DETECTION OF SIGN-LANGUAGE CONTENT IN VIDEO THROUGH POLAR MOTION PROFILES

A Thesis

by

VIRENDRA KARAPPA

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Ricardo Gutierrez-Osuna
Committee Members,	Frank Shipman
	Takashi Yamauchi
Head of Department,	Dilma Da Silva

December 2014

Major Subject: Computer Engineering

ABSTRACT

Locating sign language (SL) videos on video sharing sites (e.g., YouTube) is challenging because search engines generally do not use the visual content of videos for indexing. Instead, indexing is done solely based on textual content (e.g., title, description, metadata etc.). As a result, untagged SL videos do not appear in the search results. In this thesis, we present and evaluate an approach to detect SL content in videos based on their visual content. Our work focuses on detection of SL content and not on transcription. Our approach relies on face detection and background modeling techniques, combined with a head-centric polar representation of hand movements. The approach uses an ensemble of Haar-based face detectors to define regions of interest (ROI) and a probabilistic background model to segment movements in the ROI. The resulting two-dimensional (2D) distribution of foreground pixels in the ROI is then reduced to two 1D polar motion profiles (PMPs) by means of a polar-coordinate transformation. These profiles are then used for classification of SL videos from others.

We evaluate three distinct approaches to process information from the PMPs for classification/detection of SL videos. In the first method, we average out the PMPs across all the ROIs to obtain a single PMP vector for each video. These vectors are then used as input features for an SVM classifier. In the second method, we follow the bag-of-words approach of information retrieval to compute a distribution of PMPs (bag-of-PMPs) for each video. In the third method, we perform linear discriminant analysis (LDA) of PMPs and use the distribution of PMPs projected in the LDA space for classification. When evaluated on a dataset comprising of 205 videos (obtained from YouTube), the average PMP approach achieves a precision of 81% and recall of 94%, whereas the bag-of-PMPs approach leads to a precision of 72% and recall of 70%. In contrast to the first two methods, supervised feature extraction by the third method achieves a higher

precision (84%) and recall (94%).

Though this thesis presents a successful means by which to detect sign language in videos, our approaches do not consider temporal information, only the distribution of profiles for a given video. Future work should consider extracting temporal information from the sequence of PMPs to utilize the dynamic signatures of sign languages and potentially improve retrieval results. The SL detection techniques presented in this thesis may be used as an automatic tagging tool to annotate user-contributed videos in sharing sites such as YouTube, in this way making sign-language content more accessible to members of the deaf community.

ACKNOWLEDGEMENTS

I thank my advisor Dr. Ricardo Gutierrez-Osuna for his immense support and encouragement throughout my graduate studies. His advice and guidance has been of great help in shaping this thesis. I admire his commitment and enthusiasm.

I would also like to thank Dr. Frank Shipman for his idea of automatic tagging of sign language videos. His support and suggestions have been of great help in my research. I also thank Dr. Takashi Yamauchi for his interest in my research and for his feedback.

Lastly, I would like to thank my lab mates Avinash Parnandi, Sandesh Aryal, and Chris Liberatore for their help in editing this thesis. I would also like to thank all my lab mates for being a great support and making my graduate life fun.

TABLE OF CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

NOMENCLATURE

The following table describes the significance of various abbreviations and acronyms used throughout the thesis.

Abbreviation	Meaning
PMP	Polar motion profile
PPMP	Polar motion profile projected onto PCA subspace
LPMP	Polar motion profile projected onto LDA subspace
ROI	Region of interest around signer's face

1. INTRODUCTION*

Sign Languages (SL) rely on hand gestures combined with facial expressions and body postures to convey their message. They are the primary medium of communication for many who are deaf and hard-of-hearing [1], and serve as a substitute for spoken communication. Because sign language is a visual form of communication, video sharing websites can be very beneficial to the deaf community as a means to exchange information.

The number of videos on the web containing sign language is increasing rapidly, but only a small subset of these videos are easily available to the deaf community. The main reason for this mismatch is that search engines index videos based only on their associated metadata (e.g., text descriptions, tags). However, for many SL videos the metadata is associated with the topic (e.g., sports, politics) rather than the language being used (i.e., American Sign Language). Therefore, such videos do not show up in the search results when performing standard text queries with keywords such as American Sign Language, British Sign Language, etc. Given the size of user-contributed video sites, manual tagging is prohibitive. Instead, meaningful improvement of search results requires automated tagging. This in turn requires algorithms to detect sign language content based on visual information alone.

In this thesis we propose techniques to detect SL content in user contributed videos. Our approach relies on face detection and background modeling techniques, combined with a head-centric polar representation of hand movements. In a first step, we detect faces in the videos using an ensemble of face detectors based on Haar-like features [2, 3]. We then extract foreground hand gestures by

---

subtracting a background model based on adaptive Gaussian mixtures [4]. For each frame, we then calculate the proportion of foreground pixels along the two polar coordinates (radial, angular) on a reference frame centered on the signer's face and scaled to the face's proportions to provide translation and scale invariance. These polar motion profiles (PMPs) capture the amount of signing activity in each frame of the video.

We explore three different approaches to extract motion information from the PMPs of the videos and evaluate their SL classification performance using an SVM classifier:

(1) **Average PMP:** In the first approach, we compute the average polar motion profile across all ROIs in the video. As a result, each video is represented by a single feature vector that captures the average amount of motion around faces in the video.

(2) **Bags of PMPs:** The second approach extracts additional information by representing each video by the distribution of PMPs across frames through k-means clustering. As a result, each video is represented by a histogram containing the number of times each k-means cluster occurs in the video. Given its similarity with the bag-of-words in text retrieval, we denote this method as bag-of-PMPs. As will be described later, we compare two forms of clustering, one that operates in the original PMP space, and a second that performs principal components to de-correlate the PMP axis and reduce dimensionality to a manageable size.

(3) Supervised projection of PMPs: The third approach leverages class label information to maximize separation between SL and non-SL videos. Namely, we apply linear discriminant analysis (LDA) to project information from PMPs into a one-dimensional subspace. As a result, each video is represented by a one-dimensional (1D) sequence, i.e., the LDA projection of each ROI in the video. As before, we compare two strategies for this purpose. The first strategy performs k-means clustering to the resulting 1D sequence

to obtain a bag-of-LPMPs for each video. In contrast, the second strategy compresses the resulting 1D sequence into a vector of first-order and higher order statistics.

The main contributions of this thesis can be summarized as follows:

- We developed polar motion profiles (PMPs), a polar representation to capture motion information in a head-centric reference frame
- We evaluated techniques to extract information from PMPs for discrimination of SL and non-SL videos
- We evaluated the proposed techniques on a large corpus containing 400 videos retrieved from the web.

The rest of the document is organized as follows. Chapter 2 provides a summary of past work on SL recognition and its applicability to our study. Chapter 3 illustrates the limitations of current text-based mechanisms when used to locate SL videos. Chapter 4 describes the SL detection methods proposed in this thesis. Chapter 5 provides the details of dataset creation and results for the proposed SL detection methods. Chapter 6 draws conclusions from the work that led to this thesis and provides direction for future work.

2. BACKGROUND / RELATED WORK*

A sign language is a language which uses hand signs, gestures and body language to convey messages. This can involve simultaneously combining of hand shapes, orientation and movement of the hands, arms or body and facial expressions to fluidly express speaker's thoughts. The majority of the work concerning sign language videos focuses on the modelling these gestures for transcription (i.e., recognizing the specific signs being made). In this chapter, we provide an overview of such methods, which largely focus on modeling hand gestures and facial expressions. The majority of the methods discussed here use hand gestures for modelling signs for transcription, whereas only a few methods have used facial expressions.

2.1 Temporal modeling of signs using HMMs

In one of the earliest studies, Starner et al [5] developed an HMM classifier capable of recognizing 40 American Sign Language (ASL) words for a single signer. They presented two real-time Hidden Markov Model-based systems for recognizing sentence-level continuous American Sign Language (ASL) using a single camera to track the user's hands. The first system observed the user from a desk mounted camera and achieved 92% word accuracy. The second system mounted a camera in a hat worn by the user and achieved 98% accuracy. In a related work by Vogler and Metaxas [6] parallel HMMs were used to scale the vocabulary size. They demonstrated that parallel HMMs could outperform regular HMMs while preserving scalability. Using a vocabulary of 22 signs and a set of 400-sentences, the authors report a recognition accuracy of 94%. In a more recent study, Bowden et al. [7] proposed a sign language recognition system, which modeled

---

temporal sign transitions using a classifier bank of Markov chains combined with Independent Component Analysis. This approach achieved 98% classification rate on a lexicon of 43 words. In another method, Holden et. al [8] used geometrical features such as the direction of hand movements, angles between the movements, and roundedness of hand blobs as basic units of signs for training HMMs. On a test set of 163 signs, this approach achieved 97% recognition rate at the sentence level and 99% success at the word level.

2.2 Modeling of motion trajectories

Some other approaches involve modelling of motion trajectories of the signs. Yang et al. [9] presented an algorithm to extract and classify two-dimensional motion in an image sequence using motion trajectories. In the first step, a multi-scale segmentation was performed to generate homogenous regions in each frame of a video. In the second step, regions between consecutive frames were then matched to obtain pixel-level matches. These matches were concatenated to obtain motion trajectories across the image sequence. Finally, a time-delay network was used to learn motion patterns from the extracted trajectories. This network achieved 93% recognition on unseen test trajectories.

One of the difficult problems while modelling motion trajectories is segmenting out the gestures in-between any two signs, which generally convey no meaning. Such gesture movements that bridge two consecutive signs are known as movement epenthesis (ME). Yang et al. [10] tackled this problem of modelling movement epenthesis by using a dynamic-programming framework called Level building. In this approach, each level corresponds to the possible order of signs or ME in the test sentence. The first level is concerned with the first possible label in the sentence, and so on. At each level a best possible match for each combination of end point is stored (memorized) from the previous level. The optimal sequence of signs and ME labels is constructed

by backtracking. This approach achieved 83% word level recognition on a single view video dataset of continuous sign languages.

2.3 Content-based image retrieval

Another approach for sign language recognition uses a content-based image retrieval paradigm, where each sign language gesture is considered to be a series of images from the SL video. The set of images for each of the gesture samples are stored in an image database. Recognition of sign involves searching finding the closest match to the input images in the database. In one such approach, Dimov et al. [11] treated sign language recognition as a content-based image retrieval problem by searching the input sign image in a database of signs in the form of static images. This method achieved a word-level error rate of less than 4%.

In a similar approach, Potamias et al. [12] proposed a content based retrieval approach by storing tens of thousands of hand-shape images in a large database. They compared brute force retrieval with two other image indexing and retrieving strategies, viz. BoostMap embedding and Distance-Based Hashing. Their goal was to achieve faster search and lookup of the images in the database. Compared to brute-force search, their approach improved retrieval time by three orders of magnitude. However, they used a nearest neighbor approach to match hand-shape images, which limited classification rate to 33%.

2.4 Modeling of sub-units of signs

A few approaches have used parts of signs (sub-units) instead of whole signs for training sign language recognition models. In one such approach, Bauer et al. [13] trained HMMs using subunits of signs instead of whole signs. In a first experiment, the authors achieved a recognition rate of 93% on a vocabulary of 100 signs. For the second experiment, HMMs trained on the subunits of

the 100 signs were used to spot sign language for 50 new signs, resulting in an accuracy of 81%.

In a related approach, Nayak et.al [14] used unsupervised techniques to learn basic units of signs from continuous sentences. Given a set of sentences, a common sign model was learned using a dynamic-programming framework. The model was represented in a space of relational distributions capturing spatial relationships between low-level image features (e.g. edge points). This approach was used to build models of 18 signs from the Boston SignStream Dataset [15]. An experiment was conducted with 15 sentences with lengths varying from 3 to 15 signs. Since the number of signs involved was small, a decision rate wasn't reported on the experiments.

2.5 Data gloves

Most of the methods discussed so far are only suited for relatively small vocabularies. Recognition of larger vocabularies generally requires data gloves to provide precise information of hand movements. As an example, Liang et al. [16] used data gloves to recognize Taiwanese sign language gestures from a vocabulary of between 71 and 250 words. Their system required that gestures were performed slowly in order to detect word boundaries. Using this system, sentences of gestures based on these vocabularies could be continuously recognized in real-time with an average recognition rate of 80%.

Along similar lines, Braffort et al. [17] developed a system to recognize French Sign Language using data gloves. Features of hand position and appearance were extracted from the data gloves. A vocabulary of seven signs was used for experiments. One classifier each for conventional and non-conventional signs was trained to achieve 96% accuracy. Fang et al. [18] used 18-sensor data-gloves and three position trackers to extract hand motion information. The data-gloves collected the variation information of hand shapes with the 18-dimensional data at each hand, and the position trackers collected the variation information of orientation, position, movement trajectory.

Using this information, a simple recurrent network (SRN) was trained to segment signs and feed to the hidden Markov models (HMM). Outputs from the SRN were treated as states in an HMM, and the Lattice Viterbi algorithm employed to search the best word sequence. On a test set of 367 signs, this method achieved a recognition accuracy of 93%.

In one of the later works, Gao et al. [19, 20] proposed a Chinese Language Recognition system with a data glove as an input device. Using a state-tying HMM as the recognition model, the authors report a 95% recognition rate on a vocabulary of 5177 Chinese signs.

2.6 Hand posture and orientation

Even with data gloves, recognizing the signer's hand gestures may be difficult because they can adopt a variety of postures and orientations while signing. Somers and Whyte [21] dealt with this problem by using 3D models and silhouettes to identify accurate postures. The 3D-models were oriented at run-time to match the orientation of the signer's hand. A two-camera setup was used to match two silhouettes created from images taken at two diverse angles. This method was used to identify 4 postures of hands, out of which 3 were identified correctly by matching silhouette of image from at least one of the cameras.

2.7 Modelling of non-manual (facial) gestures

Most of the methods for sign language recognition use hand gestures and trajectories. Very few approaches involve recognition of non-manual (facial) gestures in sign language. Head movements and facial expressions are a critical part of sign language expressions. In one such approach, Erden et al. [22] developed a system to detect non-manual gestures that occur in parallel with manual gestures (hand signs and gestures). This system uses a head tracker [23] to extract rotation and translation parameters from a monocular video. These parameters were then analyzed to detect

"Head nods" and "Head shakes," and results were compared with the labels assigned by ASL linguists. They tested this system on ten ASL sequences labelled by ASL linguists. On this test, this approach missed only one head gesture for the first five sentences, zero for the next two and four for the last two sentences.

In sign language, negation is expressed through facial expressions. For example a sentence 'I don't know' is signed exactly same as 'I know', except that there is a distinct 'head shake' indicating 'Negation'. Thus, detecting negation is one of the harder problems in SL recognition. Parashar et al. [24] used motion trajectories of the face to detect 'Negation' in an ASL sentence. Motion trajectories of face were obtained by tracking the centers of the eyes in the consequent images in a stream. Using facial trajectory information, performance was improved from 88% to 92% on a vocabulary with 6 signs.

2.8 Sign language detection

Approaches developed for sign language transcription are of limited value in our context in that most of them work only modestly with relatively small vocabularies, or are signer-dependent and require large amounts of training data. In contrast, our work focuses on detecting sign language content in videos and not transcription. As an example, Cherniavsky et al. [25] developed an activity detection technique for cell-phone cameras that could determine whether a user was signing or not with 91% accuracy, even in the presence of noisy (i.e., moving) backgrounds. The algorithm was used to determine when the video phone user was signing and when they were watching the video of their conversational partner in order to effectively use network bandwidth during a sign language conversation on mobile devices. Thus, it is unlikely this algorithm would be as successful in distinguishing between sign language videos and other videos involving people gesturing.

The work proposed in this thesis grows out of a pilot study published at the ASSETS conference in 2012 by Monteiro et al. [26]. The objective of the study was to establish proof-of-concept for the feasibility of detecting SL in videos. For that reason, that study was performed on a constrained video dataset with videos containing a single signer and a static background, with movements being mainly those of the signer. This allowed the investigators to use a low-pass filter as a background detection model and a simple feature-extraction technique that computed the amount and symmetry of movements around the face. In later work, Shipman et al. [27] estimated an improvement in SL video classification from 42% for text-based queries up to 75% for queries that included video content.

The work proposed here relaxes assumptions of the study by Monteiro et al. [26] by considering videos that contain multiple signers and complex non-stationary backgrounds. This required more robust techniques for face detection and background modeling, as well as a richer feature representation of hand movements.

3. ASSESSING PROBLEMS WITH TEXT BASED SEARCH OF SL VIDEOS

The sign-language community relies on text based search mechanisms to locate sign language video content on the web. Current video search engines find videos by matching the query terms with the metadata of the video (e.g. title, tags, and comments). We studied the performance of these search mechanisms for locating SL videos using common query terms on a wide range of topics. Following Shipman et al. [27] we referred to Google Trends 2013, Yahoo top 10 news stories of 2013 and Time top 10 news of 2013 to obtain a set of internet trending topics. We compiled a list of relevant informational topics from these sources and generated 78 query terms, see Table 1. Each of these queries was appended with the term "Sign Language ASL" to locate sign language videos about those popular query topics. The queries were executed on the YouTube search engine, and each video was examined to determine whether it was on the designated topic and in sign language. A maximum of 25 videos were presented by YouTube on the first page. For example, for the "Basketball Sign Language ASL" query in Table 1. YouTube returned 24 video links, but only 15 of them were about Basketball and in Sign Language.

Table 1 Number of on-topic and in-SL videos on the first page of results

Queries	On Topic	In SL	On Topic & In SL	Retrieved
income tax	22	24	22	24
Flu	22	24	21	25
Basketball	16	22	15	24
Iphone 6	15	14	11	24
Bowling	11	23	10	24
Football	10	22	9	24
Iphone 5s	12	15	9	23
French Toast	7	23	6	24
Apple Pie	6	23	6	24
Iphone 5c	8	9	6	19

Table 1 Number of on-topic and in-SL videos on the first page of results (Continued)

Queries	On Topic	In SL	On Topic & In SL	Retrieved
SQL	6	14	6	23
Pizza Dough	6	11	5	14
Playstation 4	9	10	5	24
tax returns	5	10	5	10
JAVA	5	15	5	24
Diarrhea	4	9	4	13
Golf	4	24	4	24
Ipad Mini	4	16	4	22
Hunting	3	21	3	24
Lasagna	4	13	3	24
obamacare	4	22	3	24
knit	3	20	3	24
Mortgage	3	21	3	24
Credit card	3	22	3	24
Diet	5	18	2	24
Hockey	2	22	2	24
Tennis	2	24	2	24
CrossFit	2	22	2	24
Chili	2	24	2	24
Meatball	3	5	2	9
Ipad Air	4	9	2	17
tie a tie	3	6	2	10
blog	2	22	2	24
Diabetes Symptoms	2	5	2	7
student loan	3	17	2	24
Allergies	2	23	1	24
Running	1	21	1	24
Guacamole	1	22	1	24
Hummus	1	23	1	24
Gluten	1	16	1	24
Bitcoin	1	10	1	22
Samsung Galaxy S4	1	17	1	23
car loan	1	2	1	4
stock exchange	1	3	1	5
Javascript	1	12	1	24
Cold	0	24	0	24

Table 1 Number of on-topic and in-SL videos on the first page of results (Continued)

Queries	On Topic	In SL	On Topic & In SL	Retrieved
Labor	0	23	0	24
Balance	0	24	0	24
Back Pain	4	16	0	24
Rash	0	13	0	24
Lupus	0	23	0	24
Shooting	0	21	0	24
Sangria	0	7	0	9
ricin	0	0	0	2
DOMA	0	5	0	16
Molly	0	24	0	24
sequestration	0	1	0	3
Lupus	0	23	0	24
Snapchat	0	9	0	24
HTC one	0	8	0	11
Chromecast	0	7	0	13
Nexus 5	0	6	0	12
The Boston Marathon Bombing	0	2	0	2
file	0	23	0	24
passport	0	22	0	24
Pregnancy Symptoms	0	1	0	10
Anxiety Symptoms	0	0	0	1
Thyroid Symptoms	0	1	0	2
Hiv Symptoms	0	4	0	5
Herpes Symptoms	0	0	0	0
home loan	4	6	0	11
stock market	0	14	0	24
HTML	0	23	0	24
CSS	0	22	0	24
python	0	21	0	24
.net	0	13	0	24
C	0	23	0	24
PHP	0	20	0	24
R	0	22	0	23
Total	***241***	***1201***	***203***	***1518***

Overall, 1,518 videos were returned by YouTube on the first page for 78 queries, with an average of 19.5 videos for each query and an average of 15.4 videos being in sign language. Results are summarized in Table 2 in terms of true positives (on topic & sign language videos) and false positives (off-topic, not in sign language or both). Only 15.8% (241) of the videos were on topic and, of these, 203 (13.4%) were in sign language. In total, 1,201 videos (79.1%) were in sign language. Thus, queries were far more precise for in-SL (79%) than for on-topic (15%). One possible explanation for this result is that we had more terms dealing with the language (ASL and "sign language") than for the topic. In fact, the in-SL precision drops from 79% for topics with one term (e.g., ricin) to 63% for topics with two terms (e.g., car loan).

Table 2 True positives (on topic & sign language videos) and false positives (off-topic, not in sign language or both)

	In Sign Language	**Not in Sign Language**	**Total**
On Topic	203 (13.4%)	38 (2.5%)	241 (15.8%)
Not on Topic	998 (65.7%)	279 (18.4%)	1277 (84.1%)
Total	1201 (79.1%)	317 (20.9%)	1518 (100%)

An analysis of on-topic precision as a function of query length shows an increase from 10% for 1-term topics to 25% for 2-term topics; see Table 3. These results suggest that queries have to be well-balanced between topic and sign language keywords; a higher number of topic terms would result in lesser number of SL videos and more non-SL videos. Such balancing of queries would often be difficult and require trial and error. In addition, the user would have to filter out non-SL videos manually. Thus, there is a need for automatic SL video filtering techniques to eliminate non-SL videos from search results. Towards this goal, this thesis proposes techniques to detect SL

content in videos.

Table 3 Variation in precision with the number of topic terms

Number of topic terms	Precision (SL)	Precision (topic)
1	0.793	0.107
2	0.638	0.251

4. PROPOSED WORK*

In this chapter we describe techniques to detect SL content in user contributed videos. Our techniques rely on polar representation of signer's hand movements with respect to the signer's face. To compute robust face location, we use an ensemble of face detectors based on Haar-like features [2, 3]. We then use the face location information to extract a region of interest, such that it encompasses signer's hand movements. Alongside, we extract foreground hand gestures by subtracting a background model based on adaptive Gaussian mixtures [4]. We combine the region of interest information with the foreground frames and compute the proportion of foreground pixels along the two polar co-ordinates' (angle, distance). We refer to these representations as Polar Motion Profiles (PMPs).

In the following subsections, first we describe our approach to extract polar motion profiles (PMPs) from the foreground movements in a video. Next, we describe three different approaches to generate features from PMPs: computing the average PMP across all frames in a video, modeling the distributions of PMPs using bag-of-words techniques, and supervised dimensionality reduction of PMPs. In all three cases, the resulting information is evaluated on the basis of classification rate with a support vector machine (SVM) classifier.

4.1 Polar Motion Profile (PMP)

The overall signal processing pipeline for computation of PMP is illustrated in Figure 1. In a first step, we process the video with a face-detection algorithm to locate regions of interest (ROI) at

* © 2014 IEEE, Part of this chapter is reprinted with permission from the Karappa, V., Monteiro, C. D., Shipman, F. M., & Gutierrez-Osuna, R. (2014, May). Detection of sign-language content in video through polar motion profiles. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on* (pp. 1290-1294). IEEE.

each frame. In parallel, we generate a background model for each video, from which we identify foreground objects at each frame. At each ROI, we then extract a polar motion profile (PMP) that represents the probability of foreground objects at each polar coordinate. In the last step, we use the PMPs for SL classification. Details on each of these steps are provided on the following subsections.

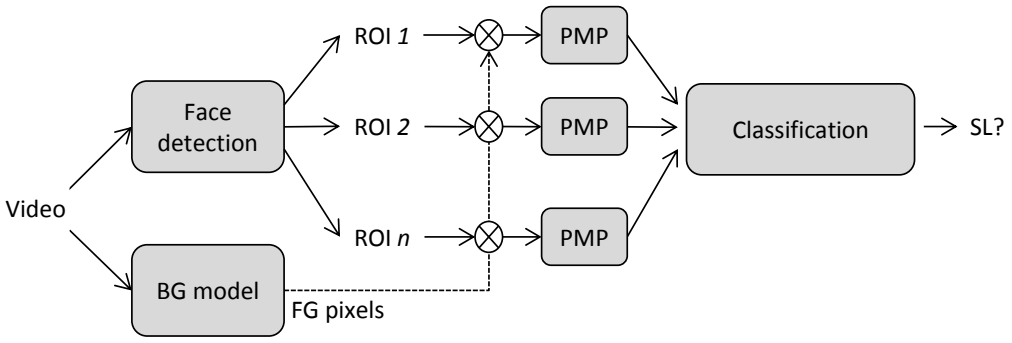

Figure 1 Signal processing pipeline. ROI: region of interest; BG/FG: background/ foreground; PMP: polar motion profile

4.1.1 Face detection

The first step in our pipeline is to use an ensemble face detection technique to find robust locations of faces in the videos. In the pilot study that motivated this work, Monteiro et al. [26] used a single Haar-cascade classifier for face detection [3]. This method worked well with static backgrounds, but does not generalize with videos that contain dynamic backgrounds. Further, the classifier was constrained to searching for a single face and therefore failed when multiple signers were present in a frame. To address these issues, we propose a new algorithm that uses 5 Haar-cascade [28] recognizers in parallel each cascade returning a list of bounding rectangles (one rectangle for each

candidate location for a face).

Figure 2 illustrates the algorithm used to merge information from the multiple Haar-cascade recognizers and remove false positives; each color rectangle represents a distinct classifier. Given a list of n bounding rectangles, we generate C_3^n (n choose 3) sets containing three rectangles each. From these C_3^n sets, we select only those sets that contain rectangles from three distinct classifiers, and refer to them as potential sets. Next, we compute the overlap between the rectangles in these potential sets, measured by computing the distance between top-left and bottom-right corners of the rectangles. We consider two rectangles to be overlapping if and only if these distances are below a threshold of 40 pixels. This threshold was determined empirically. Potential sets containing overlapping rectangles denote a true face location. In a final step, we obtain the true face location by computing the average of the left corner pixel locations of the three rectangles from the potential set; the face size is computed as the average of lengths and breadths of the three rectangles; see Figure 2.

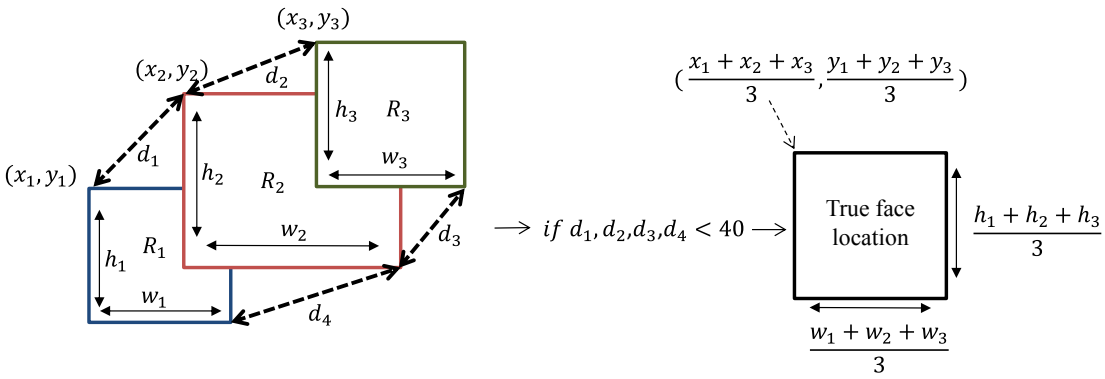

Figure 2 Robust face detection algorithm; R_1, R_2, R_3 are the bounding rectangles returned by three distinct Haar cascades (potential set); h_1, h_2, h_3 are the heights and w_1, w_2, w_3 are widths of these rectangles; (x_1, y_1), (x_2, y_2), (x_3, y_3) are the pixel locations of the top-left corners of these rectangles; d_1, d_4 are the distances between the R_1 and R_2 at their top-left and bottom-right corners; d_2, d_3 are the distances between the R_1 and R_2 at their top-left and bottom-right corners

Figure 3 Figure 4 illustrate face-detection results on a video containing multiple faces; the individual cascades return multiple potential faces, many of which are false positives; using the algorithm as described above eliminates all false positives and returns the location of the three faces in the frame.

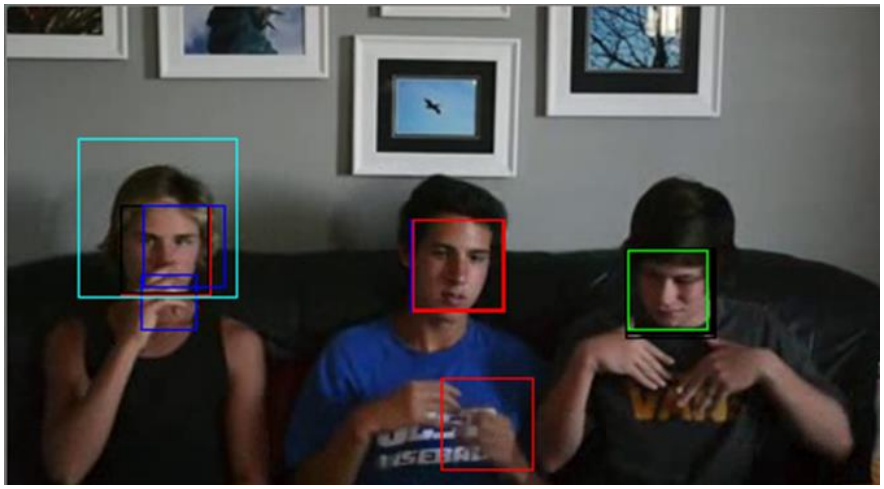

Figure 3 Faces detected by multiple Haar cascades, each denoted by a colored box

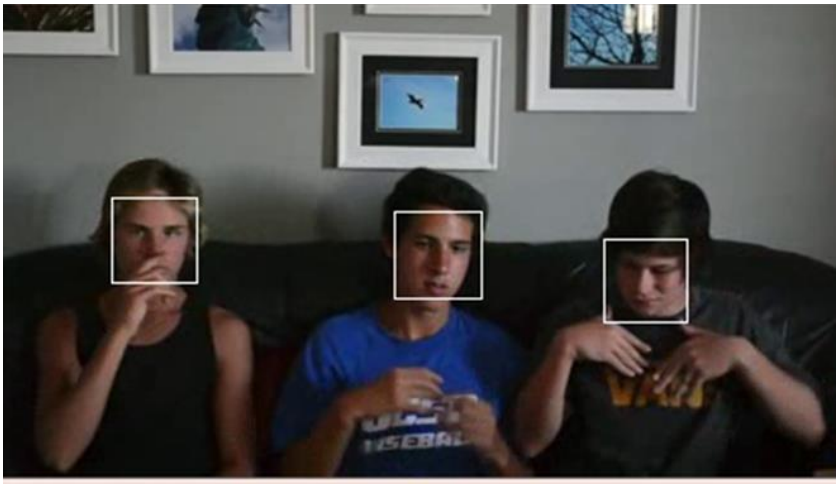

Figure 4 Post-processing of Figure 3

4.1.2 Background modeling

Once (and if) a face has been detected, we perform background subtraction to extract foreground objects in the scene. Following [4], we model the color distribution at each pixel in the video using a separate probability density function per pixel; this is necessary since individual pixels can have vastly different statistics across the video, particularly with non-stationary backgrounds. We build a background model for each pixel with an adaptive Gaussian mixture model (GMM) as proposed by [4]. In this approach, the background model is trained on a set of pixel values $X_T = \{x^{(t)}, \dots, x^{(t-T)}\}$ obtained for a time period T. The background model is denoted by $\hat{p}(\vec{x}/X_T, BG)$, a GMM of maximum M components, see equation (1)

$$\hat{p}(\vec{x}/X_T, BG) = \sum_{m=1}^{M} \hat{\pi}_m N(\vec{x}; \hat{\vec{\mu}}_m, \hat{\sigma}_m^2 I) \quad (1)$$

where $\hat{\pi}_m$, $\hat{\vec{\mu}}_m$, and $\hat{\sigma}_m$ are the mixing weights, estimated means, and estimated variances of the m^{th} Gaussian component. For every new data sample $x^{(t)}$ at time t, these parameters are updated

as shown in the equations (2), (3) and (4).

$$\hat{\pi}_m \leftarrow \hat{\pi}_m + \alpha\left(o_m^{(t)} - \hat{\pi}_m\right) - \alpha c_T \tag{2}$$

$$\hat{\vec{\mu}}_m \leftarrow \hat{\vec{\mu}}_m + o_m^{(t)}(\alpha/\hat{\pi}_m)\,\vec{\delta}_m \tag{3}$$

$$\hat{\sigma}_m^2 \leftarrow \hat{\sigma}_m^2 + o_m^{(t)}(\alpha/\hat{\pi}_m)\,(\vec{\delta}_m^T\vec{\delta}_m - \hat{\sigma}_m^2) \tag{4}$$

where $\alpha = 1/T$ describes an exponentially decaying envelope to limit the influence of old data; αc_T is a negative bias that adjusts the number of components automatically (components with negative weights are dropped); $\vec{\delta}_m$ is the Mahalanobis distance between the data sample and the m^{th} component; and $o_m^{(t)}$ represents the ownership of the t^{th} data sample $\vec{x}^{(t)}$ ($o_m^{(t)}$is set to 1 if it lies within three standard deviations, and otherwise is set to 0).

If the data sample $\vec{x}^{(t)}$ is outside three standard deviations for all the components, a new component is generated with $\hat{\pi}_{M+1} = \alpha, \hat{\vec{\mu}}_{M+1} = \vec{x}^{(t)}, \hat{\sigma}_{M+1} = \sigma_0$, where σ_0 is the initial variance (determined empirically). The component with smallest $\hat{\pi}_m$ is dropped when the maximum number of components M is reached.

Foreground objects appearing in the scene introduce data samples $\vec{\boldsymbol{x}}^{(t)}$ which are not close to any of Gaussian components. New components are generated for these with smaller weights. Thus, the background model can be approximated with the first B components with largest weights as:

$$\hat{p}(\vec{x}/X_T, BG) \sim \sum_{m=1}^{B} \hat{\pi}_m N(\vec{x}; \hat{\vec{\mu}}_m, \hat{\sigma}_m^2 I) \tag{5}$$

The components can be included in the background only when the sum of their weights is greater than a certain tunable threshold. If the weights are sorted by descending order, the number of largest components to be included in the background is given by:

$$B = arg\min_{b}(\sum_{m=1}^{b} \hat{\pi}_m > (1 - c_f)) \quad (6)$$

where c_f is a measure of the amount of data that should be included in the foreground. In our case, c_f was tuned empirically to optimize the detection of hand movements.

Figure 5 shows segmentation results obtained by applying the adaptive GMM background subtraction method. As a final step, we apply morphological erosion and dilation to remove small foreground objects; results are shown in Figure 6. This distribution of foreground pixels (on a frame by frame basis) is then used to generate polar motion profiles, as described in the following section.

Figure 5 Raw frame showing face detected locations

Figure 6 Foreground (FG) pixels returned by the background model.

The tunable parameters in this method were optimized empirically by visually inspecting the foreground separation quality. The final values used for the remainder of this work were:

- Background ratio $T_b = (1 - c_f)$, defines whether the component is significant enough to be included in the background model. Its value was set to 0.9.
- Threshold for the squared Mahalanobis distance, determines when a sample is close to the existing components: if the sample is not close to any component, a new component is generated. A smaller threshold value generates more components, whereas a higher threshold value may result in a small number of components that can grow too large. The value of this parameter was set to three times the standard deviation for the component in consideration.
- Initial variance for the newly generated components affects the speed of adaptation. A value of 15 was experimentally found to be reasonable after observing the foreground quality by varying the parameter from 10 to 100.

4.1.3 Extraction of Polar Motion Profiles

As a final step, we combine results from the face-detection and background-segmentation algorithms to extract a representation of foreground (moving) objects around each face. For every face detected on a frame, we define a region of interest (ROI) large enough* to span the range of hand motions in SLs; see Figure 7.

Once ROIs have been defined for each frame, we generate a polar motion profile (PMP) for each. The PMP is a translation-and-scale-invariant measure of the amount of signing activity computed on a polar coordinate system centered on each face and scaled to the dimensions of each face; see Figure 8. For each ROI, it is computed as the ratio of foreground to total number of pixels at each polar co-ordinate (ρ, θ):

$$PMP_i(\theta, t) = FG_i(\theta, t)/(FG_i(\theta, t) + BG_i(\theta, t)) \quad (7)$$

where $FG_i(\theta, t)$ denotes the number of foreground pixels at angular position θ for the i-th ROI of frame t, and $BG_i(\theta, t)$ is the corresponding number of background pixels. Figure 8 illustrates this process visually.

* The face-detection module returns a bounding box of size $H \times W$. From this, we define an ROI to cover $1H$ above the face center, $3H$ below the face center, $2W$ to the right of the face center, and $2W$ to the left.

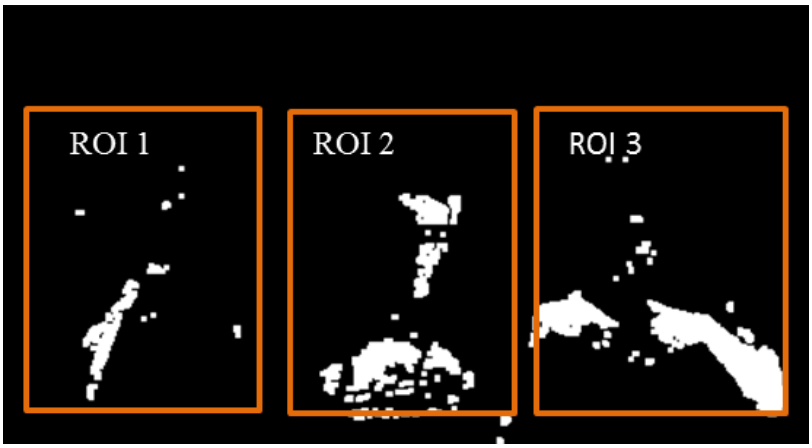

Figure 7 ROIs defined for each face detected in the frame

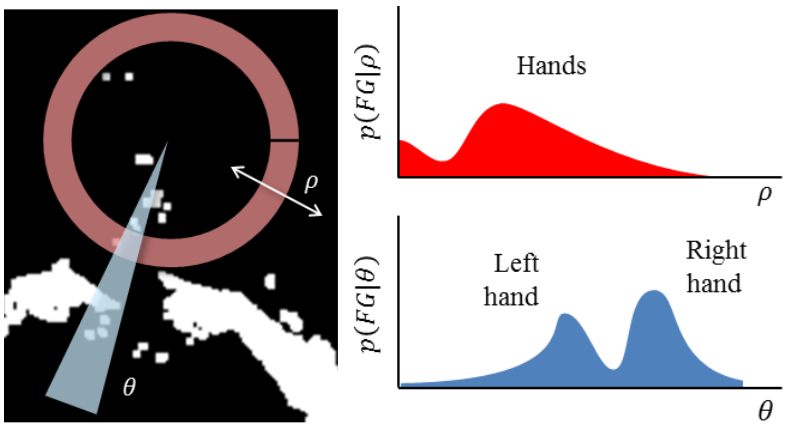

Figure 8 Computation of PMPs for a video frame.

4.2 Classifications

Once PMPs have been extracted from each video, they are passed to a classification module to determine whether or not they have sign language content. We evaluate three approaches to leverage information in the PMPs to discriminate SL videos, as described in the following subsections.

4.2.1 Classification based on average PMPs

Illustrated in Figure 9, the first classification method extracts the average of PMPs extracted from the ROIs of the video. In the case of the angular cordinate θ, the corresponding average PMP is computed as:

$$PMP(\theta) = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{R(t)}\sum_{r=1}^{R(t)}PMP_r(\theta, t) \quad (8)$$

where $R(t)$ is the number of ROIs at frame t and T is the number of frames in the video. The same process is used to derive a PMP for the radial coordinate ρ). Next, we reduce the dimensionality of the average PMPs down to 5 dimensions using PCA; the resulting features then used for training an SVM classifier.

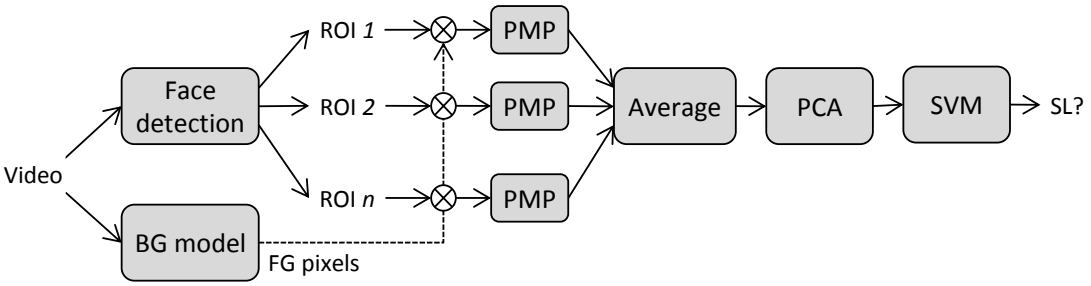

Figure 9 The overall signal processing pipeline for the average PMP classifier

4.2.2 Classification based on Bag-of-PMPs

The bag-of-words model is broadly used in information retrieval tasks. In this model, a text document is represented as a vector containing the frequency of words occurring in that document.

This representation is commonly used for text document classification; methods such as probabilistic latent semantic analysis [29] and latent dirichlet allocation [30] use this representation to extract coherent topics within a document collection. This representation has also been used in computer vision for problems as varied as learning natural scene categories [31], discovering object categories in image collections [32] and scene classification [33]. In these cases, the model is generally referred to as bag-of-visual-words.

In our approach, PMPs can be used as visual words for the discrimination of SL videos. The approach is illustrated in Figure 10. First, we generate a vocabulary of PMPs (codewords) by applying k-means clustering to a collection of SL and non-SL videos. For a new test video, we then assign each of its PMPs into the closest cluster and count the number of occurrences of each cluster. This results in a histogram of visual-word counts for each video (a bag-of-PMPs), which can then be used for classification purposes.

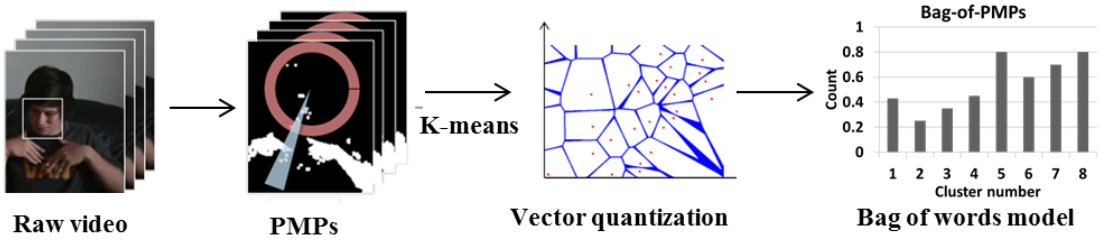

Figure 10 Improving SL detection by analyzing the distribution of PMPs.

We consider two approaches to derive a bag-of-PMPs for each video. Illustrated in Figure 11, the first approach applies k-means clustering directly on the original PMP space; we refer to this approach as bag-of-PMPs. In contrast, the second approach applies PCA to the distribution of

PMPs before applying k-means clustering; see Figure 12. Performing PCA prior to k-means clustering serves two purposes; first, it reduces the dimensionality of the PMP vector (down to 5 dimensions in our implementation), reducing computational costs at test time; second, it de-correlates the PMP dimensions, which otherwise may cause problems since k-means clustering assumes that the features are orthogonal (i.e., k-means uses the Euclidean distance). We refer to this second approach as bag-of-PPMPs.

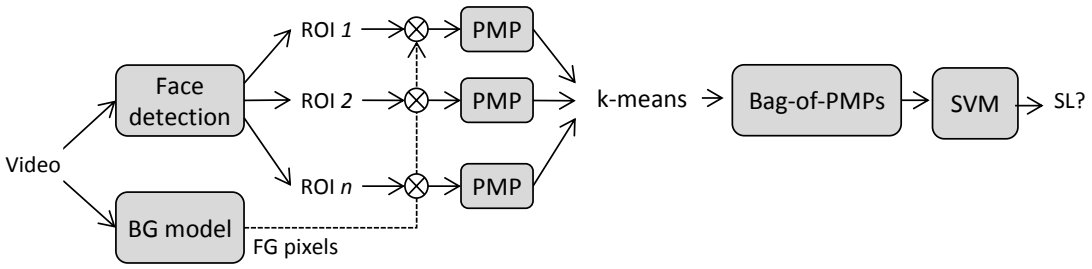

Figure 11 Overall signal processing pipeline for the Bag-of-PMPs method.

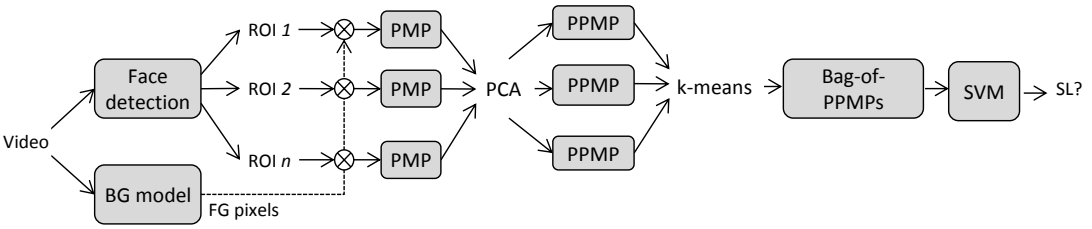

Figure 12 Overall signal processing pipeline for the Bag-of-PPMPs method

4.2.3 Supervised projection of PMPs

The third approach consists of using supervised techniques to find a projection of PMP information that maximizes the discrimination between the two classes. As before, we consider two variants of this strategy. Illustrated in Figure 13, the first variant consists of computing the Linear Discriminants Analysis on a collection of PMPs from SL and non-SL videos; this compresses the PMP information to a single dimension, which we denote by LPMP. In a second step, we perform k-means clustering to the LPMP distribution; hence we refer to this method as bag-of LPMPs.

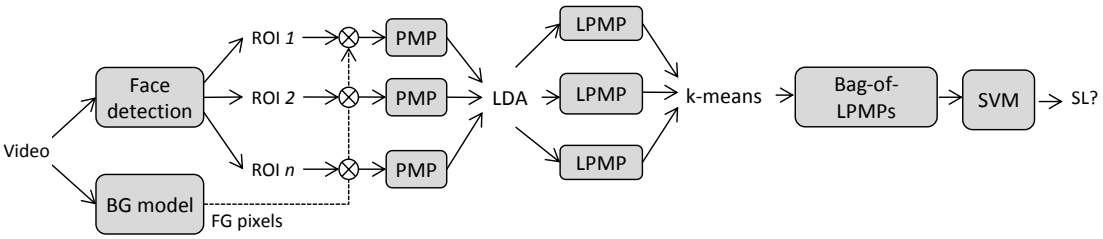

Figure 13 Overall signal processing pipeline for the Bag-of-LPMPs method

The second approach also applies LDA to the distribution of PMPs but instead extracts a number of robust statistics from the resulting 1D distribution. Computing the statistics serves two purposes; first, it handles the issues of outliers which affect the distributions obtained by k-means (centroids are affected by the outliers); second it avoids the binning problem (i.e. deciding the number of bins (k)). We refer to this method as LPMP-STATS. This approach is illustrated in Figure 14.

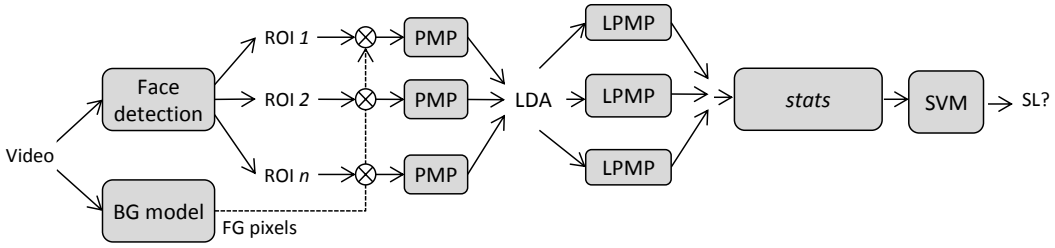

Figure 14 Overall signal processing pipeline for the LPMP-STATS classifier.

We compute the following statistical measures of the LPMPs

- Trimmed mean. The trimmed mean is a robust measure of the central tendency, in contrast with the sample mean, which may be affected by a small fraction of anomalous measurements with abnormally large deviation from the center. A trimmed mean is stated as a mean trimmed by X%, where X is the sum of the percentage of observations removed from both the upper and the lower bounds. We compute the 25% trimmed mean (TM) of the LPMPs and use it as one of the features for training our SL classifier. This mean is computed by excluding the 25% largest and 25% smallest values.
- Interquartile range. The interquartile range (IQR) is a measure of the statistical dispersion, being equal to the upper and lower quartiles. It represents the central portion of the distribution, from the 25^{th} percentile to the 75^{th} percentile,

$$IQR(LPMPs) = Q_3 - Q_1 \tag{9}$$

where Q_i is the i^{th} quartile of the LPMPs; see Figure 15.

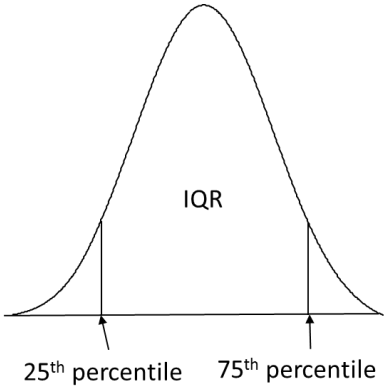

Figure 15 Interquartile range for a normal distribution curve

- Skewness. The skewness is a measure of the asymmetry of the data around the sample mean. If skewness is negative, the data are spread out more to the left of the mean than to the right. If skewness is positive, the data are spread out more towards the right. Any perfectly symmetrical distribution has a skewness of zero; see Figure 16. The skewness of a distribution is given by:

$$SK_1 = \frac{E(x-\mu)^3}{\sigma^3} \tag{10}$$

where μ is the mean of x, σ is the standard deviation of x, and E is the expected value. However, the skewness measure in equation (10) is very sensitive to outliers due to the fact that it uses the third power relative to the sample mean. To avoid this issue, we use a robust measure of skewness [34] based on quartiles:

$$SK_2(LPMPs) = Q_3 + Q_1 - 2Q_2 \,/ Q_3 - Q_1 \tag{11}$$

where SK_2 is the skewnness and Q_i is the i^{th} quartile of the LPMPs vector.

Figure 16 Positive and negative skewness

– Kurtosis. Kurtosis is a measure of how outlier-prone a distribution is. The kurtosis of the normal distribution is 3, which is the standard kurtosis coefficient. Distributions that are more outlier-prone than the normal distribution have kurtosis greater than 3 (positive kurtosis). Distributions that are less outlier-prone have kurtosis less than 3 (negative kurtosis), see Figure 17. The kurtosis of a distribution is:

$$KR_1 = \frac{E(x-\mu)^4}{\sigma^4} \tag{12}$$

However, this expression uses the sample mean, which is prone to outliers. To avoid outlier effects, we use a robust measure of kurtosis [35]:

$$KR_2(LPMPs) = \frac{(E_7 - E_5) + (E_3 - E_1)}{E_6 - E_2} \tag{13}$$

where KR_2 is the robust kurtosis measure, and E_i is the i^{th} octile, $i/8^{th}$ percentile of the data. The Moors coefficient of kurtosis for the normal distribution can be calculated to be 1.23 [36]. Thus, distributions that are more outlier-prone have kurtosis greater than 1.23, whereas distributions less outlier-prone have kurtosis less than 1.23.

SL videos tend to exhibit consistent pattern of hand movements as opposed to erratic movements in the non-SL videos. Thus, we expect most SL videos to have kurtosis values at around 1.23 and values of most non-SL videos to be greater than 1.23.

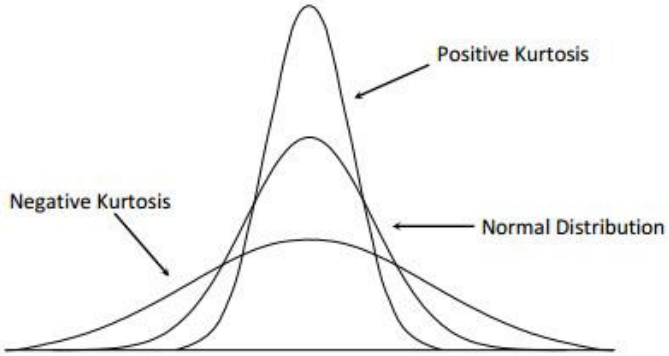

Figure 17 Positive kurtosis (kurtosis greater than 1.23) and negative kurtosis (kurtosis less than 1.23)

5. RESULTS*

We validated the SL classification methods on two sign-language video datasets specifically collected for this purpose. The first dataset, which we will refer to as **Dataset A** [26] was designed to match to the assumptions of static backgrounds and single signers of the pilot study [26]. The dataset contained 192 videos, including 98 SL videos and 94 non-SL videos. The majority of the non-SL videos had been selected by browsing for likely false-positives based on visual analysis (e.g. the whole video consisted of a gesturing presenter, weather forecaster, or other person moving their hands and arms.) This dataset was used to compare our SL classification against the previous method [26] under ideal conditions for the latter.

The second dataset, which we refer to as **Dataset B**, relaxed the assumptions of the pilot study. These videos were selected by performing the text query "*American Sign Language*" using YouTube's search function. We manually labeled as SL/non-SL the top 105 results returned by the search; the majority of these videos did actually contain SL, with only a few false positives (5%). To obtain a set of non-SL videos, we considered related video recommendations for the top 105 results from the search. Again, we manually labeled these related videos and selected 100 videos which did not contain SL. A majority of the videos in dataset B consisted of complex backgrounds, titles and captions appearing intermittently, and multiple signers.

The remaining sections of this chapter are organized as follows. First, we describe the original method proposed by Monteiro et al. [26], which serves as a reference for the work presented in this thesis. Next, we visualize the average polar motion profiles of SL and non-SL videos from

---

both the datasets and discuss their differences. Similarly, we visualize bag-of-PMPs and statistical measures of LPMPs in the following subsections

5.1 Five feature classifier (5FC) of Monteiro et al. [26]

Our work grows out of the pilot study Monteiro et al. [26], which presented a video classifier to detect SL content in videos. The authors used a single Haar-cascade for face detection, and a low-pass filter for background modeling:

$$BG(t) = (1 - \alpha)\, BG(t-1) + \alpha x(t) \tag{14}$$

where $x(t)$ is the grayscale value of the pixel at time t, and $\alpha = 0.04$. The method then extracted five video features (VF1-VF5) for each video. To measure the quantity of movement, two features were computed: (VF1) the total number of pixels computed as foreground for the given video averaged across frames and (VF2) the percentage of pixels that are included in the foreground model for at least one frame. Next, to measure the continuity of motion, (VF3) was computed as the average difference between the final foreground in one frame and the previous frame. Further, two more features were computed to measure the location of the hand motions: (VF4) the symmetry of motion as the average number of foreground pixels that are in symmetric position relative to the center of the signer's face, and (VF5) was computed as the percentage of frames with non-facial movement. When tested on Dataset A (see above), the authors showed that the symmetry of motion with respect to the face (VF4) was more accurate in classification than the other four features combined.

5.2 Average polar motion profiles

Figure 18 and Figure 19 illustrate the average polar motion profiles for SL and non-SL videos on both datasets. The angular profile $PMP(\theta)$ for SL videos shows a high proportion of foreground

pixels (i.e., moving objects) at angles near $\theta = 270°$, which correspond to hand positions directly below the signer's face. In contrast, non-SL videos show activity not only at $\theta = 270°$ but also at angles near $\theta = 90°$, which correspond to hand positions directly above the face. These results are consistent for both datasets (A, and B) which points to the generality of the angular profiles as a measure of discrimination between SL and non-SL videos.

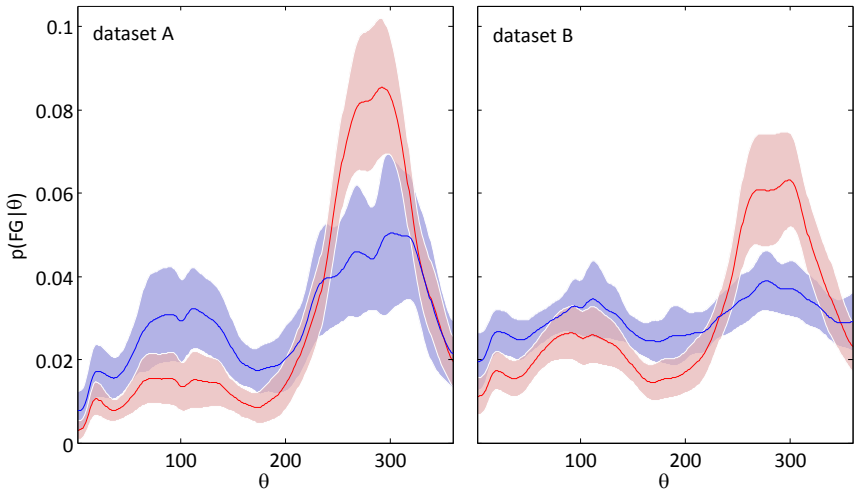

Figure 18 Angular polar motion profiles of both the datasets

The radial profile $PMP(\rho)$ for SL videos on Database A show a high proportion of foreground pixels at a broad range of distances ranging from 20% to 80% of the maximum distance, relative to the size of the ROI*. In contrast, the distribution of foreground pixels for non-SL on database A peaks at around 30% of the maximum distance, and remains rather constant at larger distances. On dataset B, however, the radial profiles for SL and non-SL videos are very similar, which suggest that radial profiles may not be a reliable measure of discrimination between SL and non-SL videos.

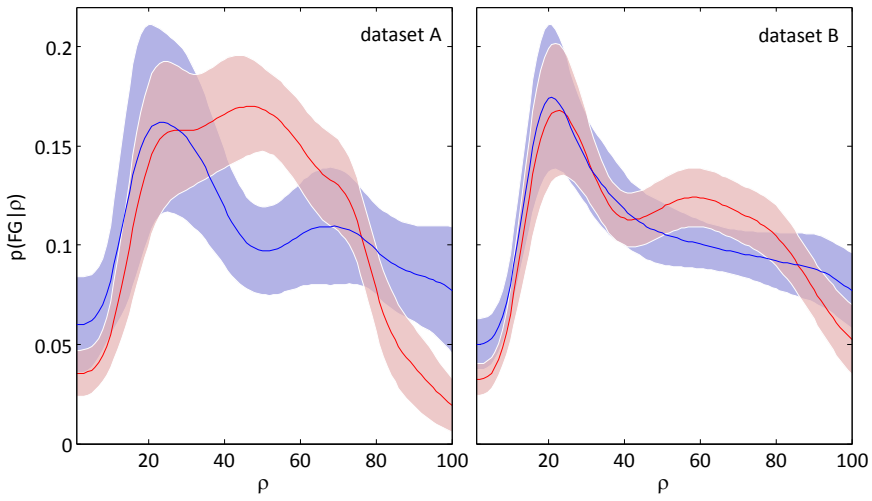

Figure 19 Radial polar motion profiles for both datasets

One may be tempted to question the relevance of symmetry in hand movements as a discriminating feature for SL – a main finding of the study [26]. However, close inspection of Figure 18 reveal a strong vertical symmetry in the angular profiles, with higher probability of foreground motions around $\theta = 270°$ (directly below the signer's face). In contrast, the radial profiles $PMP(\rho)$ appear to be less reproducible across datasets, see Figure 19.

5.3 Bag-of-words model

Figure 20 shows the distribution of PMPs (bag-of-words model) over all the videos in dataset A for cluster size k=8. The distributions computed in the original PMP space (bag-of-PMPs) are shown in the Figure 20(a). In these distributions, similar bin proportions for SL and non-SL videos can be attributed to high dimensional and correlated features in the original PMP space. This results in low degree of class separability when using the bag-of-PMPs representation. However,

when we compute the distributions of PMPs in the PCA subspace (bag-of-PPMPs), higher differences in the bin proportions for SL and non-SL videos can be obtained, see Figure 20(b). Projecting the PMPs on the PCA subspace helps de-correlate the features and a fewer set of dimensions can be used for computing the distributions. Similarly, higher differences in bin proportions are also obtained for distributions of PMPs in the LDA subspace, which leverages the class information to obtain high separability between SL and non-SL, see Figure 20(c).

For the bag-of-LPMPs representation (Figure 20 (c)), LPMPs of the non-SL videos are concentrated at bin 6, whereas for the SL videos the LPMPs are spread across the last three bins (6, 7 and 8). This difference in the distributions can be attributed to the higher degree of hand movements due to signing in SL videos compared to non-SL videos. A large number of foreground ROIs for non-SL videos contain minimal hand movements, which results in similar PMP vectors. Thus, most of the non-SL ROIs are assigned into a single bin. In contrast, SL foreground ROIs show relatively higher degree of hand movements, resulting in higher differences in PMP vectors.

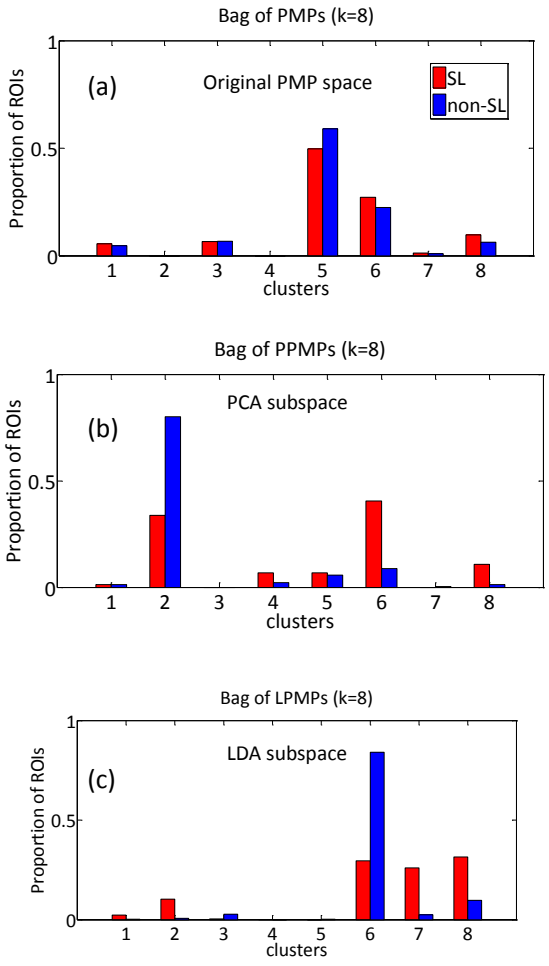

Figure 20 Individual bin (cluster) proportions for (a) bag-of-PMPs, (b) bag-of-PPMPs and (c) bag-of-LPMPs , of the videos in dataset A

To illustrate this, we show some sample foreground ROIs assigned to the bin numbers 2, 6, 7 and 8 for SL videos in Figure 21. We can see that different hand gestures and motions are spread out across these clusters. However, for non-SL videos, most of the foreground ROIs contain minimal activity and as a result higher percentage of non-SL ROIs are assigned to the 6th bin. Figure 22 shows such sample foreground ROIs from non-SL videos assigned to bin 6 and ROIs containing rare hand movements are assigned to the bin 8.

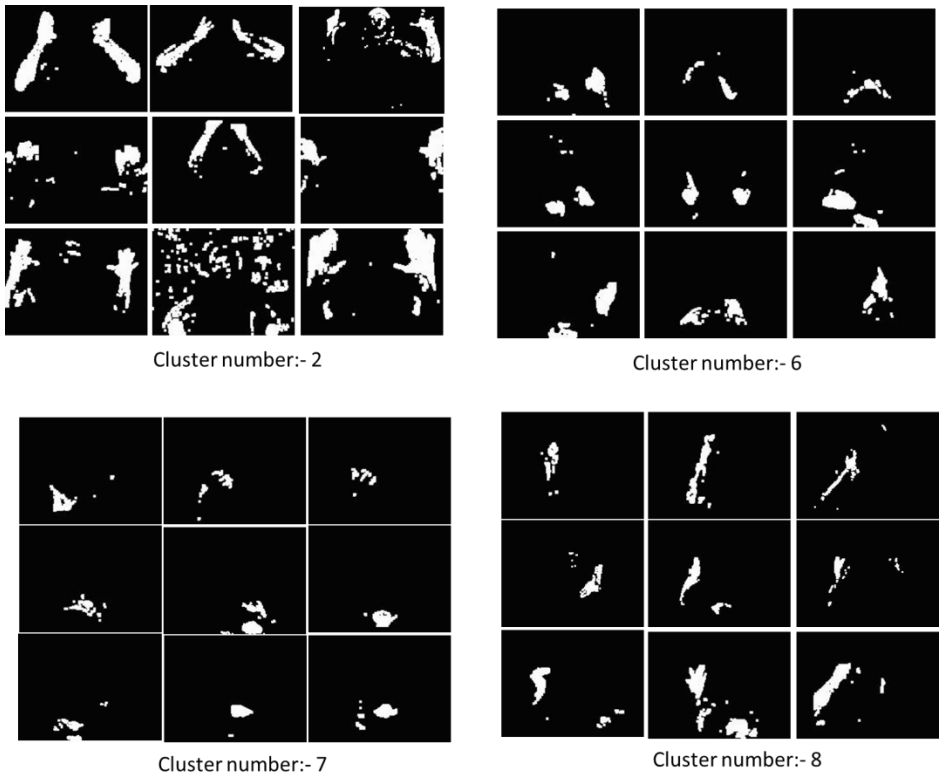

Figure 21 Clustered SL frames of clusters 2, 6, 7 and 8 for the bag-of-LPMPs, see Figure 20(b)

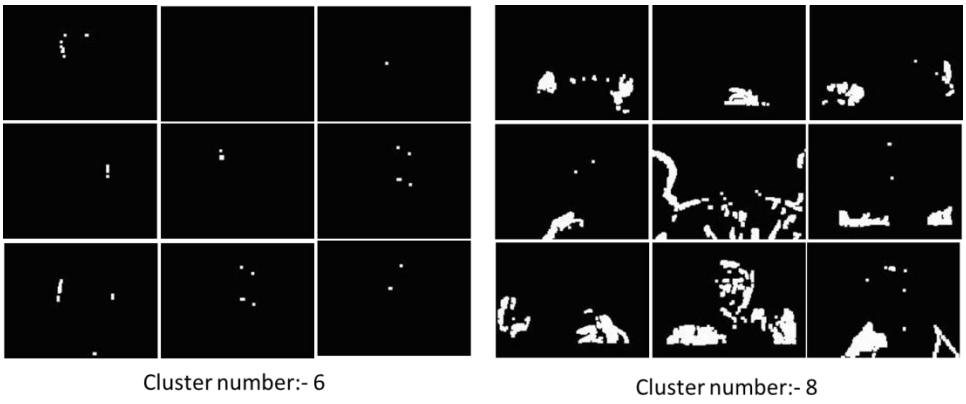

Figure 22 Shows non-SL frames assigned to the 6th and 8th clusters of the bag-of-LPMPs shown in Figure 20(b)

5.4 LPMP-Stats

As described in the section 4.2.3, we compute statistical measures of the LPMPs for the SL and non-SL videos and use these measures for classification. Figure 23 shows the plots of trimmed mean vs IQR and skewness vs kurtosis of these measures for the videos in the dataset A. We can see that high separability can be obtained between SL and non-SL classes using these statistical measures.

Most of the SL videos have IQR values higher than 0.1 whereas non-SL videos have IQR values below 0.1. Trimmed mean values also show a similar trend which can be attributed to the larger amount of activity in the ROIs of SL videos as compared to the non-SL ones. However, some non-SL videos contain a lot of foreground activity due to abrupt camera movements. As a result, such non-SL videos have trimmed mean and IQR values similar to the SL videos.

The distributions of LPMPs for SL videos are more symmetric (skewness ~ 0) as compared to non-SL videos. Further, the kurtosis values show that distributions of LPMPs for SL videos are less outlier prone as compared to non-SL videos.The trends captured by these measures are consistent across both the datasets. This shows that SL videos exhibit a certain pattern of activity which can be captured by these statistical measures and used for classification.

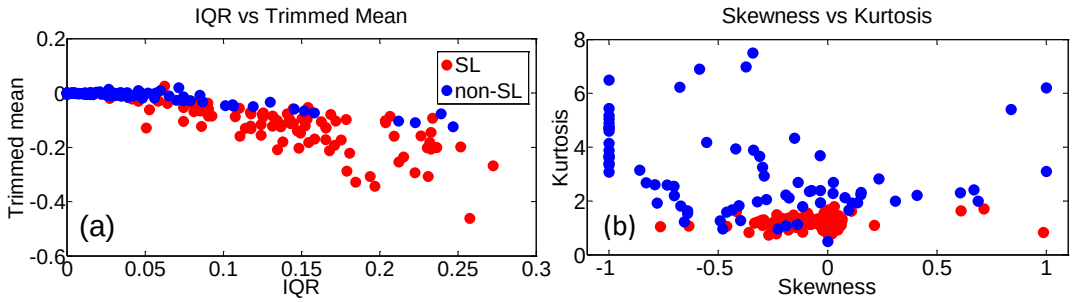

Figure 23 Trimmed mean vs IQR and Skew-ness vs Kurtosis for videos in the dataset A.

5.5 Retrieval results

5.5.1 Average PMP classifier (PMP)

We compared the average PMP classifier against the one in the pilot study [26], which we will refer to as 5FC (five feature classifier). Classification results on dataset A are shown in Table 4. We compared both classifiers as a function of the training set size (15, 30, 45 and 60 videos per class) in terms of precision, recall and F1 score (harmonic mean of precision and recall). Both methods perform comparably, with a slight advantage for 5FC in terms of precision and a slight advantage for PMP in terms of recall. Comparison of the F1 scores also shows a small advantage towards PMP.

We used the 5FC and average PMP classifiers trained on dataset A to generate class labels for the videos in dataset B, a more challenging test since both datasets had been constructed for different purposes. Table 5 summarizes the classification results on dataset B. 5FC and average PMP classifiers achieve similar precision rates as in dataset B. However, while average PMP is able to maintain the high recall rate obtained on dataset B, recall degrades dramatically for 5FC.

5.5.2 Bag-of-PMPs classifier

Next, we evaluate performance of an SVM classifier trained using the bag-of-PMPs as the features. The precision obtained by this classifier is 50% and recall rates in the range of (85-90%) for all the training video sizes; see Table 4. This classifier returns a large number of false positives. Thus, F1 scores are in the range of (63%-67%) relatively below the performance of all other methods. It shows similar performance on the dataset B.

Table 4 Classification results on dataset A

#vids/class	*Average*		*PMP distributions (Unsupervised)*		*PMP distributions (Supervised)*	
	Precision					
	5FC	PMP	Bag-of-PMPs	Bag-of-PPMPs	Bag-of-LPMPs	LPMP-STATS
15	0.82	0.78	0.5	0.82	0.85	**0.88**
30	0.84	0.81	0.5	0.83	0.84	**0.88**
45	0.81	0.82	0.5	0.84	0.83	**0.89**
60	0.82	0.82	0.52	0.86	0.83	**0.88**
#vid /class	**Recall**					
	5FC	PMP	Bag-of-PMPs	Bag-of-PPMPs	Bag-of-LPMPs	LPMP-STATS
15	0.86	0.90	0.85	0.84	0.87	0.92
30	0.88	0.92	0.88	0.87	**0.96**	0.92
45	0.91	0.93	0.9	**0.94**	0.9	0.93
60	0.91	0.93	**0.94**	0.92	**0.94**	0.93
#vids/class	**F1 Score**					
	5FC	PMP	Bag-of-PMPs	Bag-of-PPMPs	Bag-of-LPMPs	LPMP-STATS
15	0.84	0.83	0.63	0.83	0.86	**0.9**
30	0.85	0.86	0.64	0.85	0.89	**0.9**
45	0.85	0.87	0.64	0.89	0.86	**0.91**
60	0.86	0.87	0.67	0.89	0.88	**0.9**

Table 5 Classification results on dataset B

Classifier	Precision	Recall	F1 score
5FC	0.82	0.60	0.69
PMP	0.81	0.94	0.87
Bag-of-PMPs	0.55	0.76	0.64
Bag-of-PPMPs	0.72	0.70	0.71
Bag-of-LPMPs	0.74	0.78	0.76
LPMP-STATS	**0.84**	**0.94**	**0.88**

5.5.3 Bag-of-PPMPs classifier

Further, we use distribution of PMPs in PCA subspace (bag-of-PPMPs) for training another SVM classifier and evaluate its performance, see Table 4. The bag-of-PPMPs classifier achieves higher precision (82%-86%) than 5FC, PMP and bag-of-PMPs classifiers and shows comparable recall rates. Further, we used the bag-of-PPMPs classifier trained on dataset A to generate labels for dataset B. The results are shown in the Table 5. bag-of-PPMPs achieved higher F1 score than 5FC but lesser than average PMP classifier.

5.5.4 Bag-of-LPMPs classifier

Classification results of Bag-of-LPMPs classifier on dataset A are shown in Table 4. Bag-of-LPMPs classifier showed higher precision than the methods 5FC and average PMP for dataset A. The recall for Bag-of-LPMPs is comparable to the average PMP method. It rises to 96% and then falls to 90% with increase in the training set size. However, the recall for the average PMP consistently rises with increase in the training set size. Similarly, the F1 score of the Bag-of-LPMPs drops by 3% to 86% from 89% for a training set size of 45 videos. However, other methods show consistent improvement in the F1 scores with the increase in the size of the training set size. When tested on dataset B, Bag-of-LPMPs classifier showed higher precision and recall than the Bag-of-PPMPs classifier but lower than average PMP and 5FC, see Table 5.

5.5.5 LPMP-STATS classifier

To address the issue of outliers, we used robust statistical measures (section 4.2.3) of LPMPs as features for classification. The performance of the SVM classifier trained using these features is shown in Table 4. This classifier outperforms other methods consistently over different training sizes. We can observe a similar trend when tested on dataset B. It achieved higher precision and

recall than all other methods on dataset B, see Table 5.

6. CONCLUSION

6.1 Discussion

This thesis studies the feasibility of developing detection mechanisms for sign language videos. Issues in current text-based video retrieval systems motivated us to develop image and video processing techniques to locate sign language videos on the web. [26] presented a technique of detecting sign language videos on a dataset of videos containing single signers, and simpler backgrounds. This thesis extends beyond that work by considering videos containing multiple signers and complex backgrounds.

Our approach uses an ensemble face detection mechanism and an adaptive background model to compute a distribution of foreground movements in polar coordinates –the polar motion profiles (PMP) for each ROI in a video. We presented three different representations of videos by processing PMPs- as average across ROIs, bag-of-PMPs and statistical measures of LDA scores of PMPs. We trained classifiers using these representations and evaluated their performance on the original dataset in [26] (dataset A), and a new dataset collected from YouTube (dataset B).

The performance of the average PMP method on dataset A is comparable to Monteiro's 5FC classifier, as shown in Figure 24(a). However, it performs significantly better on dataset B. This can be attributed to the fact that the video pre-processing in 5FC was not designed to handle videos containing multiple signers and complex backgrounds. In contrast, in the average PMP method we make provisions to handle such cases by using an ensemble face detection technique and an adaptive background model.

Further, Bag-of-PMPs computed in the original space show relatively lower performance when compared to the Bag-of-PPMPs (PCA subspace); see Figure 24(b). The Bag-of-PMPs perform poorly across both the datasets, with precision in the range of 50%. This can be attributed to the

sensitivity of k-means towards high dimensional and correlated PMP feature space. This results in similar bin proportions of PMPs in SL and non-SL distributions and, as a result, lower separability between classes. In contrast, we achieve higher separability if bin proportions are computed in the lower dimensional de-correlated PCA subspace of PMPs. As a result, the Bag-of-PPMPs (distributions in PCA space) showed significant improvement in performance for both the datasets.

Figure 24(c) shows the performance of distributions of PMPs in LDA space (Bag-of-LPMPs) vs LPMP-Stats methods across both datasets. On dataset A, the Bag-of-LPMPs show comparable performance to the LPMP-Stats method. However, the performance of Bag-of-LPMPs degrades on the more complex dataset B. This can be attributed to the sensitivity of k-means towards outliers. In contrast, the LPMP-Stats method uses outlier-robust statistical measures and, as a result, it shows higher performance compared to Bag-of-LPMPs on dataset B.

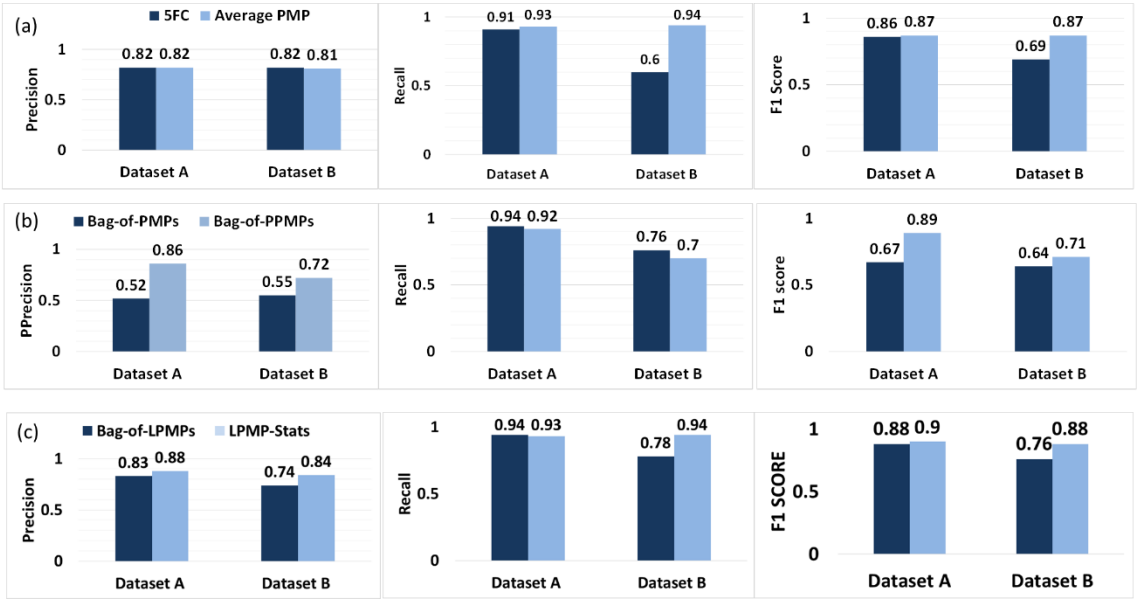

Figure 24 Precision, recall and F1 Scores of (a) 5FC vs Average PMP (b) Bag-of-PMPs vs Bag-of-PPMPs (c) Bag-of-LPMPs vs LPMP-Stats

Precision and recall is high across all the methods on dataset A except for the Bag-of-PMPs, which suffers from curse of dimensionality; see Figure 25. For dataset B, 5FC's recall falls to 60%, while the precision still remains high (80%). The SL videos in the dataset B contained multiple signers and complex backgrounds. 5FC's video processing was not designed to handle such videos. Hence, the number of true positives for 5FC decreases (low recall). Further, the non-SL videos in the dataset B were collected from the related recommendations for SL videos on YouTube with no restriction to contain SL-like movements unlike in dataset A. Thus, 5FC filters out high number of non-SL videos on the dataset B resulting in high precision (due to the 5FC's strength to filter out even the SL-like non-SL videos on dataset A).

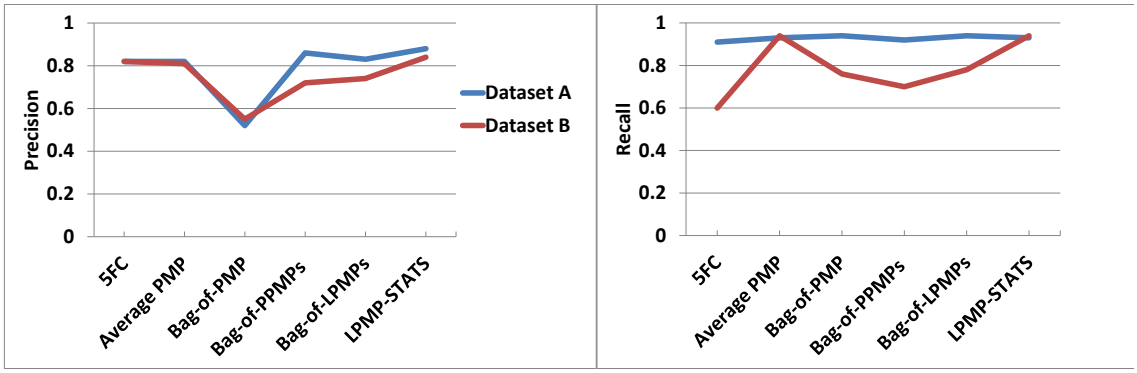

Figure 25 Precision and recall across all the methods on datasets A and B.

When we compare F1 scores across methods, as shown in Figure 26, LPMP-stats stands out as the winner for both datasets. This can be explained by the fact that it uses (1) class information to achieve higher separation between SL and non-SL, and (2) outlier-robust statistical measures. Further, we also believe that the performance of Bag-of-LPMPs and Bag-of-PPMPs can be improved by using clustering techniques that are robust to outliers. For example, k-medoids can

be used instead of k-means to perform vector quantization. This would in turn lead to improved retrieval performance. However, in the case of Bag-of-PMPs (original space), the performance cannot be improved due to the fact that dimensions in the original PMP space are highly correlated.

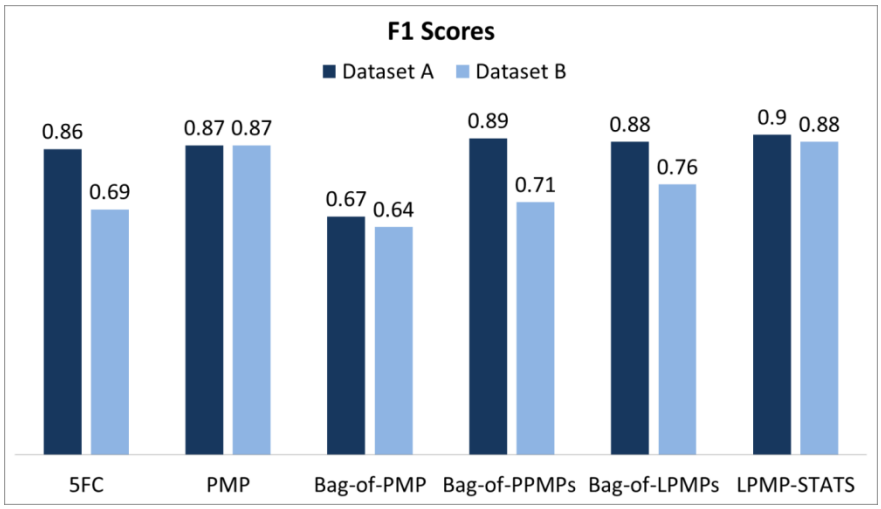

Figure 26 Overall comparison of F1 scores across all the methods

6.2 Future work

Several areas for improvement could be pursued as an extension to this thesis work. Our current methods either generate average motion profiles for each video or distributions of motion profiles. We do not consider the temporal information in the sequence of PMPs. Further improvement in retrieval results could be achieved by modelling the cues present in the sequence of these motion profiles; as an example, an HMM trained on the sequence of principal components or LDA projections may be used to extract dynamic signatures of sign languages and improve SL detection performance.

Further, the background subtraction technique can be coupled with a skin detection module to handle complex videos containing abrupt camera movements. Such movements introduce noise in the foreground data. A skin detection module can be used to remove the noisy foreground. To illustrate this, a signal processing pipeline is shown in the Figure 27. The skin detection module would act as a filter for the noisy foreground pixels, which can then be processed in a manner similar to one of our methods for classification.

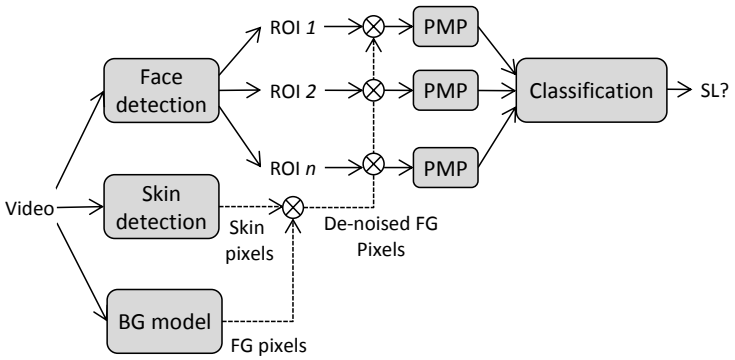

Figure 27 De-noising foreground pixels using skin detection

Additional work is also needed to improve recognition of profile faces – our methods use a single Haar cascade for profile faces. This would be needed for videos where the signers are facing each other and not the camera, while signing, as shown Figure 28(a).

Further, the methods presented in this thesis would not be suitable for videos containing a mix of signers and non-signers, as shown in the Figure 28(b). For example, the overall polar motion profile for average PMP method would be affected by the PMPs of non-signers in the video. Such videos would be tagged non-SL if they contain many non-signers. In such cases, it would become necessary to evaluate PMPs for each person independently to detect SL content. As shown in

Figure 29, a face tracking module would be needed to collect and differentiate ROIs for each person. PMPs for each person would then be evaluated independently by one of the classification approaches presented in this thesis. Next, a video would be tagged as SL, if SL content is detected in at-least one set of PMPs.

Figure 28 (a) Video containing singers and non-signers (b) video containing SL conversations (signers facing each other)

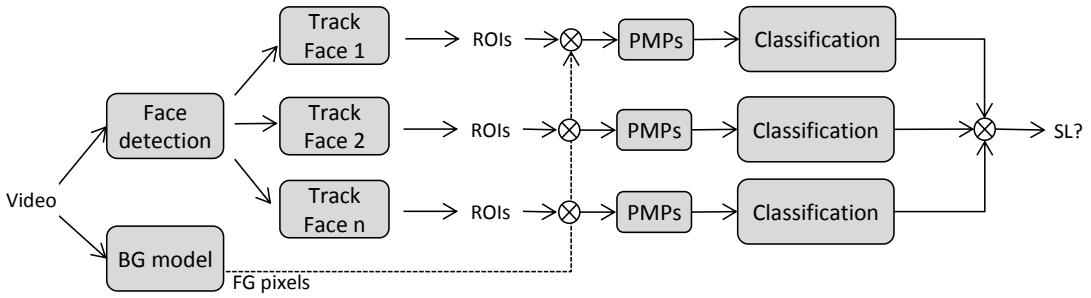

Figure 29 Signal processing pipeline to detect content separately for each person in the video.

Further work would also be required to handle videos where SL content exists only in a subset of the frames. Such videos should be divided up into smaller segments and evaluated separately; see Figure 30. For each segment of the video, face detection and background modelling are executed independently. A classification label is then obtained for each set of PMPs separately. The video is tagged as SL, if at-least one-set of PMPs is classified as SL.

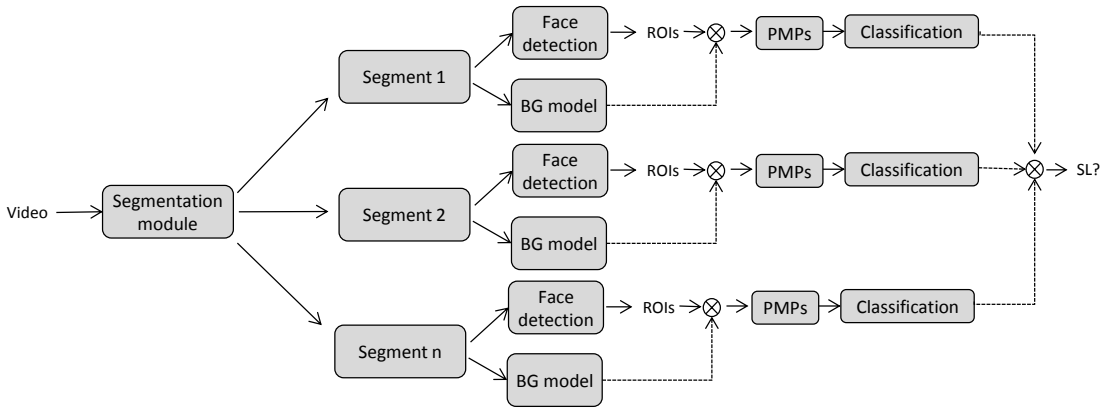

Figure 30 Evaluation of videos containing SL in segments.

Finally, using the techniques presented in this thesis, a sign language video retrieval filter could be implemented. This filter could act as a wrapper over existing search systems, thus, improving the chances of fulfilling information needs for the SL community and enabling the possibility of automatic tagging and annotations.

REFERENCES

1. NIH, *American Sign Langauge.* NIH Publication No. 11-4756, June 2011.

2. Lienhart, R. and J. Maydt. *An extended set of haar-like features for rapid object detection*. in *Image Processing. 2002. Proceedings. 2002 International Conference on*. 2002. IEEE.

3. Viola, P. and M. Jones. *Rapid object detection using a boosted cascade of simple features*. in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*. 2001. IEEE.

4. Zivkovic, Z. *Improved adaptive Gaussian mixture model for background subtraction*. in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*. 2004. IEEE.

5. Starner, T., J. Weaver, and A. Pentland, *Real-time american sign language recognition using desk and wearable computer based video.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, 1998. **20**(12): p. 1371-1375.

6. Vogler, C. and D. Metaxas. *Parallel hidden markov models for american sign language recognition*. in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*. 1999. IEEE.

7. Bowden, R., et al., *A linguistic feature vector for the visual interpretation of sign language*, in *Computer Vision-ECCV 2004*. 2004, Springer. p. 390-401.

8. Holden, E.-J., G. Lee, and R. Owens, *Australian sign language recognition.* Machine Vision and Applications, 2005. **16**(5): p. 312-320.

9. Yang, M.-H., N. Ahuja, and M. Tabb, *Extraction of 2d motion trajectories and its application to hand gesture recognition.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2002. **24**(8): p. 1061-1074.

10. Yang, R., S. Sarkar, and B. Loeding. *Enhanced level building algorithm for the movement epenthesis problem in sign language recognition*. in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. 2007. IEEE.

11. Dimov, D., A. Marinov, and N. Zlateva. *CBIR approach to the recognition of a sign language alphabet*. in *Proceedings of the 2007 international conference on Computer systems and technologies*. 2007. ACM.

12. Potamias, M. and V. Athitsos. *Nearest neighbor search methods for handshape recognition*. in *Proceedings of the 1st international conference on Pervasive Technologies Related to Assistive Environments*. 2008. ACM.

13. Bauer, B. and K.-F. Kraiss. *Video-based sign recognition using self-organizing subunits*. in *Pattern Recognition, 2002. Proceedings. 16th International Conference on Pattern*

Recognition. 2002. IEEE.

14. Nayak, S., S. Sarkar, and B. Loeding. *Unsupervised modeling of signs embedded in continuous sentences*. in *Computer Vision and Pattern Recognition-Workshops, 2005. CVPR Workshops. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2005. IEEE.

15. Neidle, C. and D. MacLaughlin, *Signstream: A tool for linguistic research on signed languages.* Sign language & linguistics, 1998. **1**(1): p. 111-114.

16. Liang, R.-H. and M. Ouhyoung. *A real-time continuous gesture recognition system for sign language*. in *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*. 1998. IEEE.

17. Braffort, A. *ARGo: An architecture for sign language recognition and interpretation*. in *Proceedings of Gesture Workshop on progress in gestural interaction*. 1996. Springer-Verlag.

18. Fang, G., et al., *Signer-independent continuous sign language recognition based on SRN/HMM*, in *Gesture and sign language in human-computer interaction*. 2002, Springer. p. 76-85.

19. Gao, W., et al., *Sign language recognition based on HMM/ANN/DP.* International journal of pattern recognition and artificial intelligence, 2000. **14**(05): p. 587-602.

20. Gao, W., et al. *Transition movement models for large vocabulary continuous sign language recognition*. in *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on Automatic Face and Gesture Recognition*. 2004. IEEE.

21. Somers, G. and R. Whyte. *Hand posture matching for irish sign language interpretation*. in *Proceedings of the 1st international symposium on Information and communication technologies*. 2003. Trinity College Dublin.

22. Erdem, U.M. and S. Sclaroff. *Automatic detection of relevant head gestures in American Sign Language communication*. in *Pattern Recognition, 2002. Proceedings. 16th International Conference on Pattern Recognition*. 2002. IEEE.

23. La Cascia, M. and S. Sclaroff. *Fast, reliable head tracking under varying illumination*. in *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on Computer Vision and Pattern Recognition.* 1999. IEEE.

24. Parashar, A.S., *Representation and interpretation of manual and non-manual information for automated American sign language recognition*, 2003, University of South Florida.

25. Cherniavsky, N., R.E. Ladner, and E.A. Riskin, *Activity detection in conversational sign language video for mobile telecommunication*, in *Proc. 8th IEEE Intl. Conf. on Automatic Face & Gesture Recognition* 2008. p. 1-6.

26. Monteiro, C.D., R. Gutierrez-Osuna, and F.M. Shipman. *Design and evaluation of classifier for identifying sign language videos in video sharing sites*. in *Proceedings of the 14th international ACM SIGACCESS conference on Computers and accessibility*. 2012. ACM.

27. Shipman, F.M., R. Gutierrez-Osuna, and C.D. Monteiro, *Identifying Sign Language Videos in Video Sharing Sites.* ACM Transactions on Accessible Computing (TACCESS), 2014. **5**(4): p. 9.

28. Lienhart, R., A. Kuranov, and V. Pisarevsky, *Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection*, in *Pattern Recognition*, B. Michaelis and G. Krell, Editors. 2003, Springer Berlin Heidelberg. p. 297-304.

29. Hofmann, T. *Probabilistic latent semantic analysis*. in *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*. 1999. Morgan Kaufmann Publishers Inc.

30. Blei, D.M., A.Y. Ng, and M.I. Jordan, *Latent dirichlet allocation.* the Journal of machine Learning research, 2003. **3**: p. 993-1022.

31. Fei-Fei, L. and P. Perona. *A bayesian hierarchical model for learning natural scene categories*. in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2005. IEEE.

32. Sivic, J., et al., *Discovering object categories in image collections.* 2005.

33. Yang, J., et al. *Evaluating bag-of-visual-words representations in scene classification*. in *Proceedings of the international workshop on multimedia information retrieval*. 2007. ACM.

34. Bowley, A.L., *Elements of statistics*. 1920: King.

35. Moors, J., *A quantile alternative for kurtosis.* The statistician, 1988: p. 25-32.

36. Kim, T.-H. and H. White, *On more robust estimation of skewness and kurtosis.* Finance Research Letters, 2004. **1**(1): p. 56-73.