k})\right) \tag{3.4}$$

Constrains on the covariance matrices are often decided by the total amount of data to be used for learning the GMM parameters. The covariance matrices Σ_k can be diagonal or full rank matrices. In addition, all parameters used in estimating the model can be shared by all of Gaussian components.

To estimate Gaussian Mixture Models, Expectation Maximization (EM) technique is often preferred if the total number of mixtures K to be estimated is known. EM technique consists of two alternating steps expectation (E) and maximization (M), iterating until the algorithm converges.

The expectation step calculates the expectation of the each component's assignment for each point in the data given the parameters μ_k, Σ_k and ϕ_k.

$$\hat{\gamma}_{ik} = \frac{\phi_k \ \mathcal{N}(\vec{x_i} \mid \vec{\mu_k}, \Sigma_k)}{\sum_{j=1}^{K} \phi_j \ \mathcal{N}(\vec{x_i} \mid \vec{\mu_j}, \Sigma_j)} \tag{3.5}$$

where the posterior probability of component k for $\vec{x_i}$ is $\hat{\gamma}_{ik}$. The maximization step maximizes the log-likelihood which is calculated in expectation step, and updates the parameters $\mu_k^{'}$, $\Sigma_k^{'}$ and $\phi_k^{'}$;

$$\mu_k^{'} = \frac{\sum_{n=1}^{N} \hat{\gamma}_{ik} \vec{x_i}}{\sum_{n=1}^{N} \hat{\gamma}_{ik}} \tag{3.6}$$

$$\Sigma_k^{'} = \frac{\sum_{n=1}^{N} \hat{\gamma}_{ik} (\vec{x_i} - \mu_k^{'})(\vec{x_i} - \mu_k^{'})^\top}{\sum_{n=1}^{N} \hat{\gamma}_{ik}} \tag{3.7}$$

$$\phi_k^{'} = \frac{\sum_{n=1}^{N} \hat{\gamma}_{ik}}{N} \tag{3.8}$$

where N is number of samples in the data.

Since the EM algorithm tries to find the local optimum given data, initialization of the model parameters to reasonable values is important. To initialize the model parameters, k-means algorithm is often used for determining cluster locations (means) and the data points allocated to the clusters. Then, within-cluster covariances are computed for each cluster to complete the initialization process.

3.4.2. Fisher Vector Encoding

The Fisher Vector (FV) is a local feature pooling technique which was proposed by Perronnin *et al.* [96] as an advancement to the popular Bag of Features (BoF) pooling technique. As a special and improved case of the Fisher Kernel, Fisher Vector representation is compact and dense unlike BoF representation which is less desirable for image classification recently, due to their sparsity and high dimensionality.

Fisher Vectors are designed to extract encodings from local image features and pool them as a global image descriptor which can be used for tasks such as learning and comparison. Bag of Features (BoF) technique commonly uses the k-means clustering to 0-th order statistics information which represents the frequency of the words allocated to the codebooks. Differently from BoF, FVs use the results of estimated GMM. Thus, they are able to encode first and second order statistics such as mean and covariance.

The encode the local image features, the FV encoding technique computes the gradient vector of the mean and the covariance of each component of estimated GMM. Afterwards, these gradient vector are used as the difference between the distribution of local image features and the mixture densities.

To extract a Fisher vector from a set of D-dimensional local image descriptor vectors $X = \{x_1, x_2, \ldots, x_N\}$ with the size of N, learned parameters of a GMM with K components are used; $\Phi = \{\phi_k, \mu_k, \Sigma_k\}$, where the covariances matrices are assumed to be diagonal $\Sigma_k = diag(\sigma_k^2)$.

For the estimated GMM density function p given parameters Φ and local image descriptors X, the gradient vector of log-likelihood can be written as;

$$G_{\Phi}^{X} = \nabla_{\Phi} \log p(X|\Phi) \tag{3.9}$$

Under the independence assumption, where the D-dimensional samples of X are assumed to be independent, the gradient of log-likelihood;

$$\mathcal{L}(X|\Phi) = \sum_{n=1}^{N} \log p(x_n|\Phi) \tag{3.10}$$

where the likelihood of x_n is generated by the GMM;

$$p(x_n|\Phi) = \sum_{k=1}^{K} \phi_k \; p_k(x_n|\Phi_k) \tag{3.11}$$

with the constrain $\sum_{k=1}^{K} \phi_k = 1$, where the probability p_k is given by;

$$p_k(x|\Phi) = \frac{1}{\sqrt{(2\pi)^D |\Sigma_k|}} \exp\left(-\frac{1}{2}(\vec{x} - \vec{\mu_k})^{\top} \Sigma_k^{-1} (\vec{x} - \vec{\mu_k})\right) \tag{3.12}$$

Then, the gradients of X are calculated with respect to the GMM parameters Φ, and are normalized. Normalized gradient with respect to weights, means and covariances are written as following;

$$G_{\phi_k}^{X} = \frac{1}{N\sqrt{\phi_k}} \sum_{n=1}^{N} (\gamma_n(k) - \phi_k) \tag{3.13}$$

$$G_{\mu_k}^X = \frac{1}{N\sqrt{\phi_k}} \sum_{n=1}^{N} \gamma_n(k) \left(\frac{x_n - \mu_k}{\sigma_k} \right) \tag{3.14}$$

$$G_{\sigma_k}^X = \frac{1}{N\sqrt{\phi_k}} \sum_{n=1}^{N} \frac{\gamma_n(k)}{\sqrt{2}} \left[\frac{(x_n - \mu_k)^2}{\sigma_k^2} - 1 \right] \tag{3.15}$$

where γ_k is the posterior probability and the soft assignment of x_n to Gaussian k as defined in Equation 3.5.

A Fisher Vector for the sample X is then computed by concatenating the calculated gradients $F_X = \{G_{\phi_k}^X, G_{\mu_k}^X, G_{\sigma_k}^X : k = 1, \ldots, K\}$. After the concatenation step, a $(2D+1)K$ dimensional FV is obtained. In addition, it should be noted that the earlier work on Fisher Vectors used the gradients with respect to ϕ_k. However, in a later study, they were removed from the extraction process to reduce dimensions of the FV vectors because they bring limited amount of information. As a result, the final form of the Fisher Vectors have $2DK$ dimensions where D is dimensions of local features and K is number of components in the mixture model.

Once the FVs are extracted, normalization is applied to the FVs. In [97], Perronnin *et al.* proposed l_2 normalization followed by power normalization to improve the performance of the Fisher vectors for linear classifiers.

3.5. Support Vector Machines

In order to evaluate the performance of features that extracted for sign language recognition, we can train a machine learning algorithm. To perform the classification for recognizing sign videos, we use Support Vector Machines (SVMs) approach which is commonly used for classification problems where videos and their labels are used in training together.

Support Vector Machines (SVMs) were proposed by Cortes and Vapnik [98] to be used in classification and regression problems. The SVM is a discriminant-based method which adopts the Vapnik's principle to never solve a more complex problem as a first step before the actual problem [99].

Main idea behind the SVMs is to learn discriminant functions without estimating the class densities or the posterior probabilities. When training SVMs, the algorithm aims to project training samples into higher dimensional space, so that it can find optimal separating hyperplane that maximizes the margin between two classes. SVMs also support non-linear classification with kernel functions where various basis function can be used as kernels to represent the training sample in a different feature space.

Considering a simpler two-class classification problem with a linearly separable data, the optimal hyperplane can be written as;

$$\begin{aligned} w^\top x_i + b \geq +1, \quad \text{for} \quad y_i = +1 \\ w^\top x_i + b \leq -1, \quad \text{for} \quad y_i = -1 \end{aligned} \tag{3.16}$$

where $X = \{x_i, y_i : i = 1, \ldots, N\}$ is the training features x_i and their labels y_i which corresponds to -1 and $+1$ for a two-class problem. To find the optimal separating hyperplane we want to find parameters w and b such that the inequalities justify. Thus, we rewrite these inequalities in the form of;

$$y_i(w^\top x_i + b) \geq +1 \tag{3.17}$$

To define the optimal separating hyperplane, the distance between the hyperplane and the closest instance is needed to be maximized. This distance is also called as the margin which can be defined as following;

$$\frac{y_i(w^\top x_i + b)}{\|w\|} \tag{3.18}$$

To maximize the margin, $\|w\|$ is needed to be minimized. Therefore, the task transform into a standard quadratic optimization problem which can be defined as following;

$$\begin{aligned} &\min_{w,b} \frac{1}{2}\|w\|^2 \\ &\quad \text{s.t.} \quad y_i(w^\top x_i + b) \geq +1, i = 1, \ldots, N \end{aligned} \tag{3.19}$$

Solution of this optimization problem gives the optimal margin. However, this optimization problem is high depends on dimensionality d of the data. Thus, using Lagrange multipliers a, we form a new formulation which changes the problem to a convex optimization problem that depends on the number instances N instead of dimensionality. Therefore, the formulation becomes;

$$\mathcal{F}(w, b, a) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{N} a_i[y_i(w^\top x_i + b) - 1] \tag{3.20}$$

Then, we find the dual of the problem by minimizing $\mathcal{F}(w, b, a)$ w.r.t w and b, and with the constrain $a_i \geq 0$. In order to find dual, we take the derivatives of $\mathcal{F}$ w.r.t to w and b to zero.

$$\nabla_w \mathcal{F}(w, b, a) = w - \sum_{i=1}^{N} a_i y_i x_i = 0 \tag{3.21}$$

$$\nabla_b \mathcal{F}(w, b, a) = \sum_{i=1}^{N} a_i y_i = 0 \tag{3.22}$$

When we plug these derivatives to the formulation in Equation 3.20, we obtain;

$$\mathcal{F}(w, b, a) = \sum_{i=1}^{N} a_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} y_i y_j a_i a_j (x_i)^\top x_j - b\sum_{i=1}^{N} a_i y_i \tag{3.23}$$

where the last term of the equation is $\sum_{i=1}^{N} a_i y_i = 0$. Finally, we put everything together with constrains and obtain the following optimization problem which can be solve using quadratic optimization that depends on the number of instances N in the

data;

$$\begin{aligned} \max_{a} W(a) &= \sum_{i=1}^{N} a_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} y_i y_j a_i a_j (x_i)^\top x_j \\ \text{s.t. } & a_i \geq 0, \quad i = 1, \ldots, N \\ & \sum_{i=1}^{N} a_i y_i = 0 \end{aligned} \tag{3.24}$$

Once we solve a_i, we can find the support vectors which correspond to the set of x_i whose $a_i \geq 0$.

If training samples are not linearly separable, the algorithm derived for the separable case will not work. In this case, we try to find a hyperplane which has the minimum error. To find such hyperplanes, optimization can be reformulated as;

$$\begin{aligned} \min_{w,b} \ & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N} \xi_i \\ & \text{s.t.} \quad y_i(w^\top x_i + b) \geq 1 - \xi_i, \ i = 1, \ldots, N \\ & \xi_i \geq 0, \ i = 1, \ldots, N \end{aligned} \tag{3.25}$$

where ξ_i can be defined as a variable which is used calculate the soft margin. C is the regularization parameter which is measured by taking l_2 norm of the relative weight vector.

After adding new constrains to the formulation in Equation 3.20, we have;

$$\mathcal{F}(w, b, \xi, a, t) = \frac{1}{2}w^\top w + C\sum_{i=1}^{N} \xi_i - \sum_{i=1}^{N} a_i[y_i(x^\top w + b) - 1 + \xi_i] - \sum_{i=1}^{N} t_i \xi_i \tag{3.26}$$

where a and r the Lagrange constrains which are both greater than 0. When we use the regularization parameters and constrains, we will have the dual form of the problem

as following;

$$\begin{aligned} \max_{a} W(a) &= \sum_{i=1}^{N} a_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} y_i y_j a_i a_j (x_i)^\top x_j \\ \text{s.t. } 0 &\leq a_i \leq C, \quad i = 1, \ldots, N \\ \sum_{i=1}^{N} a_i y_i &= 0 \end{aligned} \tag{3.27}$$

To use this optimization in a non-linear problem, we can change the dot product $x^\top x$ of dual form to a mapping function $\phi(x)$. This feature mapping allows us to map the problem to a non-linear feature space. Given a feature mapping, kernel function can be defined as;

$$K(x, v) = \phi(x)^\top \phi(v) \tag{3.28}$$

In the literature, linear and radial basis functions (RBF) are commonly used when training SVMs.

$$\begin{aligned} K_{linear}(x, v) &= x^\top v \\ K_{rbf}(x, v) &= \exp\left(-\frac{\|x - v\|^2}{\sigma^2}\right) \end{aligned} \tag{3.29}$$

In this work, we used linear SVMs to classify Fisher Vectors extracted from the sign videos.

4. PROPOSED SIGN LANGUAGE RECOGNITION METHODOLOGY

In this work, we try to adapt the Improved Dense Trajectory (IDT) approach into the sign language recognition domain. As one of the most successful methods on large-scale human action recognition, IDTs showed their value by recognizing human actions using densely extracted various spatio-temporal descriptors such as HOG, HOF and MBH. Due to their ability to capture the movements of humans by tracking the dense optical flow field extracted from videos, IDTs can be used for the domain of SLR where they capture and describe the movement of the hands of the signer.

Although IDTs have high accuracy on various action recognition datasets, they extract highly complex features based on tracking the optical flow. Therefore, they cannot be used in a real-time recognition system because they slow down feature extraction, training and especially prediction steps.

When recognizing the human actions in videos, the algorithm should try to find discriminative information by looking at the entire frame which may contain valuable information about the action. Most of the videos in action recognition datasets include actions with full body movement, background movement and human-object interaction. Therefore, recognizing such action requires computationally complex features. However, SLR problem do not require most of these actions and scene related characteristics since most of the SLR datasets contain samples with signers performing the sign in front of a static background. For this reason, reducing the complexity of the features while preserving the accuracy of the system is important.

In this thesis, we aim to reduce the complexity of the SLR system by modifying the IDT and extracting spatial-temporal descriptors separately. In our work, we first apply baseline approach by extracting IDT features and then extracting FVs to classify them with SVMs [100]. Secondly, we try to reduce the number of trajectories by filtering

the trajectories belong the region around the hands. Finally, we extract HOG, HOF and MBH descriptors from sign videos where we use crops around hands instead of the entire frame [101].

In our SLR system, we used a traditional computer vision pipeline which consist of preprocessing, feature extraction, clustering and classification. Illustration of our pipeline can be seen in Figure 4.1. According to our pipeline, we first apply PCA dimensionality reduction on HOG, HOF and MBH descriptors (also trajectory shape descriptors for IDT) separately after the feature extraction step. Even though all of these descriptors are histogram based features, they can be characterized by different sets of information such as appearance and motion. After the dimensionality reduction, we generate Gaussian Mixture Models for each descriptor type to encode Fisher Vectors for each video of the dataset. As the last step of our pipeline (see Figure 4.1), we train and test SVMs to find which combination of FVs gives the best performance for sign language recognition.

4.1. Feature Extraction

For training the proposed SLR system, we extracted two types of features; Improved Dense Trajectories (IDT) [14] and spatio-temporal descriptors that we called as Hand Descriptors. In addition to these features, we used skeleton information obtained via Microsoft Kinect v2 to find hand coordinates.

4.1.1. Filtered Improved Dense Trajectories

To extract IDT features, we used the publicly available implementation of Wang *et al.* [14]. This implementation accepts inputs as video files and returns the extracted feature as text file for that video. The output text file consists of trajectory shape, HOG, HOF and MBH descriptors for each tracked trajectory within a video. Visualization of the trajectories on a sign video is illustrated in Figure 4.2.

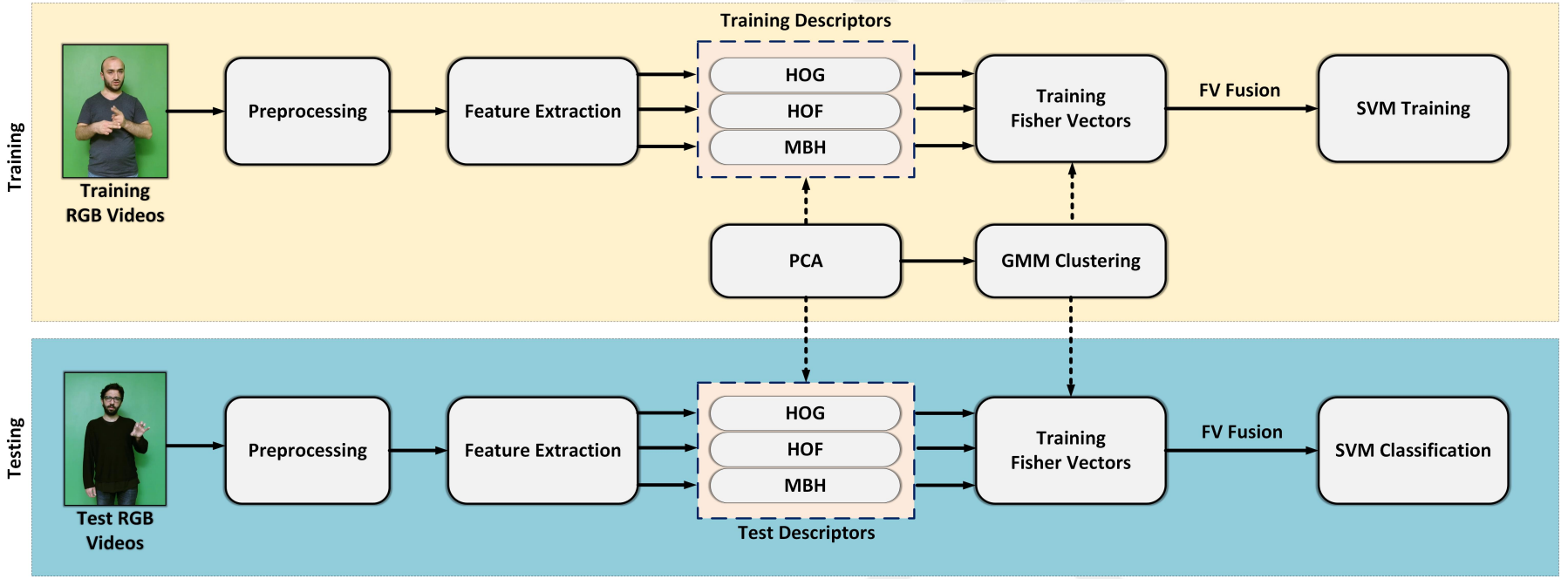

Figure 4.1. Pipeline used in this thesis for sign language classification.

For each trajectory, 96 dimensional HOG, 108 dimensional HOF and 192 dimensional MBH (96 dimensions for both optical flow channels) descriptors are extracted. While the dimensions of these descriptors unaffected by parameter changes, the dimension of trajectory shape descriptors are affected by the trajectory length parameter L. Trajectory shape descriptors are extracted using the displacement vectors of the tracked trajectory. Thus, they have $2 \times L$ dimensions.

Figure 4.2. Visualization of Improved Dense Trajectory features on a sign video.

Moreover, to see the effects of using fewer trajectories by focusing on the regions around the hands, we filtered the trajectories around the hands of the signer by using the hand pixel coordinates which were provided by Microsoft Kinect v2. To filter trajectories, we matched the hand pixel coordinates with tracked trajectory coordinates which were obtained from the extract IDT features. For each trajectory, tracking coordinates are defined as (x, y) positions of the trajectory along L frames.

4.1.2. Hand Descriptors

In order to speed up the feature extraction process, we propose to extract HOG, HOF and MBH descriptors individually from the crops that we obtained around the hands. Before extracting any descriptor, videos were first preprocessed. During preprocessing, video frames were scaled to a smaller size in order to accelerate the feature extraction. Trade-off between the accuracy and the speed is discussed in the experi-

ments chapter. After scaling, crops of hand regions were taken from scaled frames of the sign videos using the hand pixel coordinates.

For each video with N frames, HOG descriptors were extracted with parameters of cell size n_{cell}, block size n_{block} and number of bins n_{bins}. To obtain HOF and MBH descriptors, we first extracted optical flow from each two consecutive frames using Farnebäck's optical flow approach which is based on polynomial expansion [102]. However, calculating optical flow between crops of two consecutive frames is impractical because the movement of the hands cannot be observed clearly. So, we first calculate optical flow of the whole frame, and then we crop the optical flow results around the hand region.

From the optical flow crops, we computed HOF descriptors with parameters n_{cell}, n_{block} and $n_{bins} + 1$ (an extra bin for no movement). For the extraction of MBHs, horizontal (x) and vertical (y) components of the optical flow were used. To extract descriptors, x and y components were considered as grayscale images and then used to extract HOG descriptors with same parameters n_{cell}, n_{block} and n_{bins}. Finally, these descriptors were combined to obtain MBH descriptors (MBH_x and MBH_y). The visualized hand crops and extracted spatio-temporal descriptors are as in Figure 4.3.

4.2. Feature Normalization and Encoding

Since the dimensions of descriptors are too large, Principal Component Analysis (PCA) was applied separately for each descriptor and the dimensions of the descriptors were reduced. Unlike Wang *et al.* [14]'s work in which the dimensions of the descriptors were reduced by half, we reduced the dimensions of the descriptors by preserving 99% of the variance of the data. It is more important for us to preserve the variance of the data because the number of descriptors obtained from a sign language video is less than that of activity recognition videos. On top of this, reducing the dimensions of these descriptors by half results in sparser representations. This leads to a poor performance

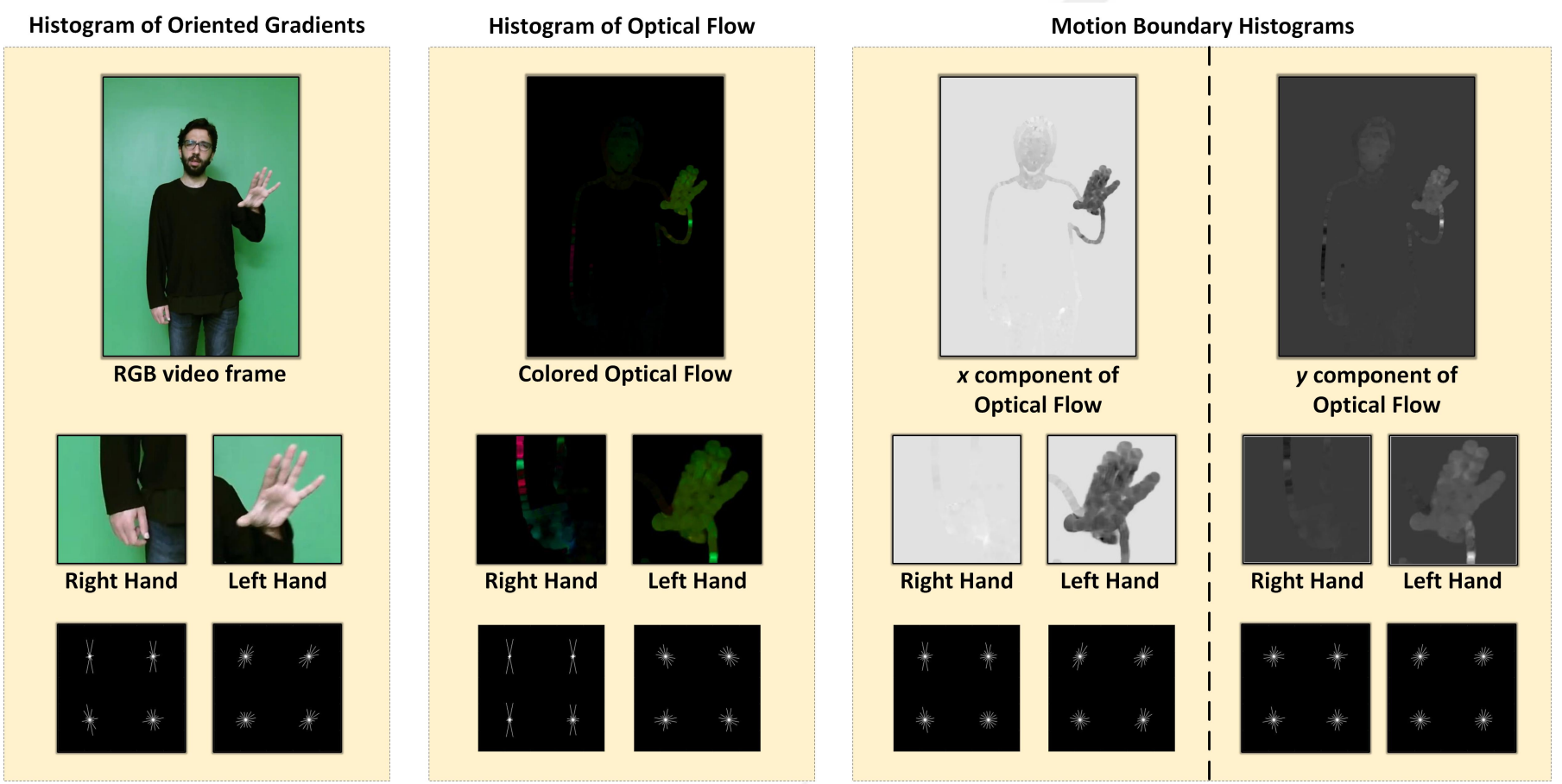

Figure 4.3. Illustration of feature extraction step of the Hand Descriptors.

in classification when using Fisher Vectors (FV) which often gives high performance with dense representations.

For the Fisher Vectors (FV), normalization is applied by default. As mentioned in Section 3.4.2, Perronnin *et al.* [97] proposed and improved version of the Fisher Vectors where they applied power normalization which is followed by l_2 normalization, and showed that normalizing the FVs can lead to improved results.

After PCA normalization, Gaussian Mixture Models (GMMs) with number of clusters k were estimated from the concatenated training descriptors to learn first and second statistics information (means and variances) about the training data. Next, $2Dk$ dimensional Fisher Vectors (FV) were extracted for each sign video in the dataset using means and variances of the GMM. D indicates the dimensions of the descriptors after PCA normalization.

4.3. Feature Classification

After encoding every sign language video in the dataset, we performed training using the FVs and their corresponding labels. Since Fisher Vectors (FV) are known to perform well in linear classifiers, we have trained linear Multi-class Support Vector Machines (SVMs) in our work.

To optimize the parameters, SVMs were trained using grid search cross-validation which is a method for finding best parameters for a classifier by performing exhaustive search over a set of parameters. In our work, we applied this technique to find the best regularization parameter C from the values $\{2^{-3}, 2^{-1}, 2^{1}, 2^{3}, 2^{5}\}$. Once we find the best classifier with parameter C, we used it to predict the test data. After predicting a set of labels given the test data, we matched them with true labels of test data and calculate an accuracy which represents the performance of our method.

5. EXPERIMENTS AND RESULTS

5.1. Datasets

In our experiments, we used the BosphorusSign dataset in order to evaluate our methodology on sign language recognition. BosphorusSign is a publicly available Turkish Sign Language (TSL) dataset which has been collected by Boğaziçi University's PILAB group. The dataset contains sign videos in three different groups; health, finance and commonly used words.

Since this dataset is recorded by using Microsoft Kinect v2, each recording in the dataset consists of RGB video (with resolution 1920×1080 at 30 fps), depth map, user mask and skeleton information. During recording, a single signer is present in front of the camera, performing TSL gestures. In the thesis, we used the Overlapping Subsequence Dataset (OSD) and the General Dataset (GD) which were derived from the BosphorusSign dataset.

5.1.1. Overlapping Subsequence Dataset

In the subset, there are 399 segmented sign videos of 10 similar signs performed by 6 different individuals (see Figure 5.1). These sign videos are divided into 305 training and 94 test videos to be used in the evaluation of the proposed method.

The subset contains similar signs for example *Urgent* and *Is it urgent?* where the sign can have multiple gloss meanings which means that some specific subsequences may be shared among the sign videos. Classes of the subset are listed with their Turkish translations in Table 5.1.1.

Figure 5.1. Six signers of the BosphorusSign Turkish Sign Language dataset.

In this work, we used this subset for finding the optimal parameters of our approach which are then used to evaluate a large-scale sign language recognition problem.

Table 5.1. Class list of the Overlapping Subsequence Dataset with translations.

Class Labels	Turkish Translations
Pain	Acı
Open	Açık
Open Auction	Açık Arttırma
Explain	Açıklamak
Urgent	Acil
Is it urgent?	Acil mi?
Emergency	Acil Servis
Name	Ad
Mensturation	Adet
Address	Adres

5.1.2. General Dataset

To see the performance of our features on a much larger data, we used the sign videos from the General category of the Bosphorus Dataset. In this dataset, there are $5,829$ segmented sign videos of 154 sign words which are performed by 6 different signers. Videos in this subset were selected from commonly used words such as *Dinner*, *Pocket*, or *Forget*. Classes of the subset are shown in Table 5.1.2.

In addition, test samples of these subsets consist of sign videos obtained from a single signer whose samples are not in the training sets. This protocol also called as *Leave One Signer Out*. Therefore, we have tested our system's performance on a test set which were not seen in the training process. This allows us to train signer-independent

Table 5.2. Class list of the General Dataset (154 classes).

A week ago	Dress up	Hide	Now	Stealing (Theft)
Address	Drink	Hide	Number	Stop
After	Drowning	Home address	One week later	Surname
Age	Early	Hour (Time)	Open	Swallow
Ask	Eat	How are you	Outside	Teaching
Bad	Email	ID	Pay	Telephone
Be	End	Immediately	Playing (Music)	Thank
Be grateful for	Escape	Inside	Pocket	Thinking
Be retired	Evening	Invite. call	Pour	Thirst
Beating	Exchange	Itch	Pull	To be fed up
Before	Exist	Jump	Question	To Bite
Being Quiet	Explain	Know	Receive	To bring
Boy	Fall	Know	Relative	To concentrate
Breakfast	Father	Late	Remember	To crawl
Breastfeeding	Feel	Learn	Repeat	To Die
Burn	Fight	Like	Request	To look like
Business Address	Find	Listen	Response	Touch
Call	Fold	Little	Run	Translate
Catch	Forget	Look	Say	Undress
Child	Freeze	Lose weight	School Address	Very
Children	Friend	Lunch	Search (Phone)	Vomit
Close	Gaining weight	Make	See	Wait
Cold	Get up	Male	Sell	Week
Collapse	Girl	Mark	Send	Win
Collide	Give	Mobile phone	Show	Woman
Come	Go	Mom	Signature	Work
Cover	God Bless	Moon	Sit	Write
Cut	God willing	Morning	Some	Year
Delay	Good	Name	Special	Yell
Dinner	Hearing	Noon	Stand up	You are welcome
Divorce	Heat	Not available	State	

recognition systems where the recognition performance is independent from the signer performing the sign.

5.2. Experiments with Improved Dense Trajectory Features

In our experiments, we first investigate the Improved Dense Trajectory (IDT) features by extracting and training them according to our pipeline. Since we propose to use these features on sign language problem, we want to see the applicability of IDT features to our problem, and establish a baseline for our problem. We also report preliminary results on the DEVISIGN dataset, providing a basis for our adaptation where we have used IDT features for signer-independent sign language classification.

To acquire a baseline result, we extract IDT features using different set of parameters; n_t temporal stride, L trajectory length and k number of clusters in GMM. Due to the high complexity of IDT features, we first scaled the videos from the resolution of 1920×1020 to 640×360. After finding a good set of parameters, we study effects of filtering trajectories by only looking trajectories around the hands.

5.2.1. Preliminary Results on Devisign Dataset

To provide an extra basis for using Improved Dense Trajectory (IDT) features on sign language classification, we have performed experiments on the DEVISIGN Chinese Sign Language dataset [100]. Since this dataset has a large vocabulary of 2000 signs with maximum of two repetitions, we have used a subset of 1200 sign videos of 200 sign classes. Results of the experiments are shown in Table 5.3.

In our experiments, we have extracted IDT features from each sign video using default parameters of trajectory length 15 and temporal stride 3. In the experiments on this dataset, we have randomly sampled trajectories and obtained PCA and GMM parameters. Then, we have extracted Fisher Vectors for each sign video. Since we have performed the random sampling, we have repeated our experiments five times

and calculate the mean accuracy and its standard deviation. In addition, we have performed leave-one-out cross validation for each user in order to investigate the signer-independence.

As can be seen in Table 5.3, using only HOG descriptors has reduced the prediction performance of our system while using the individual HOF and MBH or the combinations of these descriptors yields better results on the DEVISIGN dataset. Our experiments have also shown that one of the users (7th user) was performing the signs erroneously. Therefore, we have removed this user from the training procedure and then computed the average performance our system. After performing this experiment, we have found that excluding the faulty videos of this user has increased our system's performance by nearly 4% in average.

In the rest of this chapter, we present experimental results on the BosphorusSign dataset.

5.2.2. Temporal Stride

As the last step of extracting IDT features, temporal averaging is applied to each trajectory with a length of L. Temporal stride is defined as the temporal averaging factor for each trajectory. This factor can be calculated by dividing L with n_t, where n_t is the temporal cell size of the spatio-temporal descriptors. After calculating the temporal stride, it is used for averaging the each L/n_t trajectory.

To examine the temporal stride changes in our problem, we have extracted the IDT features with different temporal cell sizes $n_t = \{3, 6, 9\}$. Considering only the effect of n_t, we fixed the trajectory length L parameter to 10. Then, we generated GMMs with different number of clusters $k = \{8, 16, 32, 64\}$. Finally, we extracted Fisher Vectors from videos for each combinations of the descriptors HOG, HOF, MBH and trajectory shape. Results of the experiments that we have conducted can be seen in Table 5.4.

Table 5.3. Performance of Improved Dense Trajectories on the subset of DEVISIGN dataset using leave-one-out cross validation. (EU - Excluded User) (* denotes the best performing method)

Users	HOG	HOF	MBH	HOG+HOF	HOG+MBH	HOF+MBH	HOG+HOF+MBH
EU-1	46.55 ± 2.04	73.45 ± 0.74	78.15 ± 0.74	77.75 ± 0.25	77.85 ± 1.51	77.10 ± 0.76	77.30 ± 1.25
EU-2	46.55 ± 2.23	76.00 ± 1.02	81.95 ± 1.10	77.45 ± 0.48	71.65 ± 2.86	81.85 ± 0.42	78.25 ± 1.22
EU-3	60.35 ± 2.00	83.55 ± 0.38	88.10 ± 0.65	86.25 ± 0.94	85.20 ± 1.63	87.55 ± 0.33	87.45 ± 1.24
EU-4	31.85 ± 0.99	72.00 ± 0.88	75.85 ± 0.22	71.64 ± 1.23	61.70 ± 4.02	78.00 ± 1.10	74.80 ± 1.88
EU-5	64.00 ± 0.61	74.90 ± 1.64	81.90 ± 1.52	78.30 ± 0.76	78.40 ± 1.39	81.50 ± 1.54	81.50 ± 0.79
EU-6	62.70 ± 3.68	71.20 ± 2.17	77.80 ± 1.04	77.10 ± 0.65	74.90 ± 2.56	77.10 ± 1.56	77.50 ± 2.67
EU-7	31.10 ± 3.03	48.40 ± 1.78	51.40 ± 0.96	55.50 ± 1.41	46.30 ± 2.80	52.10 ± 3.54	52.40 ± 4.08
EU-8	55.70 ± 0.97	77.10 ± 0.42	78.60 ± 1.24	79.00 ± 2.18	77.90 ± 1.14	76.90 ± 1.08	76.40 ± 2.04
Average	49.85 ± 1.95	72.05 ± 1.13	$76.72 \pm 0.93^*$	75.38 ± 0.99	71.74 ± 2.24	$76.89 \pm 1.29^*$	75.70 ± 1.90
w/o User #7	52.53 ± 1.79	75.43 ± 1.03	$80.34 \pm 0.93^*$	78.21 ± 0.93	75.37 ± 2.16	$80.43 \pm 0.97^*$	79.03 ± 1.59

Table 5.4. Effects of using different temporal cell size (n_t) and cluster counts (k) on the performance of different IDT feature combinations on the OSD. (* denotes the best performing method)

Temporal Cell Size (n_t)	# Clusters (k)	Trajectory Shape	HOG	HOF	MBH	HOG+HOF	HOG+MBH	HOF+MBH	All w/o Traj	All
3	8	75.53	91.49	97.87	97.87	97.87	98.94*	98.94*	98.94*	97.87
	16	91.49	97.87	95.74	95.74	98.94*	98.94*	95.74	98.94	98.94*
	32	91.49	96.81	95.74	97.87	96.81	98.94*	96.81	97.87	98.94*
	64	95.74	98.94*	94.68	97.87	96.81	98.94*	94.68	97.87	98.94*
6	8	89.36	94.68	87.23	93.62	95.74	98.94*	96.81	98.94*	97.87
	16	91.49	96.81	88.30	92.55	94.68	98.94*	94.68	96.81	97.87
	32	91.49	96.81	88.30	95.74	96.81	98.94*	96.81	97.87	97.87
	64	94.68	96.81	95.74	96.81	95.74	98.94*	97.87	97.87	97.87
9	8	76.60	81.91	94.68	93.62	92.55	92.55	95.74	94.68	96.81
	16	79.79	82.98	97.87	93.62	96.81	92.55	97.87	97.87	96.81
	32	79.79	89.36	97.87	92.55	96.81	96.81	96.81	97.87	96.81
	64	84.04	85.11	97.87	94.68	96.81	96.81	96.81	96.81	95.74

In the results of the experiments, we have seen that increasing the temporal cell size of the descriptors has a bad effect on the accuracy of our system. As can be seen in Table 5.4, performance of the individual descriptors are slightly changed by the different temporal cell size except the trajectory shape descriptors which yield drastically less performances.

As the combinations of the descriptors have mostly yielded 98.94% accuracy on the test set, we can not observe an outcome for the changing cluster count. Considering the results we have obtained, we have chosen $n_t = 3$ as temporal cell size for the next experiment where we examined the effects of trajectory length.

5.2.3. Trajectory Length

Since the IDT features highly depend on trajectory attributes, the length of the trajectories are important. Therefore, changing the trajectory length may allow us to track long or short movements occurring in videos. To see the effects of trajectory length in our problem, we have performed experiments using trajectory length $L = \{10, 15, 20\}$, number of clusters $k = \{8, 16, 32, 64\}$ and the fixed parameter temporal cell size $n_t = 3$ which is considered as the best parameter of the previous experiment.

According to the results of the experiments (see Table 5.5), increasing the length of the trajectories improved the overall performance of our system. In our experiments, we have experienced that the performance of the descriptor combinations were improved while the performance of the individual descriptors were decreased slightly.

Thus, we can say that features extracted with trajectory length $L = 20$ and temporal cell size $n_t = 3$ parameters are suitable for sign language recognition. Although we were able to observe the effects of trajectory length and temporal stride in our problem, we were not able to observe the effects of estimating GMMs with different number of clusters k. Unlike the other parameters, increasing the number of clusters did not affect the performance of the combined descriptors.

Table 5.5. Effects of using different trajectory length (L), cluster count (k) and the fixed temporal cell size $n_t = 3$ on the performance of different IDT feature combinations on the OSD. (* denotes the best performing method)

Trajectory Length (L)	# Clusters (k)	Trajectory Shape	HOG	HOF	MBH	HOG+HOF	HOG+MBH	HOF+MBH	All w/o Traj	All
10	8	75.53	91.49	97.87	97.87	97.87	98.94*	98.94*	98.94*	97.87
	16	91.49	97.87	95.74	95.74	98.94*	98.94*	95.74	98.94*	98.94*
	32	91.49	96.81	95.74	97.87	96.81	98.94*	96.81	97.87	98.94*
	64	95.74	98.94*	94.68	97.87	96.81	98.94*	94.68	97.87	98.94*
15	8	87.23	87.23	93.62	97.87	95.74	96.81	96.81	95.74	97.87
	16	92.55	88.30	95.74	92.55	95.74	93.62	96.81	96.81	96.81
	32	91.49	94.68	95.74	97.87	97.87	96.81	96.81	97.87	97.87
	64	91.49	93.62	93.62	94.68	96.81	95.74	94.68	95.74	97.87
20	8	78.72	88.30	95.74	96.81	98.94*	98.94*	97.87	98.94*	98.94*
	16	87.23	92.55	96.81	96.81	98.94*	97.87	97.87	97.87	98.94*
	32	91.49	95.74	94.68	96.81	98.94*	98.94*	95.74	97.87	98.94*
	64	94.68	96.81	95.74	94.68	98.94*	97.87	96.81	97.87	98.94*

Moreover, we evaluate our system on the General Dataset using the set of parameters which have yielded the best results on the OSD. In the experiment, we estimated GMMs with the number of clusters $k = \{8, 16, 32, 64, 128\}$. In addition to previous experiments, we have also performed experiments using $k = 128$ with the assumption that using more clusters may preserve the density of the representations with the increasing number samples and classes.

As can be seen in Figure 5.2, the performance of all descriptors and their combinations have increased as the number of clusters increased. However, we have seen that the performance of the trajectory shape descriptors are low even with $k = 128$ (62.68%) when they were used with large number of samples unlike the previous experiments where we used the smaller dataset.

Similar to previous experiments, using the combinations of the descriptors has increased the testing accuracy of our system. On the other hand, using the combination of HOG, HOF and MBH has yielded the best results on the test set while the combination of the same descriptors has a lower accuracy when they were combined with the trajectory shape descriptors. Thus, we have observed that using trajectory shape descriptors individually not only performs poorly on large datasets but also reduces the performance of other descriptors when they are used together.

We have also observed that the accuracy of our system increased for each possible combination of the descriptors when we used more clusters for estimating GMMs. However, increasing the number of clusters started to affect the accuracy less at some point after $k = 64$. In Figure 5.2, the combination of HOG, HOF and MBH has yielded 87.74% and 87.84% for $k = 64$ and $k = 128$ respectively. Even though its accuracy is higher, we decided not to use the features extracted using the GMM with $k = 128$ since they are 2 times the size of the features extracted with $k = 64$.

We present the confusion matrix of the experiments in Figure 5.3. We can see that some of the classes were confused more than others. When we analysed these

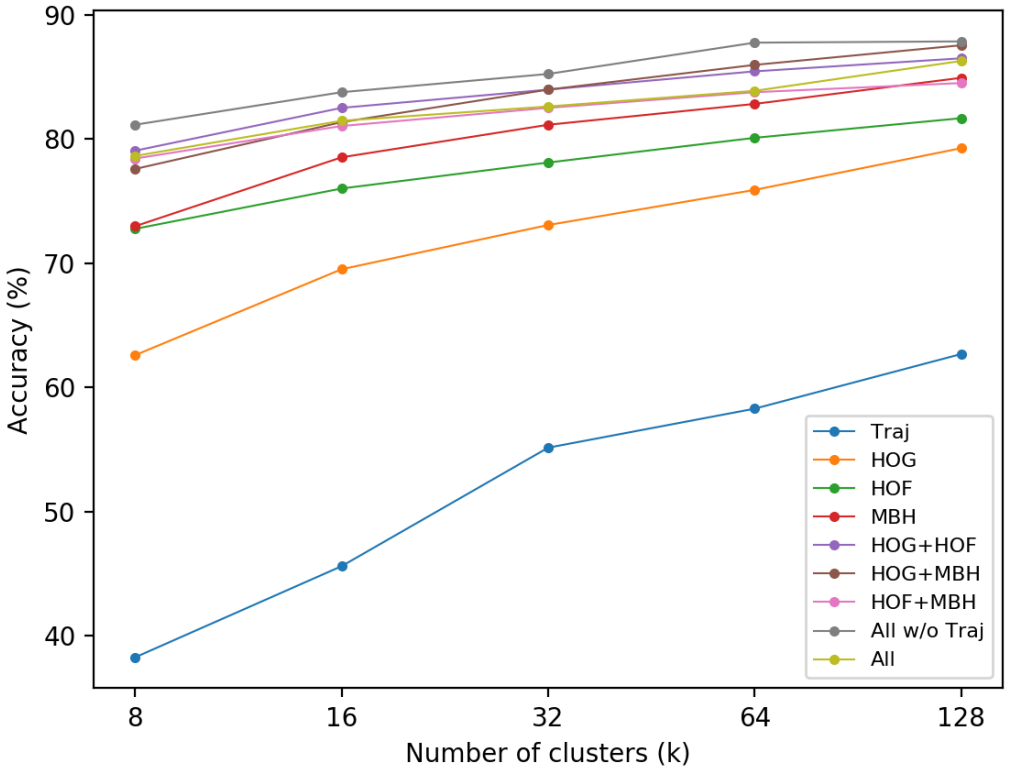

Figure 5.2. Performance of the IDT features on the General Dataset using the parameters $n_t = 3$ and $L = 20$

confusions, we have seen that *Know* sign is mostly confused with *Father*, *Woman* and *Exists* signs (see Figure 5.4). These signs have very similar arm and hand movements which can easily be confused with each other. As a result, we can say that IDT features could not separate these signs because they cannot characterize the hand shapes of the signers in more detail.

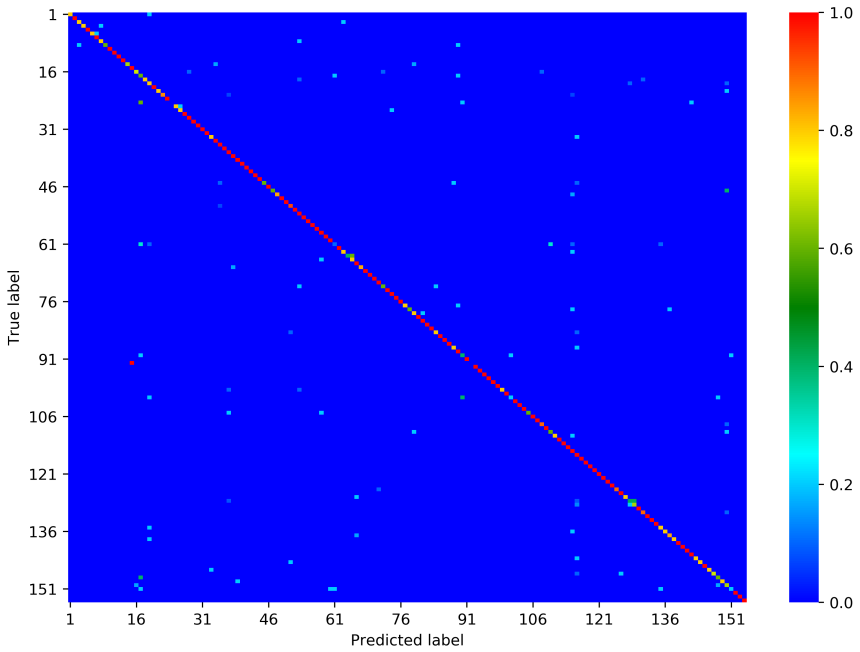

Figure 5.3. Confusion matrix of the experiment in which baseline Improved Dense Trajectories were used on the GD.

We have also found that some of the signs are visually the same but they were labelled differently. In the case of *Get Up* and *Stand Up* signs, we have seen that they are identical signs although their Turkish translations are different. In addition, we have seen that some classes such as *Question* and *Ask* signs have two different signs which were performed with different hand and arm movements. Therefore, using these signs in training makes it difficult for the classifier to separate the classes.

For the remaining experiments, we used the combination of HOG, HOF and MBH because they have achieved the best results on the GD. Furthermore, we used this combination with the parameters of $k = 64$, $L = 20$ and $n_t = 3$.

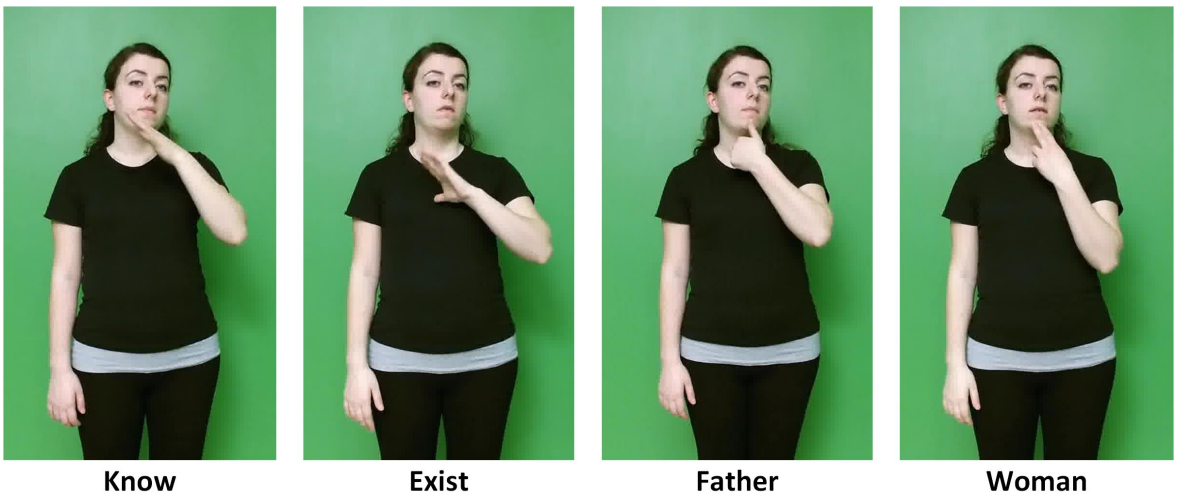

Figure 5.4. Visualization of confused signs. Ground truth: Know, Misclassified samples: Exist, Father and Woman.

5.2.4. Filtering Trajectories

After establishing a baseline model with IDT parameters, we have performed experiments on the General Dataset where we examined the effects of filtering the trajectories after feature extraction. To filter trajectories, we use the skeleton information for tracking the hand movements and match them with trajectory tracking coordinates. During the matching, each trajectory is tracked for L frames and removed if it is outside the boundaries of the hand region.

To determine a reasonable size for hand region, we have evaluated our system using hand regions with the bounding box size of 70×70, 80×80 and 90×90. In our experiment, we use the same experimental setup as the previous experiment; $n_t = 3$, $L = 20$, $k = 64$ and the combination of HOG, HOF and MBH which has yielded best result on the GD. In addition to the accuracy, we have also reported the number of total trajectories and the number of trajectories per sign which were lost during the process. All experiments in Table 5.6 were conducted on the GD.

In our experiment, we have seen that filtering trajectories around a small hand region such as 70×70 resulted in 83.02% accuracy on the test set which is lower than the baseline. However, we have experienced that the total number of trajectories used in the training step were drastically reduced from nearly 12 million trajectories to 1.5 million trajectories. Similar to the total number of trajectories, trajectories used per sign video have also reduced.

When we analysed the most confused signs for the case of small hand region, we have seen that *Get Up* and *Stand up* signs are still confused with each other as in the baseline IDT results. The confusion matrix of this experiment can be seen in Figure 5.5. After filtering procedure, our recognition system has started confusing the signs where the head and the mouth movements are present. For example, *Woman* and *Girl* signs have same arm and hand movements but they are different where the signer is mouthing the sign.

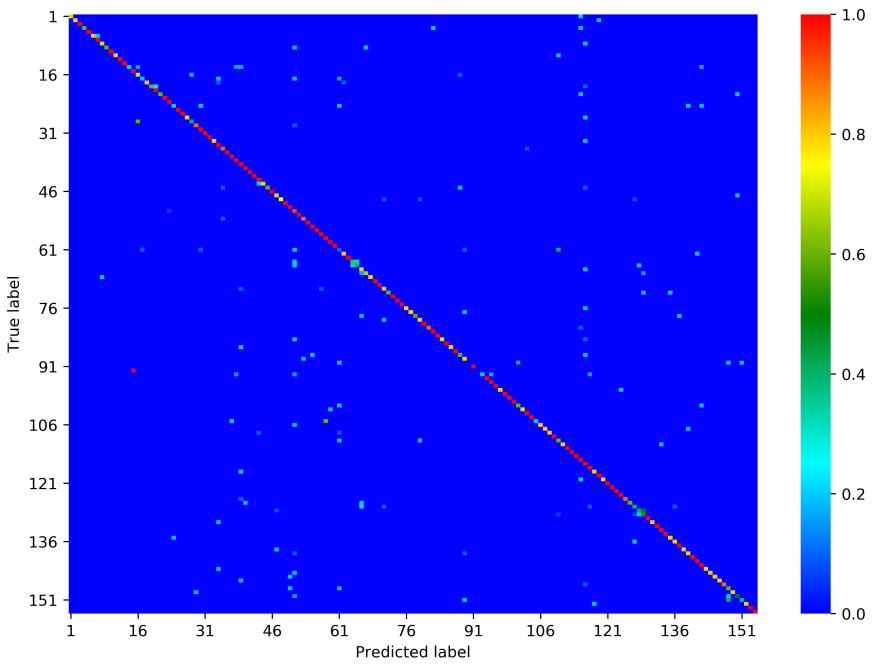

Figure 5.5. Confusion matrix of the experiment in which filtered Improved Dense Trajectories were used on the GD.

We have also seen a similar confusion of signs in the case of *Eat*, *Year* and *Age* signs (see Figure 5.6). However, our system has also started confusing signs where the finger movement and shape is important. Due to the filtering process, we lose the trajectories extracted from fingers. Therefore, our system could not recognize the signs where the finger movement is present. In our opinion, this may be caused by our parameter selection where we need a bigger bounding box around the hand regions for the filtering process.

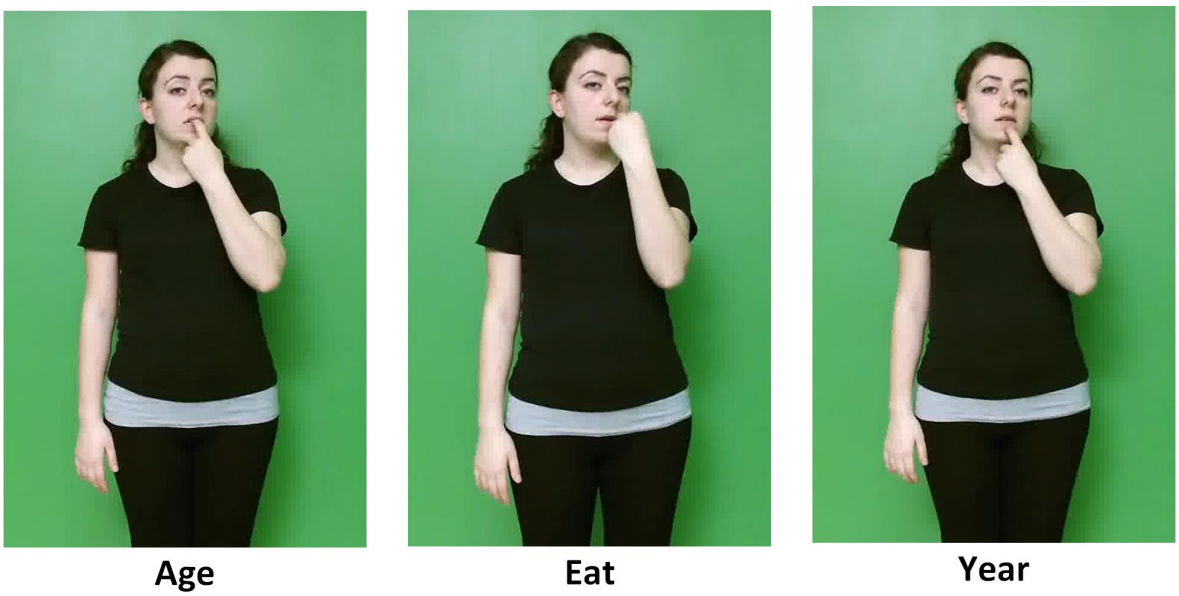

Figure 5.6. Visualization of confused signs; Age, Eat and Year.

To obtain a similar performance as the baseline, we have increased the hand region size slightly. Even though we could not reach the baseline accuracy 87.74%, we showed that reducing the number of trajectories by focusing the trajectories around hands can achieve similar results.

5.3. Experiments with Hand Descriptors

As the second part of our experiments, we have extracted HOG, HOF and MBH descriptors around left and right hands. By extracting these descriptors, we aimed to speed up the feature extraction process without losing too much accuracy. To find

Table 5.6. Performance of the combination of HOG,HOF and MBH descriptors when trajectory filtering is used with baseline parameters $n_t = 3$, $L = 20$ and $k = 64$. (* denotes the best performing method)

Hand Region Size (px)	# Training Trajectories	# Trajectories Per Sign (Avg.)	HOG+HOF+MBH (%)
70 x 70	1,348,751	276.67	83.02
80 x 80	1,672,727	343.12	84.80
90 x 90	1,987,206	407.63	85.95
Entire frame	12,386,261	2,540.77	87.74*

the optimal parameters for descriptors, we performed experiments on the OSD. Then, we used the selected parameters to evaluate our system's performance on the GD. In addition to descriptor parameters, we have also observed the effects of video resolution in terms of speed, feature size and accuracy.

5.3.1. Descriptor Parameters

To explore the effectiveness of the Hand Descriptors, we have conducted experiments using three different parameter sets as in Table 5.7. Since the detail is important for a histogram based representation, we reduced the cell size n_{cell} and increased the block size n_{block} parameters to obtain more detailed representations. However, increasing the dimensions of these descriptors may result in high dimensional Fisher Vectors (FV) for each video which can slow down the training process. Illustration of the hand crops with corresponding Hand Descriptor feature for each parameter set can be seen in Figure 5.7.

In our experiments, we first scaled down the sign videos to 640×360 to speed up the extraction process. Then, we extracted HOG, HOF and MBH descriptors from 80×80 crops which are extracted around left and right hands, for each parameter set defined in Table 5.7. Since these descriptors were extracted from each frame of a video,

Table 5.7. Parameter sets defined for the Hand Descriptors with different n_{cell} and n_{block}

Feature	Pixels per cell (px) (n_{cell})	Cells per block (n_{block})	# bins	Feature Size
HD-1	80 x 80	1 x 1	8	66
HD-2	40 x 40	2 x 2	8	264
HD-3	20 x 20	4 x 4	8	1056

f feature vectors were extracted for each video. Unlike IDT features which have high amounts of trajectories for a video, number of descriptors are reduced drastically with Hand Descriptors.

Figure 5.7. Visualization of Hand Descriptors for each parameter setup defined in Table 5.7.

After extracting the HD features, we plugged them into our pipeline. Results of the experiments can be found in Table 5.8. We have seen that using HD-2 features gave better results than the other feature sets. According to our experiments, the combination of HOG, HOF and MBH descriptors has yielded 94.68% on the OSD which is slightly lower than baseline IDT performance (97.87%).

To see the performance of the descriptors on a larger dataset, we also performed experiments on the GD using best parameter setup obtained from the previous ex-

Table 5.8. Performance of the Hand Descriptors on the OSD given features with different sizes (* denotes the best performing method)

Feature	Feature Size	HOG	HOF	MBH	HOG+HOF	HOG+MBH	HOF+MBH	All
HD-1	66	47.87	79.79	80.85	73.40	71.28	82.98	84.04
HD-2	264	74.47	86.17	89.36	92.55	92.55	89.36	94.68*
HD-3	1056	69.15	85.11	77.66	85.11	79.79	87.23	91.49

periment (see Table 5.9). As a result, our descriptors performed poorly compared to IDT features. This is an expected result because IDT features are highly complex and dense features which can extract thousands of trajectories by using the dense optical flow field.

In addition, IDT features are extracted from 8 spatial scales which make the features more scale invariant. On the contrary, our descriptors are extracted for each frame of a video. Thus, FVs extracted from these descriptors will become high dimensional sparse representations which may achieve poor performances with linear classifiers.

Table 5.9. Performance of the Hand Descriptors on the GD given HD-2 and IDT features (* denotes the best performing method)

Feature	Feature Size	HOG+HOF+MBH (%)
Our method (HD-2)	264	67.30
IDT	396	87.74*

In our experiments, we have seen that some of the confused signs are the same as the signs which were confused with each other when using the filtered IDT features. Since we only focused around the hand region in the feature extraction process, this is an expected outcome. The confusion matrix of this experiment is shown in Figure 5.8.

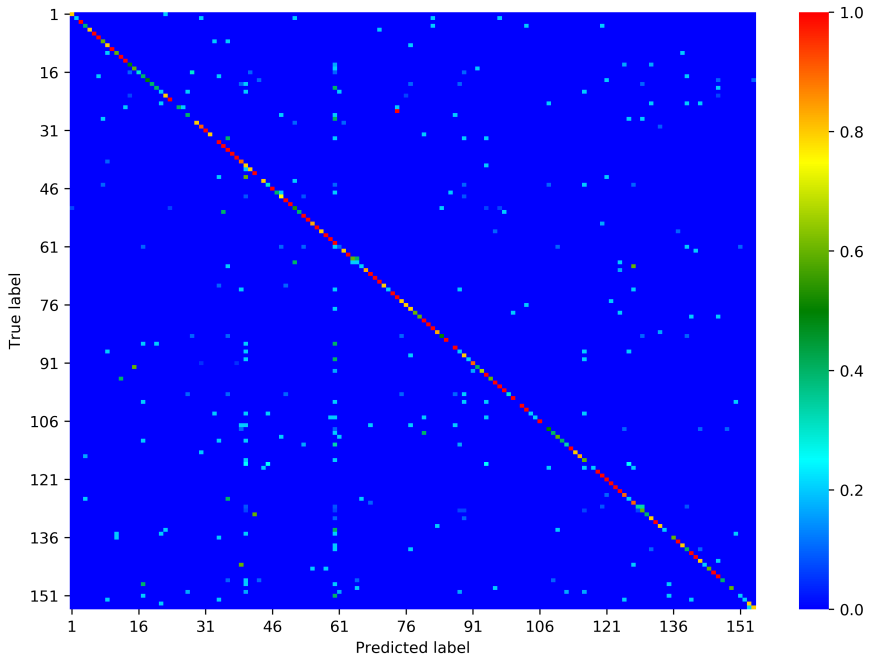

Figure 5.8. Confusion matrix of the experiment in which Hand Descriptors were used on the General Dataset.

As opposed to IDT features, our descriptors makes it harder to distinguish signs where the hand shape, rotation and speed are different. For example, in the case of *Little* and *Some* signs, we have seen that hand rotation is different. In the case of *One week later* and *Week* signs, we have seen that *One week later* sign has an extra finger movement where the signer performing *One* sign just before the *Week*. It also should be noted that these two signs can be successfully classified when we used the IDT features.

In addition, our experiments have shown that the classifier has also started to confuse signs where the hand position is also different. As can be seen in Figure 5.9, *Find* and *Stealing* signs have similar hand movements but hand are in different position. Similar confusion also exists in the case of *Child* and *Pocket* signs.

Figure 5.9. Visualization of confused signs, Top: ground truths; Little, Find and Child, Bottom: predictions; Some, Stealing and Pocket

5.3.2. Video Resolution

Considering the extraction process of spatio-temporal descriptors which highly depends on the pixel information, the resolution of the videos are crucial to the performance of the representations. In this section, we have studied the effects of changing the video resolution to the performance of our descriptors on the OSD.

We have conducted our experiments using the parameter cluster size $k = 64$. As can be seen in Table 5.10, reducing the video resolution had a positive effect on our system in terms of accuracy and speed. Although the accuracy of our system slightly dropped when we used the descriptors individually, it remained similar (94.68%) when the combination of all descriptors were used.

Table 5.10. Effects of video resolution on the performance (prediction accuracy %) and extraction speed of the Hand Descriptors. (* denotes the scale factor 3 for the descriptor parameter cell size n_{cell})

Video Resolution	Features	Extraction Time (avg, seconds)	HOG	HOF	MBH	HOG+HOF+MBH
640 x 360	HD-1	9.81	47.87	79.79	80.85	84.04
	HD-2	9.77	74.47	86.17	89.36	94.68
	HD-3	9.80	69.15	85.11	77.66	91.49
1920 x 1080	HD-1*	97.95	60.64	80.85	71.28	81.91
	HD-2*	96.77	75.53	89.36	87.23	94.68
	HD-3*	97.03	74.47	85.11	84.04	94.68

In addition to accuracy, the descriptors were extracted much faster when we use low resolution videos rather than high resolution ones as expected. While the average extraction time of descriptors from high resolution video is near 100 seconds, it is around 10 seconds when we scaled down the videos. Even though the pixel quality is degraded drastically, our descriptors are practical and efficient to be used in sign

language recognition. Efficiency of the descriptors are explored in more detail in Section 5.5.

5.4. Correcting Inconsistent Samples of the Dataset

In our previous experiments on both IDT and Hand Descriptors, we have noticed that some of the sign videos in the General Dataset have inconsistencies with their labels. We have corrected the labels of these samples given in Table 5.11.

Table 5.11. Analysis of the corrected inconsistent signs in the General Dataset.

Signs	# Corrected Training Samples	# Corrected Test Samples
Ask ↔ Question	75	15
Girl → Woman	26	5
Get up → Stand up	25	5

In the case of *Ask* and *Question* signs, we have found that both of these signs are the same and they have two different signs with same labels. To deal with this problem, we have separated the sign videos of these signs into two different labels. As a result, we have corrected 75 sign videos on the training set and 15 sign videos on the test set.

Secondly, we have seen that *Girl* and *Woman* signs are slightly different where they have different mouthing. However, in our opinion, the difference in these signs cannot also be noticeable broadly on the most the sign samples in the dataset. Therefore, we have changed the sign labels of the videos of *Girl* sign to *Woman* sign. As a result, we have changed 26 sign videos on the training set and 5 sign videos on the test set.

Finally, in our previous experiments, we have seen that *Get up* and *Stand up* signs are the same but they were labelled separately. To correct these sign videos, we

have changed all of the sign videos of *Get up* sign to *Stand up* sign. As a result, we have changed 25 sign videos on the training set and 5 sign videos on the test set.

After we corrected the labels, we have conducted experiments using our pipeline on a clearer version of the General Dataset which have 152 sign classes. Our experiments and their comparisons with the previous experiments are shown in Table 5.12. In our experiments, we have seen that using samples with corrected labels has increased the performance of our system for all descriptors. In addition, our system has successfully classified the inconsistent signs after the label correction process.

Table 5.12. Comparison of the prediction performances (% accuracy) of using corrected General Dataset.

Features	154 Classes (Before)	152 Classes (After)
IDT	87.74	89.30
Filtered IDT	85.95	87.31
Hand Descriptors	67.30	68.34

5.5. Computational Complexity and Memory Requirement

In this section, we explore the computational complexity of IDT and our descriptors when they are used in sign language recognition. We also compare the efficiency of these descriptors in terms of extraction speed and encoding-training cost on the small dataset (OSD). Then, we briefly report the storage and memory needed in order to train these descriptors using our methodology on both datasets.

We first examine the effects of video resolution to the feature extraction speed and number of descriptors extracted. As can be seen in Table 5.13, extracting IDT features at both 640×360 and 1920×1080 resolutions is much slower than extracting HD-2 features. In our experiment, we have also seen that the number of descriptors extracted in the IDT method is still high at 640×360 resolution but significantly lower

than the features extracted at resolution 1920×1080. In the case of HD-2 features, we should note that reducing the resolution of the video has an appreciable effect on the extraction speed while the number of descriptors remain unchanged because features are extracted frame-wise.

Table 5.13. IDT and HD-2 feature extraction speed and number of descriptors for low and high resolution sign videos of the OSD.

Features	# Descriptor Dimensions	Video Duration (avg, seconds)	Video Resolution	Extraction Time Per Video (avg, seconds)	# Descriptors Per Video (avg)
IDT	396	2.82	640 x 360	23.78	2,554.82
			1920 x 1080	224.82	15,851.24
HD-2	264		640 x 360	9.77	82.59
			1920 x 1080	96.77	82.59

The dimensionality of the descriptors is also critical for the encoding and the training steps. In our next experiment, we explore the changes in dimensionality and descriptor count when different descriptors are in our system as inputs. According to Table 5.14, we have seen noticeable changes in the total number of descriptors for the encoding step. Since the number of descriptors are the concatenation of the entire training feature set, it is expected that their number will be much lower when HD-2 features are used.

For this experiment, we should also note that the Fisher Vector dimensionality is $2Dk$ where D is the dimensionality of the descriptors after PCA, and the k is the number of clusters in the GMM. As can be seen in Table 5.14, the dimensionality of the FVs are similar in both IDT and filtered IDT features even with fewer number of trajectories. This is an unexpected result because only focusing around the hand region should be represented with less dimensions since the hand regions contain less information compared to the entire frame.

To examine this problem, we have conducted experiments which can help us to understand the effects of reducing the explained variance of the PCA. Results of this experiment are shown in Table 5.15. In our experiments, we have seen that lowering the variance to 90% gives a similar prediction performance while it reduces the dimensionality by half. The rest of the variance is probably due to noise. Therefore, we can say that hand movements are the most dominant features in sign language.

Table 5.14. Number of total descriptors and dimensionality of the Fisher Vectors used in encoding and training steps of the proposed methodology for each descriptor type.

Feature	# Descriptor Dimensions	Total # of Descriptors	Fisher Vector Dimensions	# of Fisher Vectors
IDT	396	779,221	39,936	305
Filtered IDT	396	137,757	40,448	305
HD-2	264	25,190	27,000	305

Since the number of features extracted with the IDT method is higher than the Hand Descriptors, it is expected that our descriptors will require less storage and memory with the reducing number of features and dimensionality. As can be seen in Table 5.16, HD-2 features require less memory and storage in encoding and training steps on both datasets.

5.6. Notes on Temporal Modelling

Considering the temporal relationship between hand movements in consecutive frames, temporal modelling is an important step for sign language recognition. While most of the work on temporal modelling often uses Dynamic Time Warping (DTW), Hidden Markov Models (HMMs) and recently Long-short Term Memory Networks (LSTM), IDT method simply computes the average of trajectories within a temporal stride. As the temporal averaging reduces the number of trajectories, it also helps the spatio-temporal grid of the tracked trajectory to be more temporally compact.

Table 5.15. Effects of reducing the variance on the performance of the Fisher Vectors.

Explained Variance (%)	# of clusters	Dimensions	Accuracy (%)
80	8	1,312	76,62
	16	2,624	78,83
	32	5,248	81,55
	64	10,496	82,39
90	8	2,224	78,09
	16	4,448	80,29
	32	8,896	82,60
	64	17,792	84,49
99	8	5,040	80,40
	16	10,080	82,70
	32	20,160	83,54
	64	40,320	85,95

Table 5.16. Memory and storage requirements of the descriptors for the training step.

Dataset	Feature	# Descriptor Dimensions	Encoding Memory Requirement (MB)	Training Memory Requirement (MB)
OSD	IDT	396	2,592.01	69.43
	Filtered IDT	396	458.24	70.43
	HD-2	264	50.27	45.69
GD	IDT	396	41,201.86	981.89
	Filtered IDT	396	6,610.28	999.06
	HD-2	264	832.40	634.74

However, we have seen that using temporal averaging on our descriptors had a negative effect. After temporal averaging, the number of our descriptors has been reduced even more and the FVs extracted from those descriptors have became sparser. Thus, these FVs have became unrecognizable by linear classifiers (SVMs).

In addition to temporal averaging, we have also tried to train LSTM networks by feeding our descriptors as sequential inputs. However, LSTM networks did not even converge due to the insufficient number of samples which were used as sequences. In addition, Grushin *et al.* [103] have pointed out that LSTMs achieve better accuracy if the features are extracted using a neural network. Thus, using histogram based features to train LSTM is not an optimal solution for our problem. Therefore, we have decided not use any temporal modelling except the optical flow which was computed from two consequent frames.

6. CONCLUSION

Improved Dense Trajectories is a state-of-the-art trajectory based human action recognition method which can be considered as one of the most successful methods even compared to recent deep learning approaches. Although deep learning methods have became more popular in recent years, the performance of IDT features is still comparable. However, due to its complexity and computational load, IDT features are difficult to extract and store.

In this thesis, we have tried to use the IDT features, adapting and simplifying them for sign language recognition. Sign languages are very different from spoken languages. They use hand gestures, upper body pose information and facial expression in order to form the language. Therefore, we have concentrated on the hand region and extracted HOG, HOF and MBH features which represent the shape and the motion.

In our experiments, we have first extracted IDT features and represent them as Fisher Vectors which are then used as inputs for training a linear SVM to perform sign language classification. After this experiment, we performed a filtering process on IDT features where we remove the trajectories outside the bounding box which was formed around left and right hands. Lastly, we have extracted HOG, HOF and MBH descriptors from hand crops and compared the results with IDT features on two subsets (OSD and General) of the BosphorusSign dataset.

While our system has achieved 98.94% accuracy on the test set of the OSD using the baseline IDT approach, it has achieved 87.74% accuracy on the General dataset. After trajectory filtering, we were able to reach 85.95% accuracy on the General dataset which is a satisfying results because we removed most of the trajectories and focused only the trajectories around the hands. In addition to IDT results, our system has yielded 94.68% accuracy on the OSD when we used the descriptors which were extracted

around the hand region. However, these descriptors have reduced the accuracy of our system by 20% on the General Dataset.

Our experiments have shown that hand region resolution is very crucial in sign language recognition. This is an expected outcome because both IDT features and our descriptors highly depend on the pixel density around the region where they are extracted. Thus, reducing the pixel density leads to a decrease in the quality of optical flow and the pixel intensity gradients which are important for extracting the spatio-temporal descriptors. However, reducing the quality also leads to a increase in extraction speed of the descriptors. Considering this trade-off between speed and quality, we have tried to perform sign language recognition with an optimal set of parameters and descriptors which will give us the best results on both datasets.

Moreover, we have investigated the effects of using IDT features around the hands on sign language recognition. As expected, the hand region produces the most important trajectories. In addition to trajectory filtering, we also proposed to extract HOG, HOF and MBH descriptors around the hand region with the expectation of speeding up the feature extraction process while preserving the accuracy. Our experiments have shown that descriptors extracted around hands lose their characteristics and density with the increasing number of samples while the speed of the feature extraction increases.

Since this is the first major study on the BosphorusSign dataset, we have noticed some inconsistencies with the signs in the dataset. Our experiments have shown that some of the signs were labelled differently even though they have the same arm and hand movements. We have also noticed that some of the signs have various movements within the same sign class. We have corrected the labels on the datasets and report results with the updated protocol as well as the original protocol.

We have also investigated the computational complexity and memory requirement of these descriptors in sign language recognition. We have found that number of

trajectories are extremely low when we use sign videos with the low resolution for feature extraction. Along with the computational complexity, the memory needed for our pipeline has also been reduced drastically when we used sign videos with low resolution.

In all of our experiments, we had tried to adapt the IDT approach to the sign language recognition problem by filtering the trajectories the around hands, extracting the descriptors from the hand region without any dense sampling or using spatial scales, and changing the resolution. While all of these operations have an effect on the feature extraction speed, complexity of the features or the memory needed for the training step, they reduced the classification accuracy of our system.

Last but not least, our experiments shown that IDT and our method are practical to be used in sign language recognition where they can extract valuable representations from hand shape and hand movements of the signer performing the sign. As future work, we are planning to integrate these features into a deep learning model to investigate their behaviour.

REFERENCES

1. Kadous, M. W., "Machine recognition of Auslan signs using PowerGloves: Towards large-lexicon recognition of sign language", *Proceedings of the Workshop on the Integration of Gesture in Language and Speech*, Vol. 165, pp. 165–174, 1996.

2. Hernandez-Rebollar, J. L., R. W. Lindeman and N. Kyriakopoulos, "A multi-class pattern recognition system for practical finger spelling translation", *Multimodal Interfaces, 2002. Proceedings. Fourth IEEE International Conference on*, pp. 185–190, IEEE, 2002.

3. Zhang, L.-G., Y. Chen, G. Fang *et al.*, "A Vision-based Sign Language Recognition System Using Tied-mixture Density HMM", *Proceedings of the 6th International Conference on Multimodal Interfaces*, pp. 198–204, ACM, 2004.

4. Forster, J., C. Schmidt, T. Hoyoux *et al.*, "RWTH-PHOENIX-Weather: A Large Vocabulary Sign Language Recognition and Translation Corpus", *LREC*, pp. 3785–3789, 2012.

5. Chai, X., H. Wanga, M. Zhoub *et al.*, *DEVISIGN: Dataset and Evaluation for 3D Sign Language Recognition*, Tech. rep., Beijing, 2015.

6. Escalera, S., X. Baró, J. Gonzàlez *et al.*, "ChaLearn Looking at People Challenge 2014: Dataset and Results", *Computer Vision - ECCV 2014 Workshops*, pp. 459–473, Springer International Publishing, 2015.

7. Schuldt, C., I. Laptev and B. Caputo, "Recognizing human actions: a local SVM approach", *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, Vol. 3, pp. 32–36, IEEE, August 2004.

8. Rodriguez, M. D., J. Ahmed and M. Shah, "Action MACH a spatio-temporal Maximum Average Correlation Height filter for action recognition", *2008 IEEE*

Conference on Computer Vision and Pattern Recognition, pp. 1–8, IEEE, June 2008.

9. Kuehne, H., H. Jhuang, E. Garrote *et al.*, "HMDB: A large video database for human motion recognition", *2011 International Conference on Computer Vision*, pp. 2556–2563, November 2011.

10. Soomro, K., A. R. Zamir and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild", *arXiv:1212.0402*, 2012.

11. Abu-El-Haija, S., N. Kothari, J. Lee *et al.*, "Youtube-8m: A large-scale video classification benchmark", *arXiv:1609.08675*, 2016.

12. Kay, W., J. Carreira, K. Simonyan *et al.*, "The kinetics human action video dataset", *arXiv:1705.06950*, 2017.

13. Wang, H., A. Kläser, C. Schmid *et al.*, "Action recognition by dense trajectories", *2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3169–3176, IEEE, June 2011.

14. Wang, H. and C. Schmid, "Action Recognition with Improved Trajectories", *2013 IEEE International Conference on Computer Vision*, pp. 3551–3558, IEEE, December 2013.

15. Peng, X., L. Wang, Z. Cai *et al.*, "Action and Gesture Temporal Spotting with Super Vector Representation", *Computer Vision - ECCV 2014 Workshops*, pp. 518–527, Springer International Publishing, 2015.

16. Vogler, C. and D. Metaxas, "Adapting hidden Markov models for ASL recognition by using three-dimensional computer vision methods", *1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*, Vol. 1, pp. 156–161, October 1997.

17. Liang, R. and M. Ouhyoung, "A sign language recognition system using hidden markov model and context sensitive search", *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, pp. 59–66, Hongkong, 1996.

18. Kim, J.-S., W. Jang and Z. Bien, "A dynamic gesture recognition system for the Korean sign language (KSL)", *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, Vol. 26, No. 2, pp. 354–359, April 1996.

19. Simpson, P. K., "Fuzzy min-max neural networks", *Proceedings of the IEEE International Joint Conference on Neural Networks*, Vol. 2, pp. 1658–1669, November 1991.

20. Hienz, H., B. Bauer and K.-F. Kraiss, "HMM-based continuous sign language recognition using stochastic grammars", *Gesture-Based Communication in Human-Computer Interaction*, pp. 185–196, Springer Berlin Heidelberg, 1999.

21. Holden, E.-J. and R. Owens, "Visual Sign Language Recognition", *Multi-Image Analysis*, pp. 270–287, Springer Berlin Heidelberg, 2001.

22. Imagawa, K., S. Lu and S. Igi, "Color-based hands tracking system for sign language recognition", *Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 462–467, IEEE, April 1998.

23. Grzeszcuk, R., G. Bradski, M. H. Chu *et al.*, "Stereo based gesture recognition invariant to 3D pose and lighting", *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000*, Vol. 1, pp. 826–833, 2000.

24. Goh, P. and E. j. Holden, "Dynamic Fingerspelling Recognition using Geometric and Motion Features", *2006 International Conference on Image Processing*, pp. 2741–2744, October 2006.

25. Cooper, H. and R. Bowden, "Large Lexicon Detection of Sign Language", *Human–Computer Interaction*, pp. 88–97, Springer Berlin Heidelberg, 2007.

26. Liwicki, S. and M. Everingham, "Automatic recognition of fingerspelled words in British Sign Language", *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 50–57, June 2009.

27. Zhang, Z., "Microsoft Kinect Sensor and Its Effect", *IEEE MultiMedia*, Vol. 19, No. 2, pp. 4–10, February 2012.

28. Ho, T. K., "Random decision forests", *Proceedings of 3rd International Conference on Document Analysis and Recognition*, Vol. 1, pp. 278–282, August 1995.

29. Shotton, J., A. Fitzgibbon, M. Cook *et al.*, "Real-time human pose recognition in parts from single depth images", *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011*, pp. 1297–1304, June 2011.

30. Kadir, T., R. Bowden, E.-J. Ong *et al.*, "Minimal Training, Large Lexicon, Unconstrained Sign Language Recognition.", *British Machine Vision Conference*, pp. 1–10, 2004.

31. Freund, Y. and R. E. Schapire, "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting", *Journal of Computer and System Sciences*, Vol. 55, No. 1, pp. 119–139, 1997.

32. Viola, P. and M. J. Jones, "Robust Real-Time Face Detection", *International Journal of Computer Vision*, Vol. 57, pp. 137–154, May 2004.

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

34. Wong, S.-F. and R. Cipolla, "Real-Time Adaptive Hand Motion Recognition Using a Sparse Bayesian Classifier", *Computer Vision in Human-Computer Interaction*, pp. 170–179, Springer Berlin Heidelberg, 2005.

35. Bradski, G. R. and J. W. Davis, "Motion segmentation and pose recognition with motion history gradients", *Machine Vision and Applications*, Vol. 13, pp. 174–184, July 2002.

36. Bobick, A. and J. Davis, "Real-time recognition of activity using temporal templates", *Applications of Computer Vision, 1996. WACV '96., Proceedings 3rd IEEE Workshop on*, pp. 39–42, IEEE, December 1996.

37. Bobick, A. and J. Davis, "An appearance-based representation of action", *Proceedings of 13th International Conference on Pattern Recognition*, Vol. 1, pp. 307–312, August 1996.

38. Nandakumar, K., K. W. Wan, S. M. A. Chan *et al.*, "A Multi-modal Gesture Recognition System Using Audio, Video, and Skeletal Joint Data", *Proceedings of the 15th ACM on International Conference on Multimodal Interaction*, ICMI '13, pp. 475–482, ACM, 2013.

39. Laptev, I., "On Space-Time Interest Points", *International Journal of Computer Vision*, Vol. 64, pp. 107–123, September 2005.

40. Camgöz, N. C., A. A. Kındıroğlu and L. Akarun, "Sign Language Recognition for Assisting the Deaf in Hospitals", *International Workshop on Human Behavior Understanding*, pp. 89–101, Springer, 2016.

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

42. Rabiner, L. and B. Juang, "An introduction to hidden Markov models", *IEEE ASSP Magazine*, Vol. 3, No. 1, pp. 4–16, January 1986.

43. Berndt, D. J. and J. Clifford, "Using Dynamic Time Warping to Find Patterns in Time Series", *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, Vol. 10, pp. 359–370, AAAI Press, 1994.

44. Starner, T., J. Weaver and A. Pentland, "Real-time American sign language recognition using desk and wearable computer based video", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 12, pp. 1371–1375, December 1998.

45. Vogler, C. and D. Metaxas, "Handshapes and Movements: Multiple-Channel American Sign Language Recognition", *Gesture-Based Communication in Human-Computer Interaction*, pp. 247–258, Springer Berlin Heidelberg, 2004.

46. Chai, X., G. Li, X. Chen *et al.*, "VisualComm: A Tool to Support Communication Between Deaf and Hearing Persons with the Kinect", *Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility*, p. 76, ACM, 2013.

47. Keskin, C., A. T. Cemgil and L. Akarun, "DTW Based Clustering to Improve Hand Gesture Recognition", *International Workshop on Human Behavior Understanding*, pp. 72–81, Springer Berlin Heidelberg, 2011.

48. Camgöz, N. C., *Human-Computer Interaction Platform for the Hearing Impaired in Healthcare and Finance Applications*, Master's Thesis, Boğaziçi University, 2016.

49. Hasanuzzaman, M., V. Ampornaramveth, T. Zhang *et al.*, "Real-time Vision-based Gesture Recognition for Human Robot Interaction", *2004 IEEE International Conference on Robotics and Biomimetics*, pp. 413–418, August 2004.

50. Corradini, A., "Dynamic time warping for off-line recognition of a small gesture vocabulary", *Proceedings IEEE ICCV Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, pp. 82–89, 2001.

51. Wang, S. B., A. Quattoni, L. P. Morency *et al.*, "Hidden Conditional Random Fields for Gesture Recognition", *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, Vol. 2, pp. 1521–1527, IEEE, 2006.

52. Camgöz, N. C., A. A. Kindiroglu and L. Akarun, "Gesture Recognition Using Template Based Random Forest Classifiers", *Computer Vision - ECCV 2014 Workshops*, pp. 579–594, Springer International Publishing, 2015.

53. Breiman, L., "Random forests", *Machine learning*, Vol. 45, No. 1, pp. 5–32, October 2001.

54. Pigou, L., A. van den Oord, S. Dieleman *et al.*, "Beyond Temporal Pooling: Recurrence and Temporal Convolutions for Gesture Recognition in Video", *International Journal of Computer Vision*, Vol. 126, No. 2, pp. 430–439, April 2018.

55. Kang, B., S. Tripathi and T. Q. Nguyen, "Real-time sign language fingerspelling recognition using convolutional neural networks from depth map", *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, pp. 136–140, IEEE, November 2015.

56. Koller, O., H. Ney and R. Bowden, "Deep Hand: How to Train a CNN on 1 Million Hand Images When Your Data is Continuous and Weakly Labelled", *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3793–3802, IEEE, June 2016.

57. Molchanov, P., S. Gupta, K. Kim *et al.*, "Hand gesture recognition with 3D convolutional neural networks", *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1–7, IEEE, June 2015.

58. Camgoz, N. C., S. Hadfield, O. Koller *et al.*, "Using Convolutional 3D Neural Networks for User-independent continuous gesture recognition", *2016 23rd International Conference on Pattern Recognition (ICPR)*, pp. 49–54, IEEE, December 2016.

59. Neverova, N., C. Wolf, G. Paci *et al.*, "A Multi-scale Approach to Gesture Detection and Recognition", *2013 IEEE International Conference on Computer Vision Workshops*, pp. 484–491, IEEE, December 2013.

60. Neverova, N., C. Wolf, G. Taylor *et al.*, "ModDrop: Adaptive Multi-Modal Gesture Recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 38, No. 8, pp. 1692–1706, August 2016.

61. Camgoz, N. C., S. Hadfield, O. Koller *et al.*, "Neural Sign Language Translation", *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, in press.

62. Neubig, G., "Neural machine translation and sequence-to-sequence models: A tutorial", *arXiv:1703.01619*, 2017.

63. Camgöz, N. C., A. A. Kindiroglu, S. Karabüklü *et al.*, "BosphorusSign: A Turkish Sign Language Recognition Corpus in Health and Finance Domains", *LREC*, 2016.

64. Escalante, H. J., V. Ponce-López, J. Wan *et al.*, "ChaLearn Joint Contest on Multimedia Challenges Beyond Visual Analysis: An overview", *2016 23rd International Conference on Pattern Recognition (ICPR)*, pp. 67–73, IEEE, December 2016.

65. Galvin, B., B. Mccane, K. Novins *et al.*, "Recovering Motion Fields: An Evaluation of Eight Optical Flow Algorithms", *British Machine Vision Conference*, Vol. 98, pp. 195–204, 1998.

66. Laptev, I., M. Marszalek, C. Schmid *et al.*, "Learning realistic human actions from movies", *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, IEEE, June 2008.

67. Dalal, N., B. Triggs and C. Schmid, "Human Detection Using Oriented Histograms of Flow and Appearance", *Computer Vision – ECCV 2006*, pp. 428–441, Springer Berlin Heidelberg, 2006.

68. Sánchez, J., F. Perronnin, T. Mensink *et al.*, "Image Classification with the Fisher Vector: Theory and Practice", *International Journal of Computer Vision*, Vol. 105, pp. 222–245, December 2013.

69. Sivic, J. and A. Zisserman, "Efficient Visual Search of Videos Cast as Text Retrieval", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 31, pp. 591–606, April 2009.

70. Gaidon, A., Z. Harchaoui and C. Schmid, "Activity representation with motion hierarchies", *International Journal of Computer Vision*, Vol. 107, pp. 219–238, May 2014.

71. Vrigkas, M., V. Karavasilis, C. Nikou *et al.*, "Matching Mixtures of Curves for Human Action Recognition", *Comput. Vis. Image Underst.*, Vol. 119, pp. 27–40, February 2014.

72. Quattoni, A., S. Wang, L. P. Morency *et al.*, "Hidden Conditional Random Fields", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 29, pp. 1848–1852, October 2007.

73. Wang, Y. and G. Mori, "Max-margin hidden conditional random fields for human action recognition", *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 872–879, IEEE, June 2009.

74. Maji, S., L. Bourdev and J. Malik, "Action recognition from a distributed representation of pose and appearance", *2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3177–3184, June 2011.

75. Rahmani, H., A. Mahmood, D. Q. Huynh *et al.*, "Real time action recognition using histograms of depth gradients and random decision forests", *IEEE Winter Conference on Applications of Computer Vision*, pp. 626–633, IEEE, March 2014.

76. Krizhevsky, A., I. Sutskever and G. E. Hinton, "Imagenet classification with deep convolutional neural networks", *Advances in neural information processing systems*, pp. 1097–1105, 2012.

77. Deng, J., A. Berg, S. Satheesh *et al.*, "ILSVRC-2012", `http://www.image-net.org/challenges/LSVRC`, 2012, accessed at June 2018.

78. Lecun, Y., L. Bottou, Y. Bengio *et al.*, "Gradient-based learning applied to document recognition", *Proceedings of the IEEE*, Vol. 86, pp. 2278–2324, November 1998.

79. Karpathy, A., G. Toderici, S. Shetty *et al.*, "Large-Scale Video Classification with Convolutional Neural Networks", *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1725–1732, June 2014.

80. Donahue, J., L. A. Hendricks, S. Guadarrama *et al.*, "Long-term recurrent convolutional networks for visual recognition and description", *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2625–2634, June 2015.

81. Hochreiter, S. and J. Schmidhuber, "Long Short-Term Memory", *Neural Computation*, Vol. 9, pp. 1735–1780, November 1997.

82. Simonyan, K. and A. Zisserman, "Two-stream Convolutional Networks for Action Recognition in Videos", *Proceedings of the 27th International Conference on Neural Information Processing Systems*, Vol. 1, pp. 568–576, MIT Press, 2014.

83. Goodale, M. A. and A. D. Milner, "Separate visual pathways for perception and action", *Trends in neurosciences*, Vol. 15, No. 1, pp. 20–25, 1992.

84. Kuehne, H., H. Jhuang, E. Garrote *et al.*, "HMDB: A large video database for human motion recognition", *2011 International Conference on Computer Vision*, pp. 2556–2563, IEEE, November 2011.

85. Wu, Z., Y.-G. Jiang, X. Wang *et al.*, "Multi-Stream Multi-Class Fusion of Deep Networks for Video Classification", *Proceedings of the 2016 ACM on Multimedia Conference*, pp. 791–800, ACM, 2016.

86. Wang, L., Y. Qiao and X. Tang, "Action recognition with trajectory-pooled deep-convolutional descriptors", *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4305–4314, June 2015.

87. Ng, J. Y.-H., M. Hausknecht, S. Vijayanarasimhan *et al.*, "Beyond short snippets: Deep networks for video classification", *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4694–4702, IEEE, June 2015.

88. Tran, D., L. Bourdev, R. Fergus *et al.*, "Learning Spatiotemporal Features with 3D Convolutional Networks", *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 4489–4497, IEEE, December 2015.

89. Carreira, J. and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset", *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4724–4733, IEEE, 2017.

90. Ioffe, S. and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift", *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, pp. 448–456, JMLR.org, 2015.

91. Bay, H., T. Tuytelaars and L. Van Gool, "SURF: Speeded Up Robust Features", *Computer Vision – ECCV 2006*, pp. 404–417, Springer Berlin Heidelberg, 2006.

92. Fischler, M. A. and R. C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography", *Commun. ACM*, Vol. 24, No. 6, pp. 381–395, June 1981.

93. Csurka, G., C. R. Dance, L. Fan *et al.*, "Visual categorization with bags of keypoints", *In Workshop on Statistical Learning in Computer Vision, ECCV*, Vol. 1, pp. 1–2, 2004.

94. Jégou, H., M. Douze, C. Schmid *et al.*, "Aggregating local descriptors into a compact image representation", *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 3304–3311, IEEE, June 2010.

95. Titterington, D. M., A. F. Smith and U. E. Makov, *Statistical analysis of finite mixture distributions*, Wiley, 1985.

96. Perronnin, F. and C. Dance, "Fisher Kernels on Visual Vocabularies for Image Categorization", *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, IEEE, June 2007.

97. Perronnin, F., J. Sánchez and T. Mensink, "Improving the Fisher Kernel for Large-Scale Image Classification", *Computer Vision – ECCV 2010*, pp. 143–156, Springer Berlin Heidelberg, 2010.

98. Cortes, C. and V. Vapnik, "Support-vector networks", *Machine Learning*, Vol. 20, No. 3, pp. 273–297, September 1995.

99. Vapnik, V., *The nature of statistical learning theory*, Springer science & business media, 2013.

100. Özdemir, O., N. C. Camgöz and L. Akarun, "Isolated sign language recognition using Improved Dense Trajectories", *2016 24th Signal Processing and Communication Application Conference (SIU)*, pp. 1961–1964, IEEE, May 2016.

101. Özdemir, O., A. A. Kindiroglu and L. Akarun, "Isolated sign language recognition with fast hand descriptors", *2018 26th Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4, IEEE, May 2018.

102. Farnebäck, G., "Two-Frame Motion Estimation Based on Polynomial Expansion", *Scandinavian conference on Image Analysis*, pp. 363–370, Springer Berlin Heidelberg, 2003.

103. Grushin, A., D. D. Monner, J. A. Reggia *et al.*, "Robust human action recognition via long short-term memory", *The 2013 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, August 2013.