EVALUATION OF ALTERNATIVE FACE DETECTION TECHNIQUES AND VIDEO SEGMENT LENGTHS ON SIGN LANGUAGE DETECTION

A Thesis

by

SATYAKIRAN DUGGINA

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,	Frank Shipman
Co-chair of Committee,	Ricardo Gutierrez-Osuna
Committee Member,	Ergun Akleman
Head of Department,	Dilma Da Silva

December 2015

Major Subject: Computer Science

ABSTRACT

Sign language is the primary medium of communication for people who are hearing impaired. Sign language videos are hard to discover in video sharing sites as the text-based search is based on metadata rather than the content of the videos. The sign language community currently shares content through ad-hoc mechanisms as no library meets their requirements. Low cost or even real-time classification techniques are valuable to create a sign language digital library with its content being updated as new videos are uploaded to YouTube and other video sharing sites.

Prior research was able to detect sign language videos using face detection and background subtraction with recall and precision that is suitable to create a digital library. This approach analyzed one minute of each video being classified. Polar Motion Profiles achieved better recall with videos containing multiple signers but at a significant computational cost as it included five face trackers. This thesis explores techniques to reduce the computation time involved in feature extraction without overly impacting precision and recall deeply.

This thesis explores three optimizations to the above techniques. First, we compared the individual performance of the five face detectors and determined the best performing single face detector. Second, we evaluated the performance detection using Polar Motion Profiles when face detection was performed on sampled frames rather than detecting in every frame. From our results, Polar Motion Profiles performed well even when the information between frames is sacrificed. Finally, we looked at the effect of using shorter video segment lengths for feature extraction. We found that the drop in precision is minor as video segments were made shorter from the initial empirical length of a minute.

Through our work, we found an empirical configuration that can classify videos with close to two orders of magnitude less computation but with precision and recall not too much below the original voting scheme. Our model improves detection time of sign language videos that in turn would help enrich the digital library with fresh content quickly. Future work can be focused on enabling diarization by segmenting the video to find sign language content and non-sign language content with effective background subtraction techniques for shorter videos.

DEDICATION

To my Mom, Dad,

Sister and Brother-in-law,

for their immense support and care

ACKNOWLEDGEMENTS

I express my sincere gratitude to Dr. Frank Shipman, whose vision drove this thesis all the way. His immense support and encouragement through the hardest of my times is unforgettable.

I thank Dr. Ricardo Gutierrez-Osuna, who promptly responded to clarify my doubts. I thank Dr. Ergun Akleman for his support to my research. I thank the Department of Computer Science & Engineering faculty and staff at TAMU who handles the comfort of a student as a priority.

Lastly, I would like to thank my lab mates Caio, Sampath, Suinn, Gabriel, Joshua, Meghanath, Akshay and Saketh for their suggestions and support.

NOMENCLATURE

SL	Sign Language
ASL	American Sign Language
BSL	British Sign Language
Non-SL	Not in Sign Language
ROI	Region of Interest
PMP	Polar Motion Profile
SVM	Support Vector Machine
RBF	Radial Basis Function

TABLE OF CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

Page

1. INTRODUCTION

Sign language is the medium of communication for people who are hearing impaired. The sign language community shares videos through ad-hoc mechanisms as current libraries do not meet their requirements. With the rising popularity of video sharing sites like YouTube and Vimeo, the volume of sign language content available is steadily growing. However, finding the relevant content is hard as information needs fundamentally depend on content whereas the search tools provided by video sharing sites use metadata for the discovery of content. Manual tagging of videos is not an option as numerous videos are being posted online every minute. Automatic detection techniques would enable the enrichment of a sign language digital library with fresh content.

A pilot study to create a digital library by Monteiro et al. [3] proved that relevant content to SL community can be discovered by classifying videos based on content features. Further work by Karappa et al. [4] relaxed the constraints on videos in the pilot study and improved the recall and applicability of the earlier approach but with a considerable computational cost. Background subtraction, face detection, and polar motion profile generation combine to create a resource intensive process. Ways to reduce the amount of computation in each step will help to minimize the computation time to extract features which in turn would make the discovery of sign language content more applicable to the vast numbers of videos uploaded to sharing sites.

In this thesis, we propose techniques to minimize the amount of time taken to extract features from a video. The current design generates polar motion profiles for each video from the data generated through face detection and background subtraction. A certain length of the video is processed frame by frame for tracking hand movements. In each frame, faces are detected by using an ensemble of face detectors that use Haar-like features. The ensemble of face detectors is used to detect faces accurately and reduce the number of false positives in a frame. In parallel, the video is processed to track hands by background subtraction and the data from face detection. A Gaussian Mixture Model (GMM) is used for background subtraction and the parameters are decided empirically. Using data from face locations and foreground mask, the proportion of foreground pixels are calculated along the polar coordinate system with signer's face as the center of coordinate system and face proportions are scaled to provide translation and invariance. The generated Polar Motion Profiles are then averaged per video and are used to train an SVM classifier.

With this approach, we found that the amount of computation time in face detection is ten times to the computation time in background subtraction. Polar Motion Profiles can be generated only when data from both operations is available. Thus, reducing the amount of time taken for face detection is the focus of our efforts to improve the efficiency of the process. Hence, we evaluated the impact of alternate face detection techniques and different lengths of video segments on the precision and recall of the classifier. The following three approaches are evaluated:

1) Individual face detectors: In this approach, we replace the ensemble of face detectors with individual face detectors. Although accurate face detection is important for generating polar motion profiles, using five independent face trackers duplicates efforts. By testing the performance of the five individual trackers, we can determine which performs best and how much worse it is than when the five trackers are combined.
2) Shorter video segment lengths: Currently, a segment of one minute of each video is processed for feature extraction. We evaluate the impact of analyzing shorter segments of videos on the performance of the classifier. The reduction in video segment length saves computation by having fewer frames to be processed during feature extraction and also enables finer-grained diarization of videos containing sign-language and non-sign language content.
3) Frame sampling for face detection: The nature of sign language videos, where signers are most often deliberately signing to the camera, results in the face tending to be slow-moving if it moves at all. Thus, the change in the region of interest might not be significant between frames. Hence, we detect faces by sampling faces at regular intervals and evaluate how reduced sampling rates effect the overall recall and precision.

The three optimizations, using a single face tracker, processing short video segments, and only applying face tracking to sampled frames in the segment, can significantly reduce the computational cost of sign language detection. This thesis reports on the effects of the three optimizations individually and recommends and assesses a combination of the optimizations.

In the next section, we discuss the techniques used by researchers to recognize and detect sign language targeting varied applications. Then we provide a brief explanation as to why automatic detection of sign language is preferred over manual tagging by quantifying the video content being generated every minute in Section 3. In Section 4, we discuss our proposed work for this thesis and give an overview of the face detection, background subtraction and PMP generation. In section 5, we provide the data obtained in the evaluation of our approaches and discuss the performance of a recommended configuration. Section 6 concludes this thesis with a discussion of what we achieved and the future research feasible in sign language detection.

2. BACKGROUND / RELATED WORK

Sign language involves hand gestures, facial expressions and postures of the body to communicate. A significant amount of research has aimed at transcribing the lexical signs in sign language communication. Such a capability would be useful for those not in the sign language community to understand the videos in sign language and would also enable search over the content of sign language. However, this is a very hard problem and not likely to be applicable for real-world data in the near future. But the development of a sign language digital library need not involve understanding the content of videos but just the detection of sign language in video. In this section, we will discuss various techniques to recognize signs in a limited vocabulary first and techniques to detect sign language in the later section.

2.1 Sensor and Glove-based Recognition

Recognizing sign language from standard video is hard. One approach to recognizing sign language augments the video data. Signers may have to wear specially colored gloves that enable better hand shape detection or sensors like data gloves so the hand movements are tracked to recognize and transcribe the sign. Similarly, 3D video or trackers fall into this category as such data is not part of standard video.

Starner et al. [5] used a desk and wearable computer to track signers' hands and designed a Hidden Markov Model based system for recognizing American Sign Language (ASL). They experimented with a vocabulary of 40 signs and attained a word accuracy of 92% for a signer observed through desk computer and a word accuracy of

98% for signer wearing a hat that has a mounted camera. Earlier in [6], they recognized ASL from videos without explicitly modeling the fingers. In this system, they attained a word accuracy of 99% when tracking the signer wearing colored gloves and a word accuracy of 92% for signer not wearing data gloves. This approach was also tested with a limited vocabulary.

Assan & Grobel [7] developed a prototype that can recognize signs in real time when the signer is wearing special gloves. They used different colors for each finger and the palm. Handshape is recognized with a model that is comprised of colored areas, feature of those areas and the relation between those areas. The background is constrained to be uniform. Localizing the body of the signer, they extracted center-of-gravity of each finger and used it as a feature for the classifier. The classifier is signer dependent and the performance of classifier degraded when the signer for testing is not same as the person in training.

Liang & Ouhyoung [8] used a Hidden Markov Method (HMM) to recognize real-time continuous gestures that are part of sign language employing a DataGlove^{TM}. They segmented sentences explicitly before classification and decoupled sub-sign component-level and sign-level classification in which case they needed 51 components to recognize 71 to 250 signs, which is in agreement with the findings of linguists that a limited number of components can be combined to form a great number of sign words. They achieved a recognition rate of 80% on real-time gestures when they were performed slowly. The constraint on the speed of the gestures is to detect the word boundaries. The approach was signer dependent and the measures obtained are significantly worse when the signer in the test is different.

Bauer & Kraiss [9] used sub-units for recognition rather than the whole signs as well. This approach brings in the advantage that the HMMs need not be retrained as new signs are added. Datagloves are used for data acquisition and designed the system to classify 250 signs. They employed self-organized subunits since it is hard to define them for sign language. They achieved an accuracy of 92.5% for 100 signs and 81% accuracy without retraining of subunit HMMs after adding 50 new signs.

Hienz et al. [10] were able to track hands based on shape information alone. Signers were constrained to be in front of a dark background wearing long sleeved dark clothing. They localized elbows and shoulders, along with hands and face of a signer by using a color coded glove and colored markers. This way they were able to track the position and movement of hands with reference to the body of the signer. For feature extraction, they obtained 3D measurements by proposing a simple geometric model of the hand to estimate the hand’s distance to the camera using the shoulder, elbow, and hands 2D positions. By measuring 3D distances with multiple cameras directly, this approach provided better accuracy but at the cost of computational complexity compared to 2D distances. To recognize the simple types of patterns in German Sign Language, they developed a rule-based classifier that was able to obtain an accuracy of 95%.

2.2 Video-only Recognition of Signs

The approaches in the above sub-section get better recognition because the signer wears sensors or signs in a setting where additional data is captured in addition to the standard video. Also, most of the above approaches are signer dependent and cannot provide similar performance across signers. They do not fit the task to create a digital library as our task is to classify videos from the content without such constraints. The following approaches try to recognize signs in a video without such additional data.

Yang et al. [11] proposed an algorithm for extraction and classification of two-dimensional motion based on motion trajectories. Homogeneous regions are generated in each frame by performing multiscale segmentation. Two view correspondences are obtained by matching regions between consecutive frames. Pixel matches are defined by computing affine transformations from each pair of corresponding regions. Pixel-level motion trajectories are obtained by concatenating pixels matches over consecutive image pairs. A time-delay neural network is used to learn motion patterns from the extracted trajectories. They applied the proposed method for recognizing 40 hand gestures of ASL and obtained 98.14% recognition rate on training trajectories and 93.42% recognition on unseen test trajectories.

Somers & Whyte [12] matched the orientation of signer's hand using a set of three-dimensional hand models that are oriented at run time. They extracted a silhouette of the signer's hand and matched it against the pre-existing silhouettes of Irish sign language. The closeness of a match is determined by the Chamfer Distance Algorithm using four stereo pairs with each containing hand postures. Only one of the four is correctly identified by both and two were correctly matched in only one of the images and one was not matched in both images. This approach is highly susceptible to loss of finger information.

Dimov et al. [13] interpreted the task of recognizing letters from sign language alphabets as Content-Based Image Retrieval (CBIR). They created a database with a large pool of images for each letter in the sign language alphabets. When a random frame is given, they match the frame to the images in the database to find the closest match. For seven signs, they collected over 344 images and attained a recognition rate greater than 96%.

Similarly, Potamias & Athitsos [14] examined the use of embedding-based and hash table-based indexing methods for hand shape recognition by matching frames to existing images in a database comprising of tens of thousands of images of various hand shape appearances. They evaluated BoostMap and Distance-Based Hashing and found that input images can be matched at interactive speeds. BoostMap is 59 times faster than brute-force search and achieved a 99% of its retrieval accuracy. The maximum classification accuracy that can be achieved is only 33.1%; an upper-limit found using brute-force method.

2.3 Detection in Captured Videos

The approaches discussed above aim to recognize signs or parts of signs with limited vocabularies and constraints on the signer's position and the background. Recognizing sign language is useful for translating sign language content to non-signers. It is also helpful to reduce the bandwidth by employing avatars on both ends and transfer only the meaning rather than a high-quality video of the signer. But the techniques developed are not designed to be applied to the sign language video being recorded and shared via YouTube and other sites. They do not include sufficient vocabularies, do not work on full-speed sign language, and often expect a controlled setting for video capture. As such, they are not appropriate to the task of processing shared sign language videos. Here we discuss prior work on detecting sign language content.

Cherniavsky et al. [15] developed an activity detection technique to reduce the bandwidth of mobile video communication when the user is making gestures. Their system achieved an accuracy of 91% to detect if a user is signing, even after relaxing gloves and background constraints. The aim of the work is to reduce the bandwidth whenever there are no gestures. Hence, their technique cannot be applied to create a digital library as there is no way to distinguish if a user is signing or just making gestures.

For the problem of creating a digital library for sign language community, the classification need not involve recognition, but can be achieved by detection of SL in videos. Monteiro et al. [3] developed an initial technique for detecting videos with sign language. Based on common video analysis, they developed five features that were expected to be potentially valuable for creating a digital library. They found that a measure of the symmetry of movement relative to face was the best feature for the classification of videos. They achieved 82% precision and 90% recall with an SVM classifier trained with all the five features.

Karappa et al. [4] further relaxed the constraints in proof-of-concept study of Monteiro et al. [3] and included videos with multiple signers. They developed an accurate face detection technique using multiple face detectors based on Haar-like features in parallel. Using this technique, false negatives were reduced. Using the data from face detection and the foreground components in the video, motion was modeled using polar motion profiles. Upon training an SVM classifier with polar motion profiles, they were able to attain 81% accuracy and 94% recall on a dataset generated by collecting sign language and related videos from YouTube.

This thesis explores a variety of techniques to improve the computational efficiency of using polar motion profiles to distinguish sign language videos. The results provide data regarding how precision and recall will be impacted when trying to reduce the amount of time taken to extract features.

3. QUANTIFYING THE PROBLEM

With the evolution of the internet, the amount of video content shared has been rapidly rising. Video sharing sites have become a great destination for creators to share their content. YouTube has been a major component of the video sharing space since its inception and the amount of content uploaded to YouTube each minute has been increasing rapidly. For example, the hours of content uploaded to YouTube saw 200% growth from 2013 to 2014 as can be seen in Figure 1. By 2014, more than 300 hours of video were uploaded per minute. Using the empirical estimate of the average length of each video is 5 minutes results in a rough estimate of 3600 videos uploaded per minute.

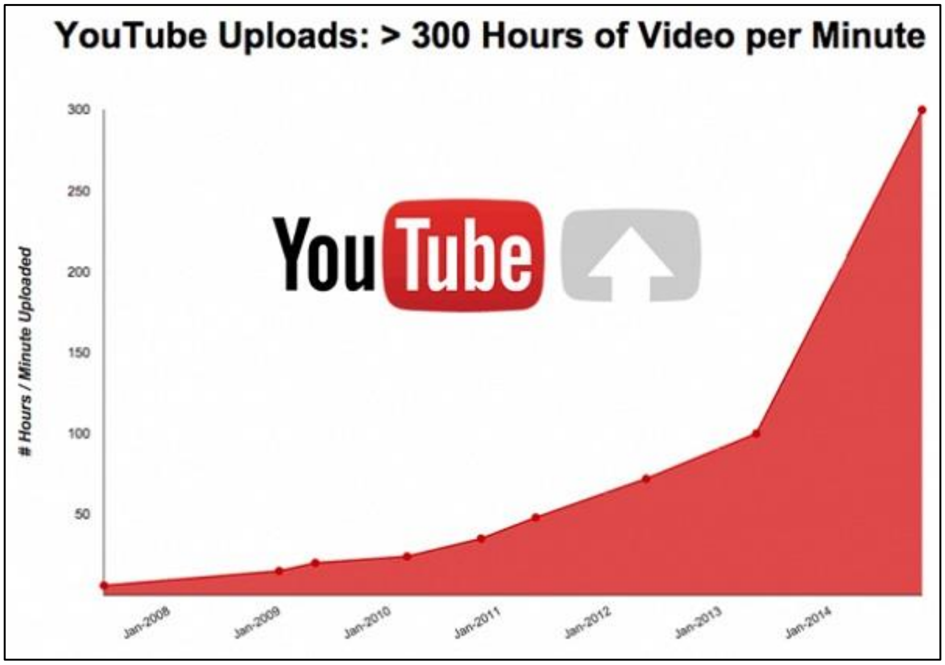

Figure 1 Hours of video uploaded to YouTube per minute [1]

Industry experts believe that the growth of uploads will continue due to the accessibility of high-quality mobile cameras. Due to the increasing mobility of cameras and cameras in almost always available smartphones, the population of content creators and the amount of content uploaded by prior content creators are both likely to grow.

Even without any growth in the quantity of content uploaded, the system would need to be able to process 3600 videos in a minute to classify new videos, not to mention the existing corpus. Hence, reducing the time taken to extract features from a video is important for many applications of sign language detection.

The approach explored by Monteiro et al. discussed in the related work section can extract features in near real time; that is it took approximately 1 minute to process each of the 1 minute video segments on a typical desktop computer. The work by Karappa et al. [4] obtained better recall and precision, but the feature extraction is not real-time due to the time-intensive face detection technique employed. If the time taken by the face detection algorithm is reduced, background subtraction can be completed in parallel and polar motion profiles generation can be pipelined.

4. PROPOSED WORK

In this section, we describe the approaches we focused to reduce the computation time in feature extraction for detection of sign language videos. Monteiro et al. [3] found that the symmetry of movement of hands with reference to the signer was the best feature among the five features they tried to detect sign language videos in their research. Later work by Karappa et al. [4] also used hand tracking to classify sign language videos. The work presented here uses the system developed by Karappa et al. [4] as base system as this approach had increased recall over a broader set of sign language videos. The particular focus of this thesis is to assess the impact of alternative techniques for reducing the feature extraction time.

In the following subsections, we will first describe the face detection and background subtraction used in the research and the areas we focused on reducing the computation time. Using the data obtained from face detection and background subtraction, we generated Polar Motion Profiles, a model developed by Karappa et al. [4]. The extracted features are used to train an SVM classifier for the purpose of detecting SL videos. Next, we will describe the three approaches for reducing the time taken to classify sign language. Figure 2 show the architecture of the classifier system. The face detection and background subtraction are done in parallel.

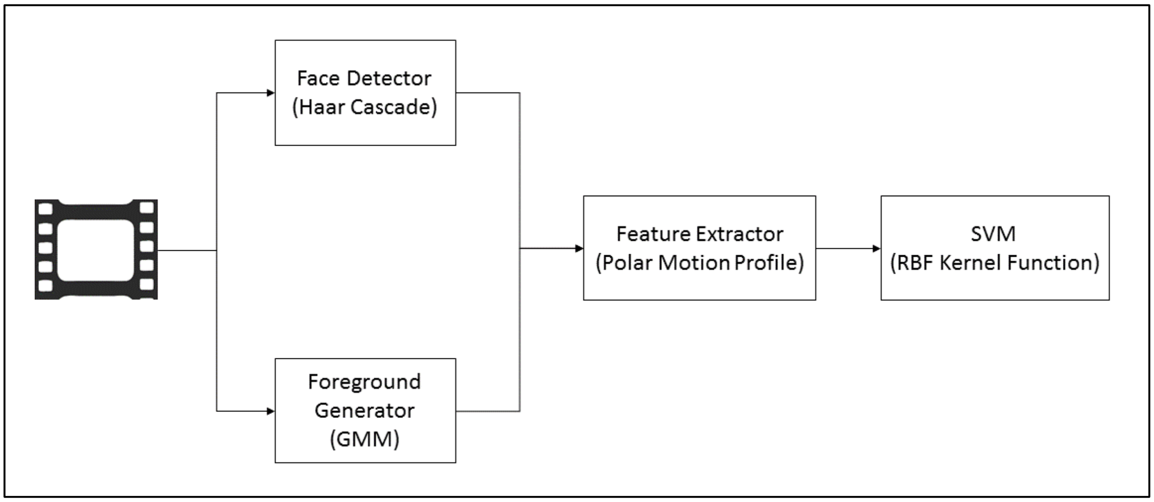
Face Detector (Haar Cascade)
Foreground Generator (GMM)
Feature Extractor (Polar Motion Profile)
SVM (RBF Kernel Function)

Figure 2 Architecture of the classifer system

4.1 Hand Tracking

Hand gestures are an important part of sign language communication. The relative symmetry of hand movements has been used in most research works. In a sign language video, hands are constantly moving while the movement of the signer's body and head is relatively smaller. In videos with a relatively slow changing background, hands can be tracked by using background subtraction. The foreground pixels obtained by background subtraction can be attributed to a signer by defining a region of interest around the signer's face. In the following subsections, we will describe the techniques used to detect faces and background subtraction.

4.1.1 Face Detection

For face detection, we evaluated the five face detectors provided in OpenCV, a BSD-licensed library free for academic and commercial use; and the ensemble model, developed by Karappa et al. [4] to accurately detect faces by taking a majority of votes of the five face trackers. The assumption we had to test single face detector instead of the ensemble is that even if there is a false positive, non-activity in the ROI defined might not contribute to feature extraction.

The frontal face detectors provided in OpenCV are based on object detectors proposed by Viola & Jones [16] and improved by Lienhart & Maydt [17]. A cascade/tree of boosted classifiers working with Haar-like features is trained with a few hundred sample views of frontal faces scaled to same size and some arbitrary images serving as negative samples. The classifier can be easily resized in order to find objects of different sizes rather than resizing the image. To find a face with no information on size, the trained classifier is applied to a region of interest with an output of detection and the search window is moved across the image.

Each overall classifier is comprised of simpler classifiers (stages) that are either cascaded or made recursively in a tree-like structure as shown in Figure 3 as long as a candidate is either rejected at some stage or passed all stages.

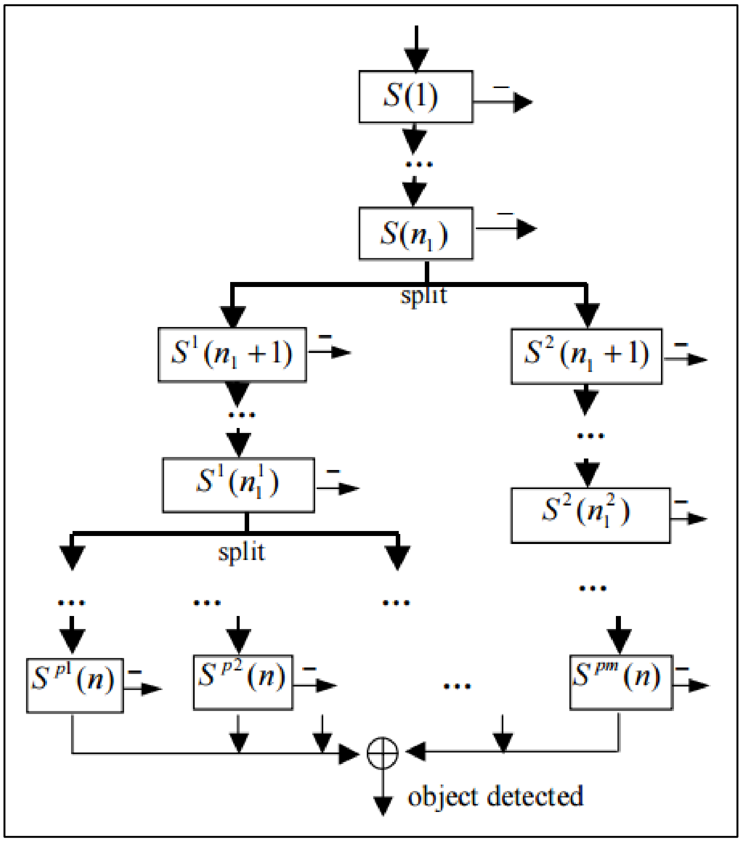

Figure 3 Tree of classifiers [2]

The face detectors tested include either multi-level decision trees or stumps. Decision trees with at least two leaves are used as the basic classifiers. A stump is a machine learning model with a one-level decision tree. These basic classifiers take Haar-like features as inputs. Complex classifiers are built at each stage of the cascades using adaptive boosting via Discrete Adaboost and Gentle Adaboost.

The face detectors tested in this thesis are as follows [18]:

- Frontal face detectors using a cascade of stage classifiers contributed by Rainer Lienhart:

1. *Haar-cascade Frontal Face Default*: Stump-based 24x24 discrete adaboost frontal face detector
2. *Haar-cascade Frontal Face Alt*: Stump-based 20x20 gentle adaboost frontal face detector
3. *Haar-cascade Frontal Face Alt2*: Tree-based 20x20 gentle adaboost frontal face detector

- Frontal face detector using tree of stage classifiers contributed by Rainer Lienhart:

1. *Haar-cascade Frontal Face Alt Tree*: Stump-based 20x20 gentle adaboost frontal face detector

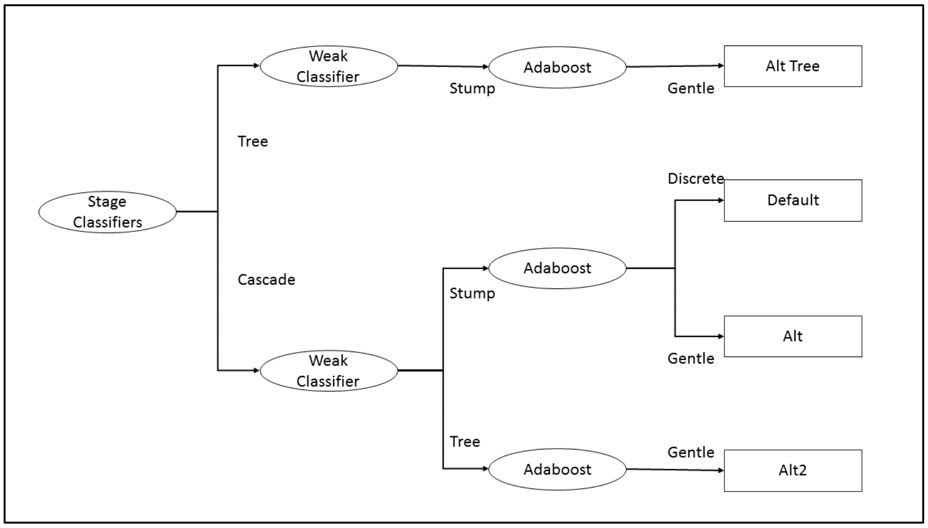

Figure 4 Frontal face detector training design decisions

- Profile face detector contributed by David Bradley from Princeton University:

    - *Haar-cascade Profile Face*: 20x20 profile face detector

- *Ensemble* of above five cascade detectors

  Karappa et al. [4] developed this algorithm in which a given frame is passed through each of the five cascade detectors to detect faces that might include false positives. Using the bounding boxes obtained from all the detectors, a combination ${}^{n}_{3}C$ sets are formed. Each set of three bounding boxes is tested for overlap and discarded the false positives if the boxes do not overlap. An empirical threshold of 40 pixels between the corners of the bounding boxes is used to determine the overlap. If overlap is detected, the average of corresponding corners is taken as the bounding box for the face location.

We examine the accuracy and computation time of the above-discussed face detection techniques to find the balance between the computation time involved for face detection and their impact on precision and recall for sign language detection.

4.1.2 Background Subtraction

Background subtraction is a common computer vision task. Friedman & Russell [19] proposed Gaussian Mixture Model (GMM) for background subtraction. OpenCV has an implementation of GMM proposed by Zivkovic [20] which is very fast and also performs shadow detection.

The tunable parameters for background subtraction using Zivkovic implementation are [21]:

- nmixtures: Maximum allowed number of mixture components where the actual number per pixel is determined dynamically

- backgroundRatio: Threshold that defines whether the component is significant enough to be included in the background model. The default value of 0.9 is used.
- varThresholdGen: Threshold for the squared Mahalanobis distance that helps determine if a sample is close to the existing components. A new component is generated if it is not close to any component. A smaller threshold generates more components while a higher threshold results in fewer components which can grow too large. The default value, i.e., three times the standard deviation is used.
- fVarInit: Initial variance for the newly generated components. This value affects the adaptation speed. The default value of 15 is used.
- fVarMin: Minimum variance for a generated component
- fVarMax: Maximum variance for a generated component
- fCT: Complexity reduction parameter defines the number of samples needed to accept that the component exists. A value of 0 would result in an algorithm similar to the standard Stauffer & Grimson algorithm. The value is set to 0.05.
- nShadowDetection: Value to mark shadow pixels in the generated foreground mask. The default value of 127 is used.
- fTau: Threshold that determines how darker the shadow can be.

After generating the foreground mask, we apply morphological opening, i.e., erosion followed by dilation to remove noise and to fill gaps in the detected objects.

4.2 Feature Extraction

Face detection and background subtraction can be done in parallel. Using the bounding box from face detection, a region of interest (ROI) around each face is generated. Within the ROI, the proportion of foreground pixels is computed in the polar coordinate system. The computed proportion of foreground pixels along the radial and angular coordinates is termed as Polar Motion Profile [4]. For a given ROI, PMP is computed as the ratio of foreground to total number of pixels along the polar coordinates (ρ, θ):

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

PMP provides a measure of activity in proximity to a person that is translation and scale invariant. The Polar Motion Profiles are used as features to train a SVM classifier that is used to classify videos containing sign language from non-sign language videos.

4.3 Training and Classification

The PMP generated for each face in a video is averaged along the radial and angular coordinates. The average of PMPs in a video along the angular coordinate is computed as

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

In the same way, the average along radial component is computed. The averaged PMPs are reduced to 5 dimensions through Principal Component Analysis. The resulting outcome is used as a feature to train an SVM classifier with an RBF kernel.

4.4 Optimization Approaches

There are a variety of approaches available to reduce the computation time. Each of these techniques may detrimentally affect classification accuracy. This thesis explores reduce face detection computation time by using a single fact detector instead of the ensemble approach, only applying face detection on sampled frames, and processing fewer frames by reducing the length of the video segments used for classification. Background subtraction happens at real-time, i.e., a second of a video is processed in a second. If the face detection time can match the background subtraction, the lag to generate PMPs can be minimized.

4.4.1 Individual Face Detector for Face Detection

Face detection is computationally expensive. The ensemble of five face detectors proved to be effective in reducing false positives but at the expense of computational cost. False positives in face detection can introduce PMPs with no signer in the region. When there is no signer, the activity in the region of interest might become trivial due to morphological opening (erosion and dilation) after background subtraction. Hence, the PMP for the false positive may not significantly affect the performance of the classifier. To support this hypothesis, we replaced the ensemble of face detectors with each of the five individual face detectors and measured the effects on recall and computation time.

4.4.2 Face Detection on Sampled Frames

This approach to reducing computation for face detection examines how performing face detection on sampled frames (i.e. on every Nth frame) affects sign language detection. We had reason to believe that sampling would have limited effects on performance as signers’ faces and bodies tend to be relatively stationary while signing in the majority of sign language videos on video sharing sites. The ROIs for the frames between sampled frames were the last computed ROI.

4.4.3 Shorter Video Segments

A one minute segment of the original video was used to perform classification by both Monteiro et al. and Karappa et al. This means that face detection and background subtraction was performed on each frame in that segment. Thus, reducing the length of the chosen segments reduces computation but the resulting PMPs may be less representative of the overall video. How short is too short? Answering how varying the length of the selected segment affects sign language detection accuracy not only informs the design of optimized SL detectors but helps answer to what degree fine-grained diarization, that is recognizing segments of a video that include sign language from those that do not, can be achieved.

4.4.4 Recommended Overall Configuration

Using the results from the above three assessments, we define a recommended configuration that we expect to substantially lower computation time while not sacrificing too much precision and recall. We report on the overall performance in terms of both computation and accuracy.

5. RESULTS

For validating the approaches against known results, we used the existing corpora created by Monteiro et al. [3] and Karappa et al. [4] referred from now on as dataset A and dataset B respectively. Both of the datasets were collected from online video sharing sites like YouTube and were manually labeled as sign language and non-sign language videos.

Monteiro et al. [3] created dataset A for their proof-of-concept study. The dataset includes 100 sign language videos containing static backgrounds and a single signer; 100 non-sign language videos that were mostly videos thought to be likely false-positives as they a person making random hand gestures.

Karappa et al. [4] created dataset B for their research to classify sign language videos using Polar Motion Profiles. This dataset relaxes the constraints of requiring a static background and a single signer that were used to create dataset A. Sign language videos were obtained by a query ‘American Sign Language’ in YouTube. The resulting videos were manually labeled into sign language and non-sign language corpus. The non-sign language videos were obtained by collecting the related videos suggested by YouTube which resulted from the search query. This corpus contained 111 sign language videos with no constraints on either background or the number of signers and 116 non-sign language videos that are considered related to the sign language videos in the corpus by the video sharing site. As such, this dataset closely resembles the set of videos that would need to be classified when creating a sign language digital library.

In the subsections, we will discuss the results obtained for the approaches discussed in the last section. In all the approaches, we also obtained results from the model designed by Karappa et al. [4] as a reference to compare the recall and precision achieved by our approaches. Each value reported is the average of 50 iterations with each iteration choosing samples randomly from the dataset except for the values reported in subsection 5.4, which are an average of 500 iterations.

5.1 Time to Process a Video Segment of a Minute Length

We ran the five individual face detection techniques and the ensemble technique on all the videos from dataset A and dataset B to find the amount of computation time. The length of each video segment was chosen to be one minute. Dataset A comprised videos with a resolution of 120p and dataset B contained videos with a resolution of 240p. Detecting faces in both datasets provided us data on how scaling the resolution of videos will impact the time taken for face detection task. Most video sharing sites provide multiple resolutions for a given video. Although a video with lower resolution can be chosen, higher resolution videos can be a good choice for classifying videos with a signer in a snippet of the video rather than occupying the full frame (although we do not have data on this aspect). Hence, we chose both to test the algorithms on both datasets to calculate the time taken for detecting faces with different resolutions.

The individual face detectors are able to detect faces in the range of a minute to one and half minutes for the one minute videos in dataset A. Thus, the processing is real-time for videos of resolution 120p. The ensemble of five cascades is able to detect faces in 10-11 minutes. The results can be seen in Figure 5.

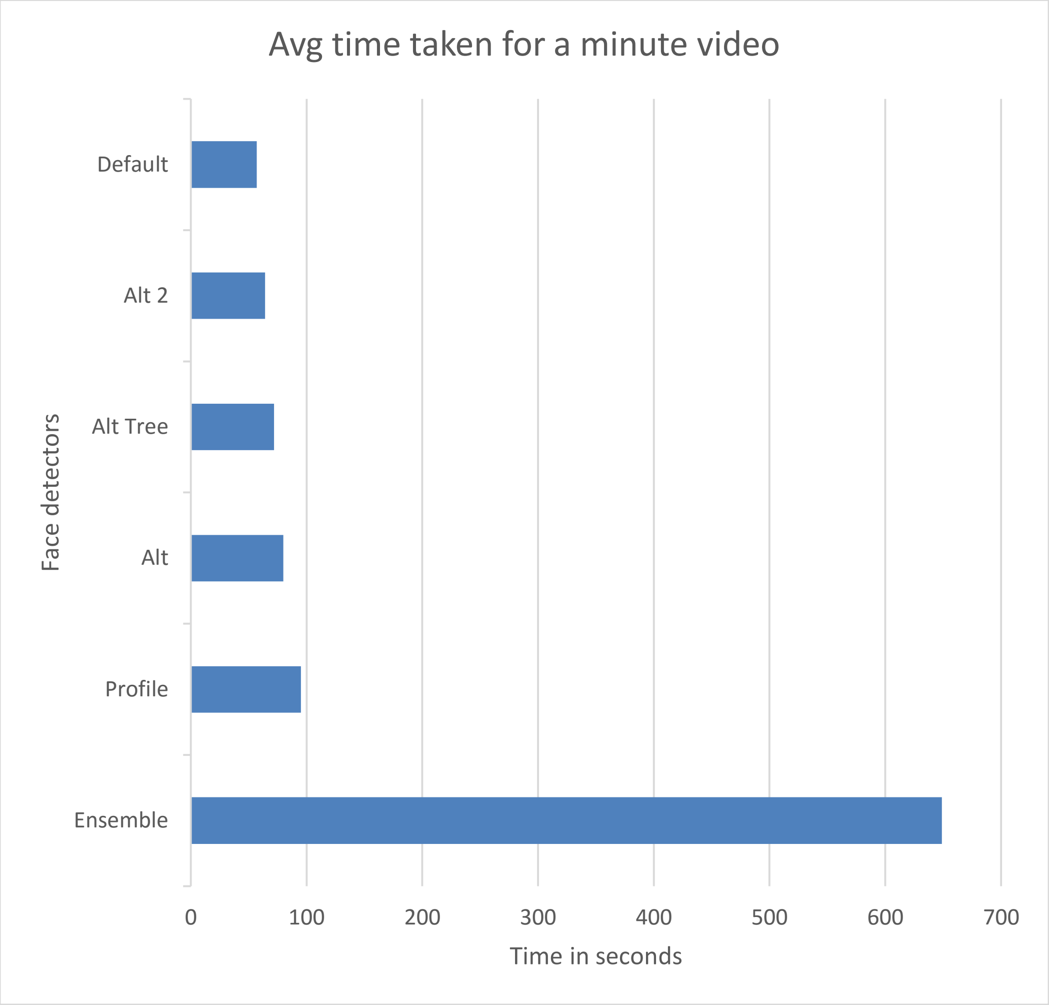

Figure 5 Average time taken by each face detector for a minute video on dataset A

Figure 6 presents the results of the same process for the higher-resolution videos in dataset B. The order of time taken by face detectors is the same except that Alt2 and Alt Tree frontal face detectors traded places for dataset B. The processing time to detect faces increased by half a minute to one and half minutes for dataset B for the individual face detectors as the resolution of videos has been increased from 120p to 240p. The ensemble approach took an average of 14-15 minutes for a video of one minute length. The results obtained for dataset B can be seen in Figure 6.

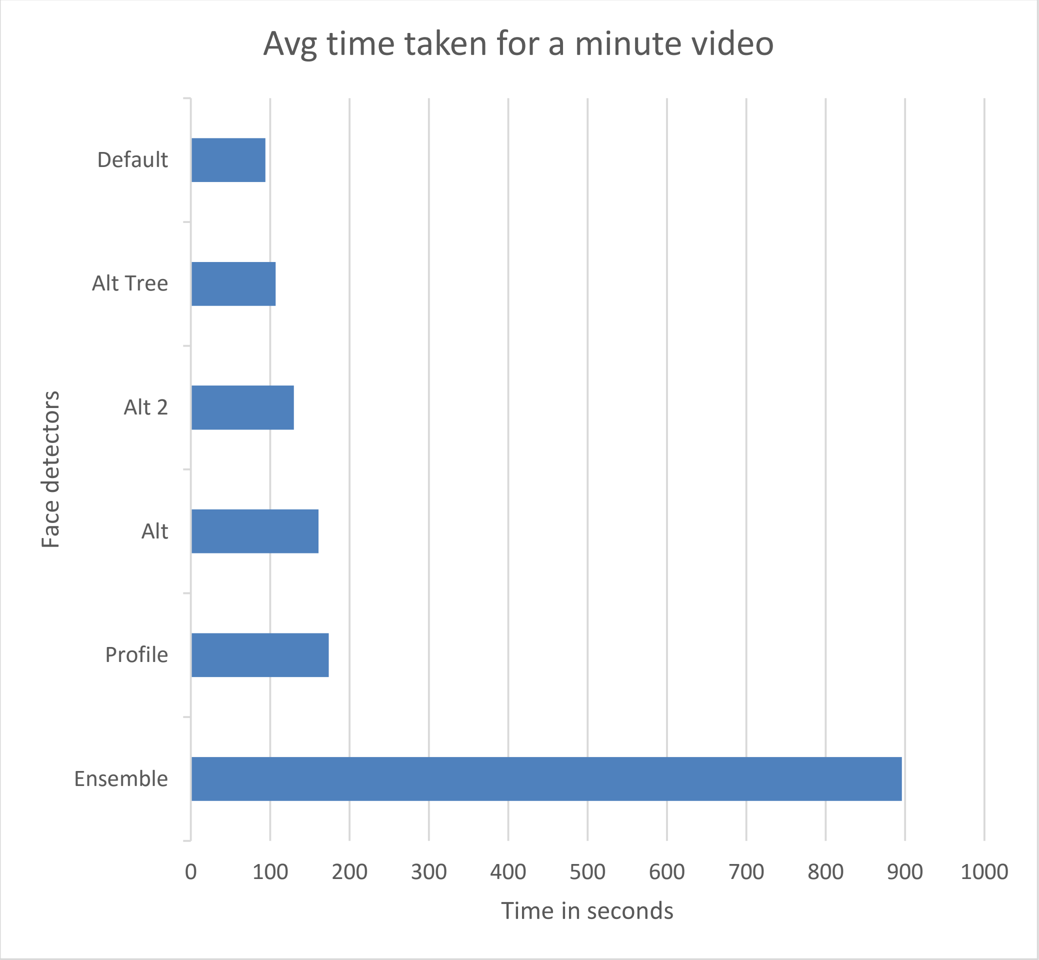

Figure 6 Average time taken by each face detector for a minute video on dataset B

As expected, raising the resolution of the videos increases the time taken to detect faces. While face detection is done in near real-time for videos with a resolution of 120p with a single face detector, the minimum resolution currently supported on YouTube is 144p and some videos start at 240p resolution. Hence, even if the ensemble approach is replaced with a single face detector, optimizations are still needed to reduce face detection to real-time.

5.2 Evaluation of Individual Face Detectors for Sign Language Detection

The data above indicates that using a single face detector in place of the ensemble of detectors can bring down the computation time for face detection by a minimum of five times and up to ten times depending on the particular selection. But such a choice may negatively affect the accuracy of results. To determine the effect on accuracy, we evaluated the performance of the classifier using Polar Motion Profiles generated from face locations detected through each of the five individual face detectors.

In addition, we also explored the effect of training set size on accuracy of the face detectors. The x-axis on each of Figures 7 through 12 indicates the number of samples from sign language and non-language videos used for training while the rest of the videos in the dataset serve as test samples.

Figure 7 and Figure 8 compare the precision of results obtained when using individual face detectors to the ensemble of face detectors. Precision measures the probability that a video classified as a sign language video really was a sign language video, thus it is primarily a measure of false positives. The precision of classifiers with frontal face detectors using a cascade of classifiers (Default, Alt and Alt2) is almost equivalent to the precision obtained by the classifier with the ensemble of face detectors. Within the individual face detectors, the Alt Tree classifier shows the highest variance in precision between the data sets – it performs the worst on dataset A and the best on dataset B in terms of precision.

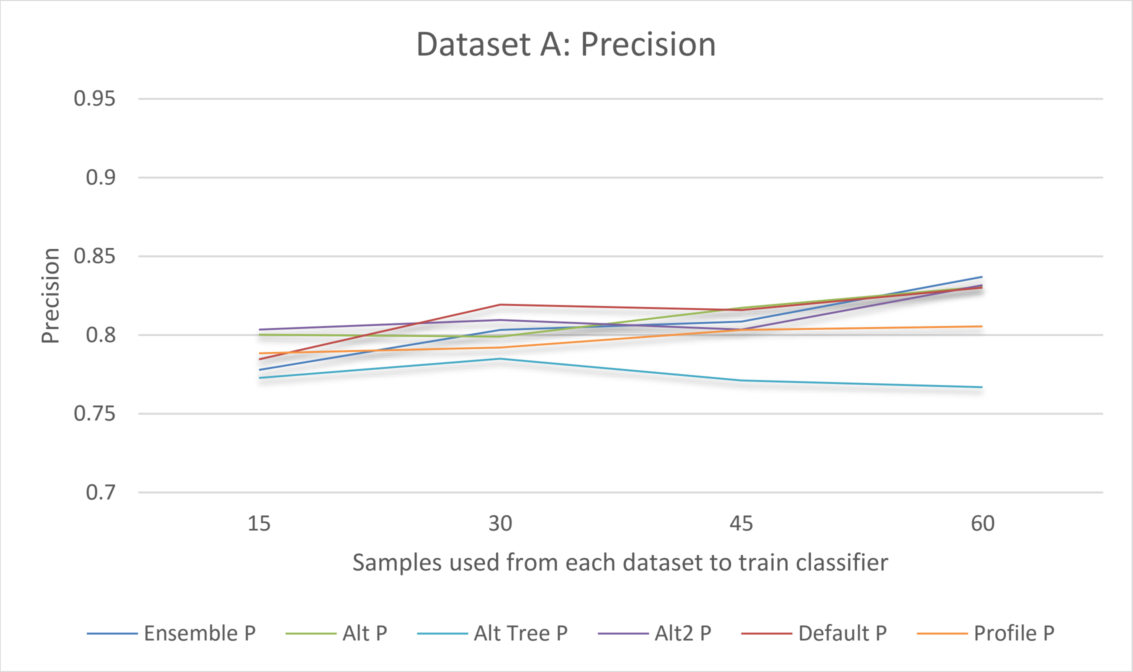

Figure 7 Precision obtained when using different face detectors on dataset A

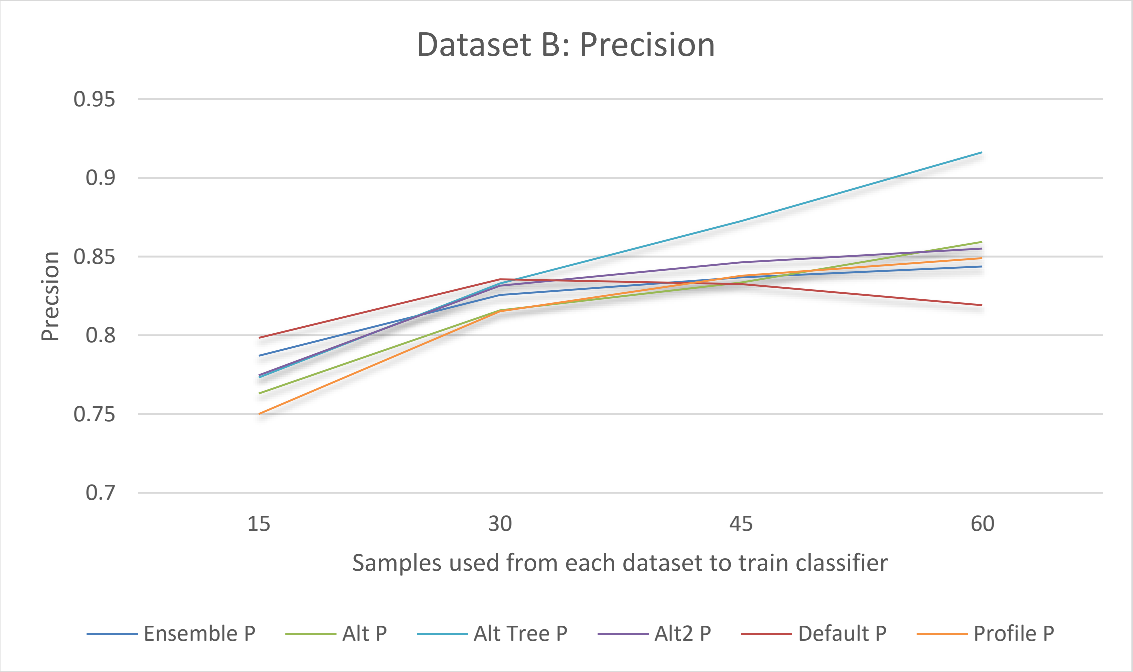

Figure 8 Precision obtained when using different face detectors on dataset B

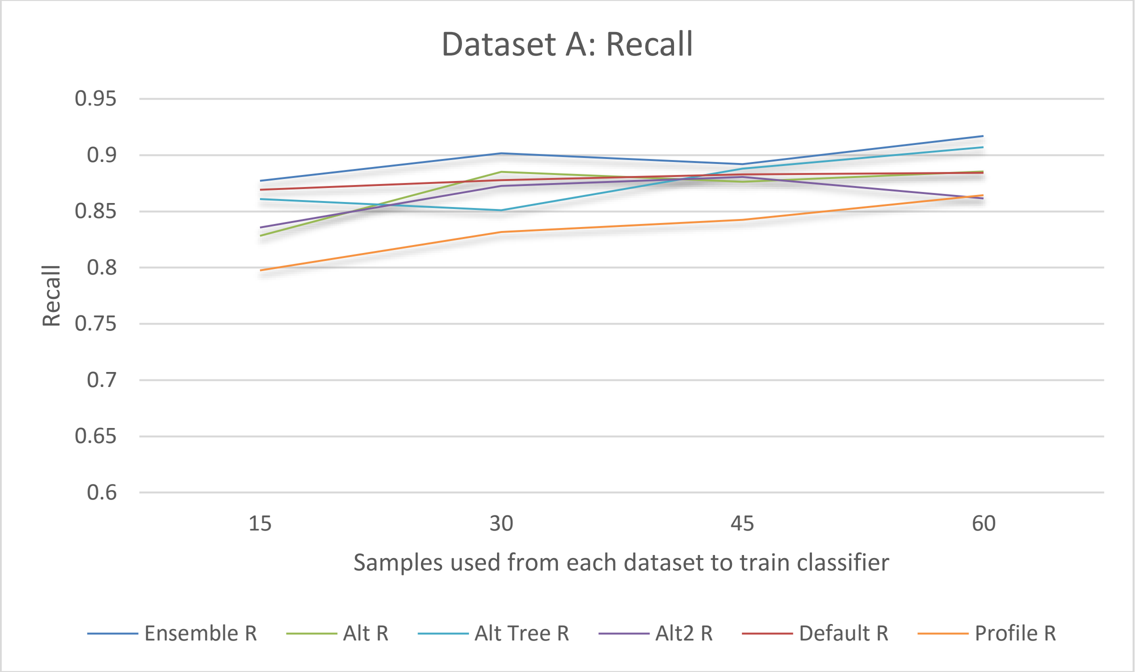

Figure 9 Recall obtained when using different face detectors on dataset A

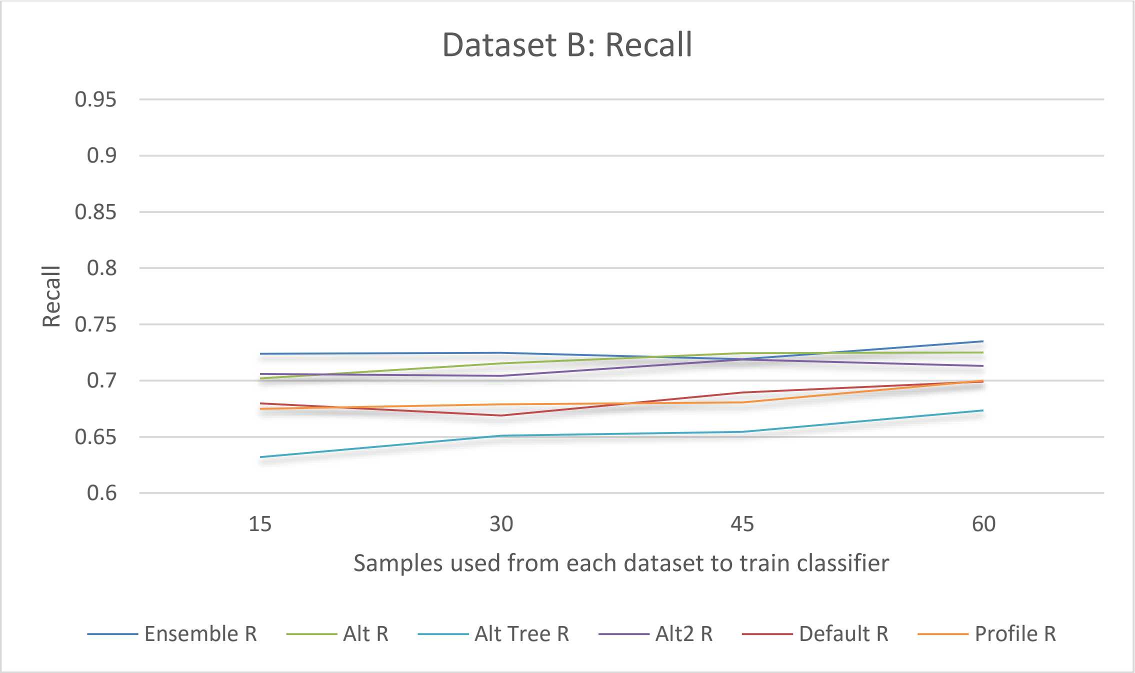

Figure 10 Recall obtained when using different face detectors on dataset B

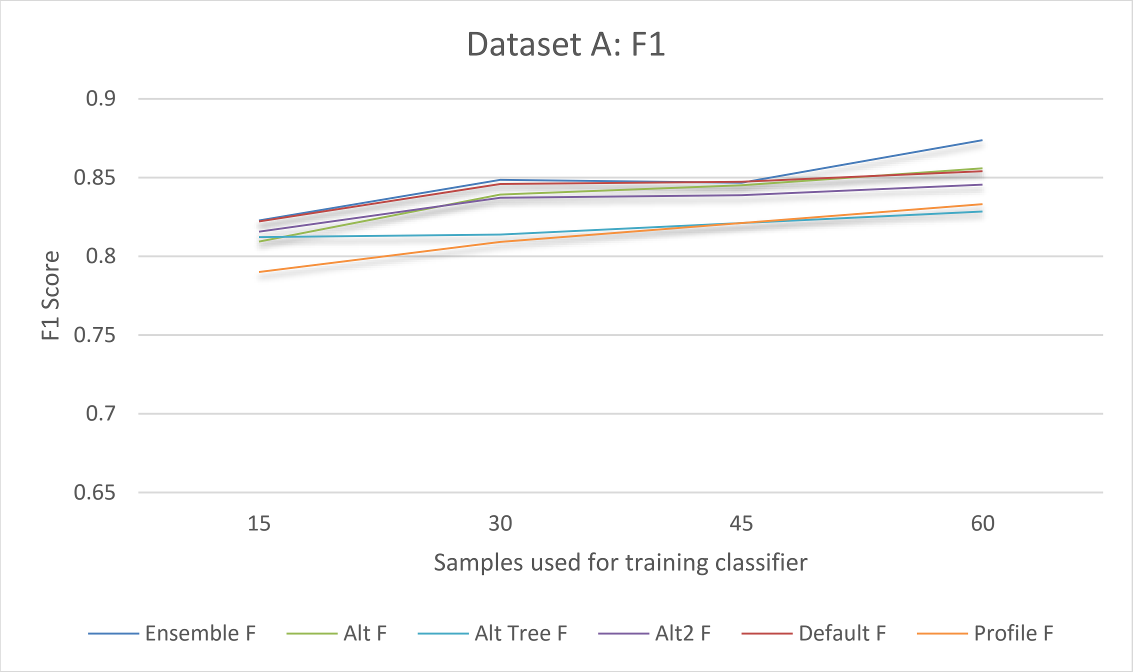

Figure 11 F1 obtained when using different face detectors on dataset A

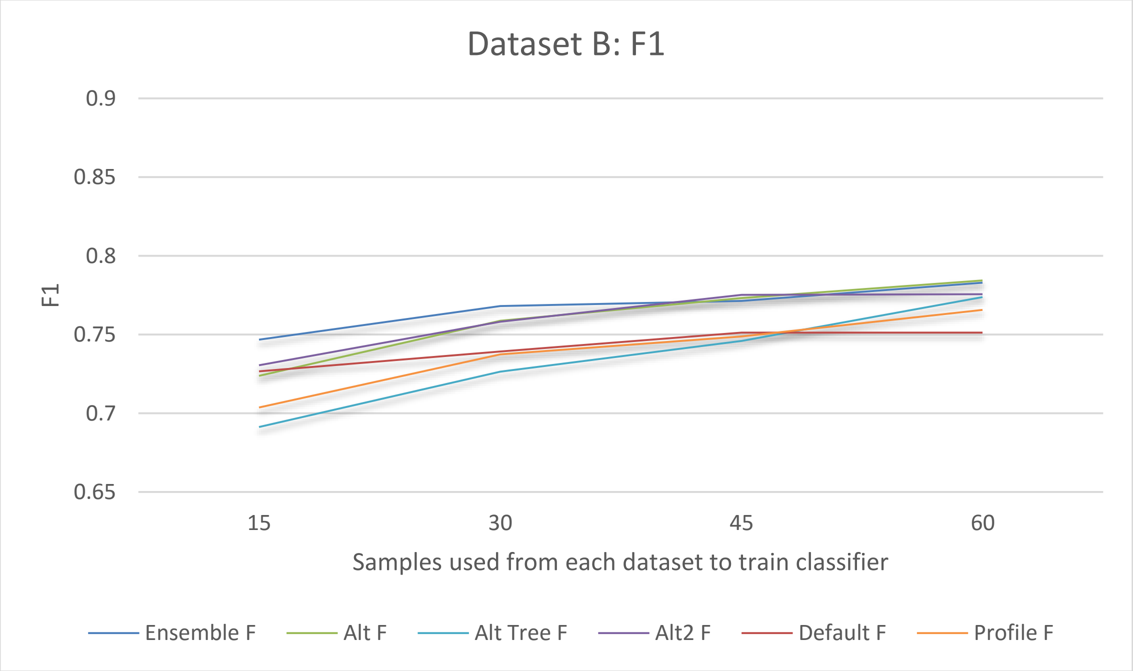

Figure 12 F1 obtained when using different face detectors on dataset B

Figures 9 and 10 compare the effects of the different face detection techniques on recall. Recall is the probability that a sign language video will be identified as one by the classifier, thus it is primarily a measure of false negatives. The figures show that the recall results for dataset B are considerably lower than they are for dataset A with one notable exception. The recall obtained by the classifier with the profile face detector is highest on dataset B which might point to dataset B containing more profile faces when compared to dataset A, which is true by manual observation. But the accuracy attained by that classifier is low and inconsistency based on content makes it a bad candidate for face detection module.

In most applications, both precision and recall are important for classifiers. Figures 11 and 12 present the F1 score for the classifiers based on the six face detection approaches. F1 is the harmonic mean of recall and precision and is frequently used in the information retrieval community to assess overall accuracy. Figures 11 and Figure 12 show that the ensemble of face detectors is the best face detection technique to be used when training with only limited number of training samples and performs well overall. As the number of training samples are increased, the classifiers with frontal face detectors employing a cascade of stage classifiers and adaptive boosting i.e., alt and alt2, performed well in both datasets. Overall, the range of F1 scores shows that using a single face detector instead of the ensemble detector does not substantially impede sign language detection.

Thus, based on accuracy, our recommendation for an approach to face detection in the sign language classifier is either the alt or alt2 frontal face detectors. In the above subsection, we found that alt2 frontal face detector has a computational advantage over the alt frontal face detector. Taking both accuracy and computation time into consideration, we chose the alt2 frontal face detector for over the alternatives for our recommended configuration.

Figure 13 gives an overview of the design decisions in choosing a face detector for detecting faces that are provided to Polar Motion Profile generation. Cascade of stage classifiers provides consistent performance and although classifier with detector using discrete adaboost performed well in dataset A, it could not maintain its performance in dataset B.

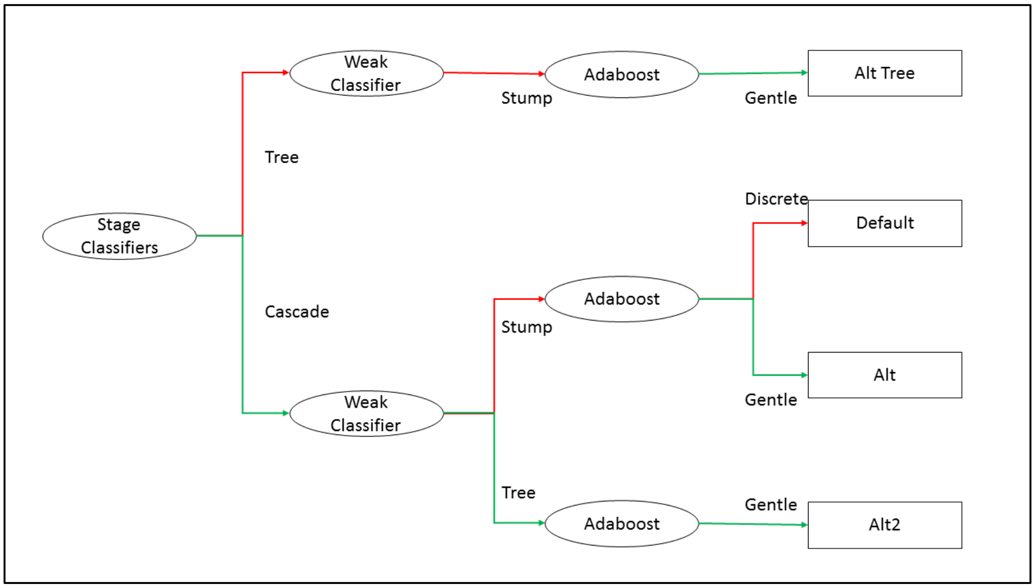

Figure 13 Frontal face detectors design choice for our system

5.3 Evaluation of Performance of Classifier with Shorter Segments of Videos

Shortening the length of video segments for feature extraction and classification provides two advantages: first, the amount of computational time for feature extraction can be substantially reduced, and second, it enables more fine grained later diarization of videos containing both sign language and non-sign language content. We evaluated classifiers with the individual face detectors to find how they performed relative to the classifier with the ensemble of face detectors with shorter segments of videos. Figures 14, 15, 16, 17, 18, and 19 show the precision, recall and F1 as the length of segments are reduced from 60 sec to 5 sec.

To select the shorter segment, we took the segment at the center of the first-minute segment of the full video. For example, for a two-minute video, a 30 sec segment would be chosen from the 15 second point to the 45 second point in the original video. This is done to avoid the non-sign language start up portions at the beginning of videos. For training the classifier, we used 50 samples from each of the sign language and non-sign language corpus.

Figures 14 and 15 show how precision varied as function of the length of the segment of video processed. For dataset A, the precision holds relatively steady for all techniques except for alt tree, which degrades quickly. The alt tree approach performed so poorly with dataset B that it does not appear in Figures 15, 17, and 19. Although the classifier with the default frontal face detector obtained precision near that of classifiers with alt, alt2, profile, and ensemble face detectors for dataset A, the precision obtained by the default face detector was reduced for dataset B. In both datasets, classifier with alt and alt2 frontal face detectors achieved a precision equivalent to that of ensemble of face detectors.

Figures 16 and 17 present the recall performance as segment length is shortened. As opposed to the precision results, the shorter the segment the worse the recall was for all classifiers. Similarly, Figures 18 and 19 show the F1 scores for the alternative face detectors as the segment lengths vary. Classifiers with the alt tree frontal face detector or the profile face detector could not achieve a reliable performance. Overall, the alt and alt2 frontal face detectors achieved accuracy comparable to the ensemble of face detectors across the range of segment lengths. As alt2 outperforms alt for short segments in dataset A and dataset B and has lower computational cost, these results reaffirm the selection of alt2 as an appropriate choice for our recommended configuration.

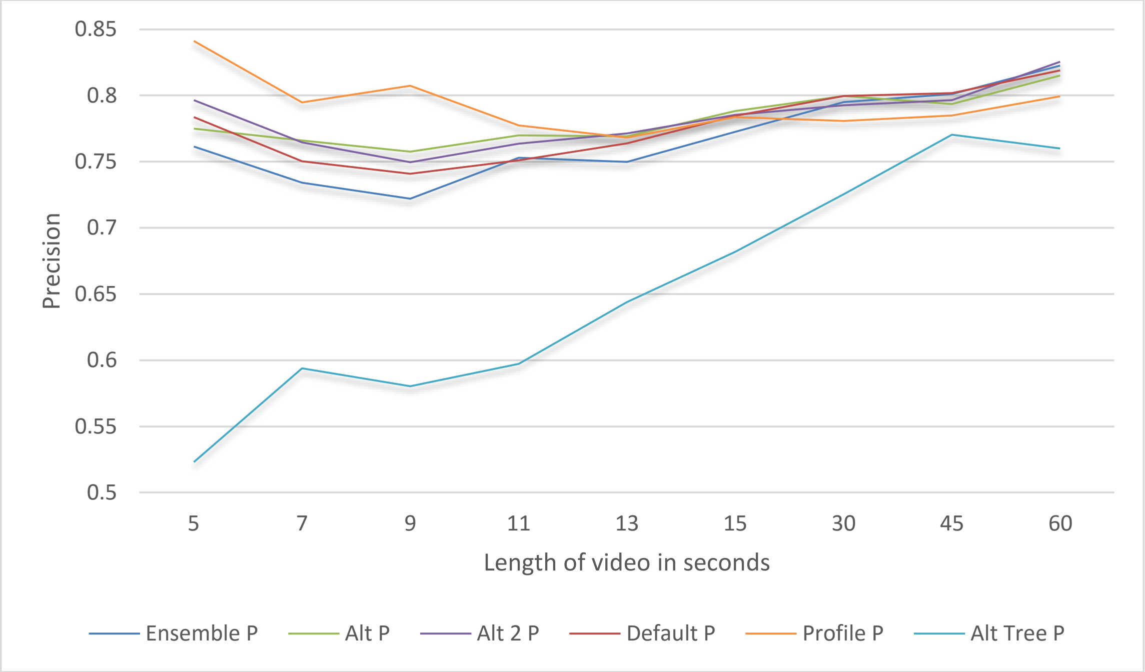

Figure 14 Precision obtained for shorter video segments in dataset A

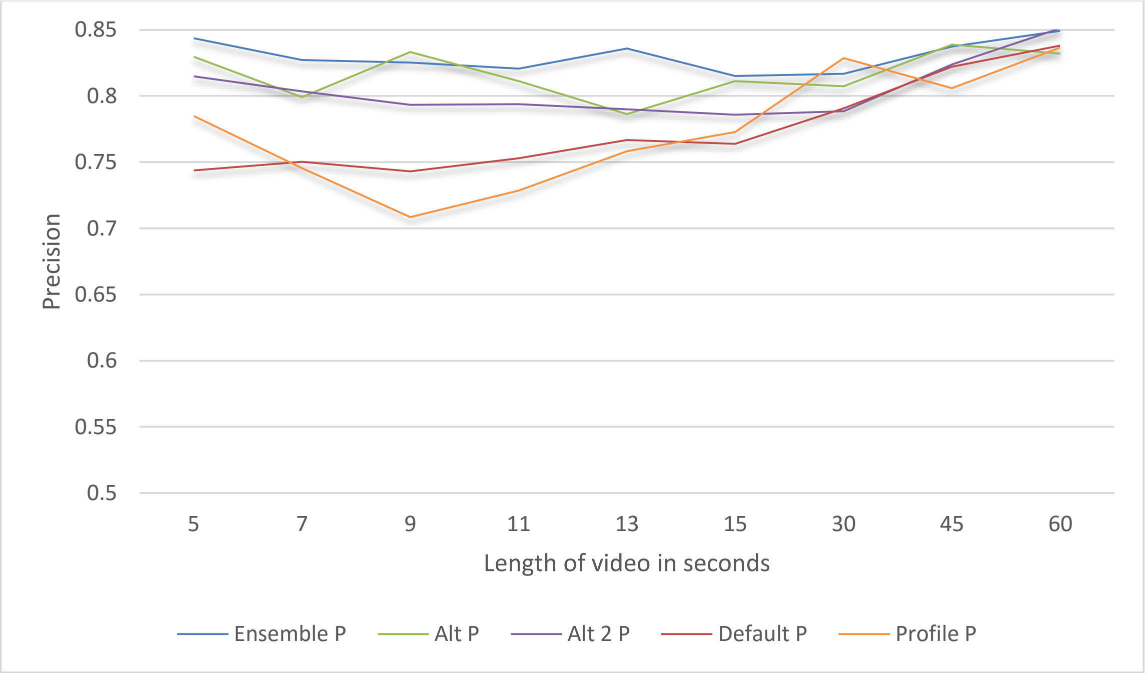

Figure 15 Precision obtained for shorter video segments in dataset B

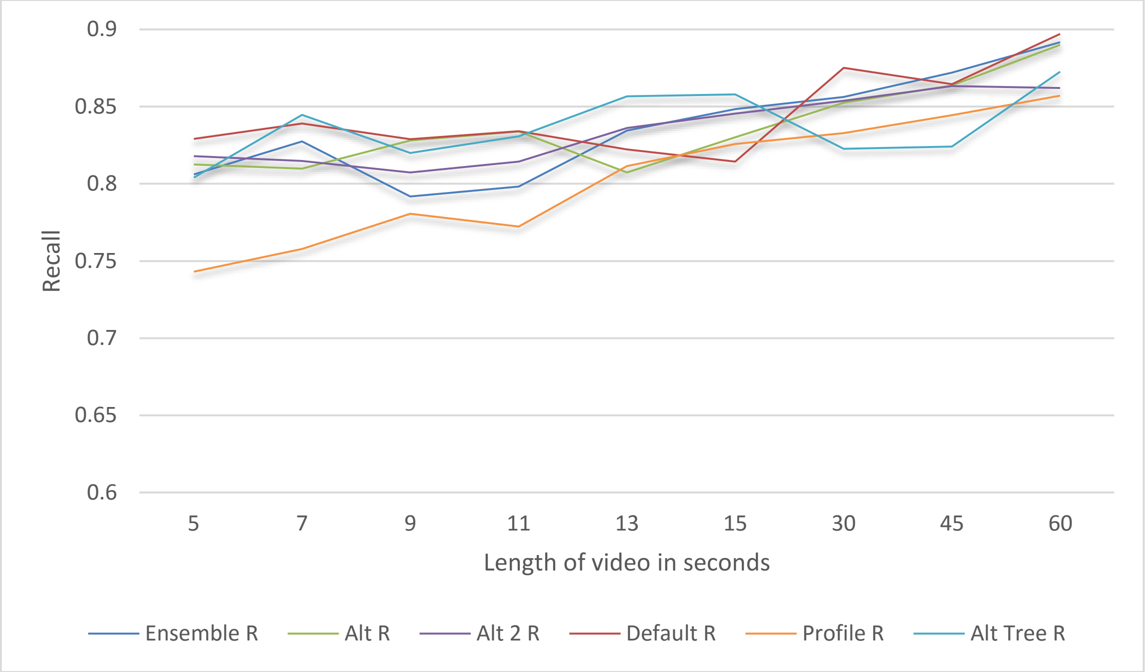

Figure 16 Recall obtained for shorter video segments in dataset A

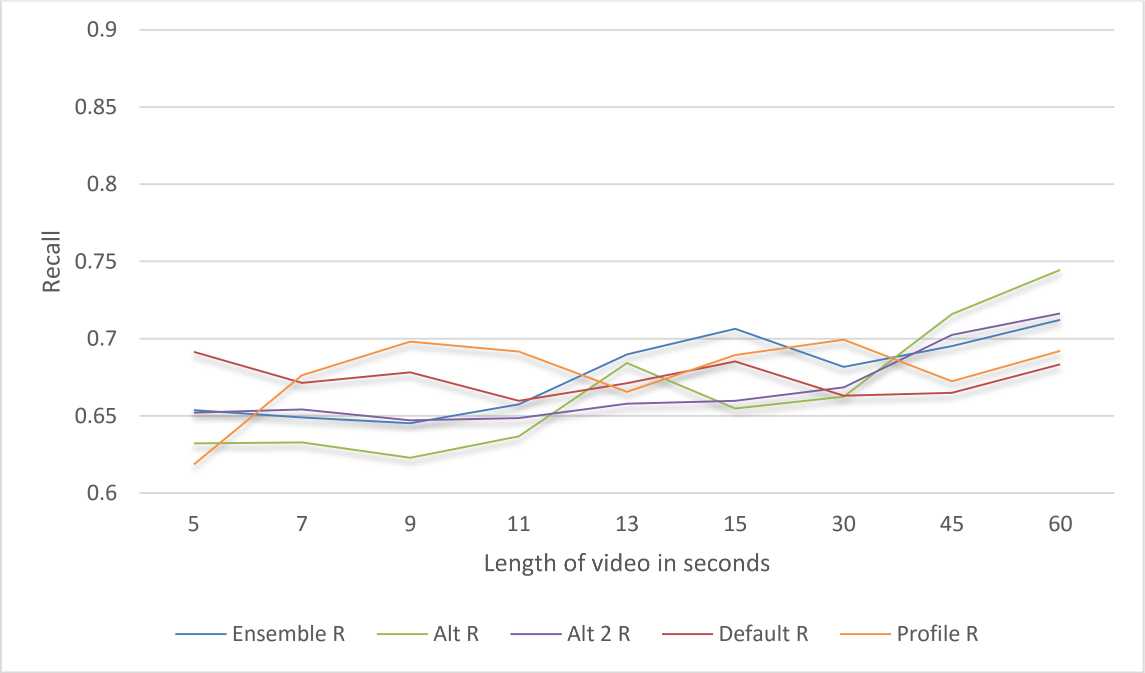

Figure 17 Recall obtained for shorter video segments in dataset B

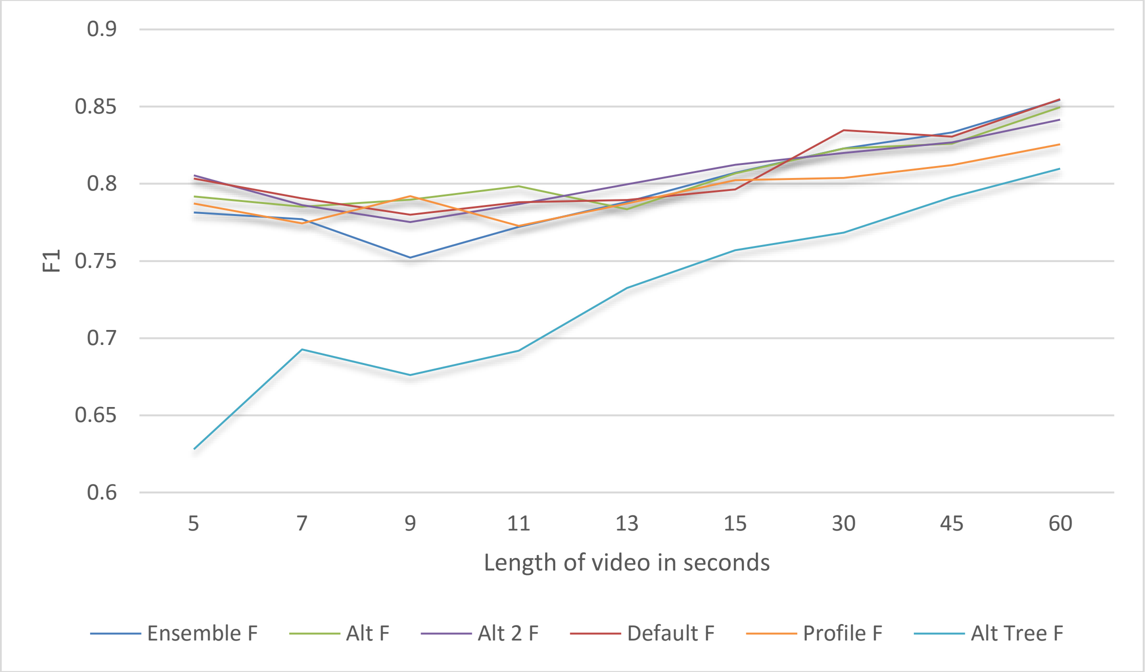

Figure 18 F1 obtained for shorter video segments in dataset A

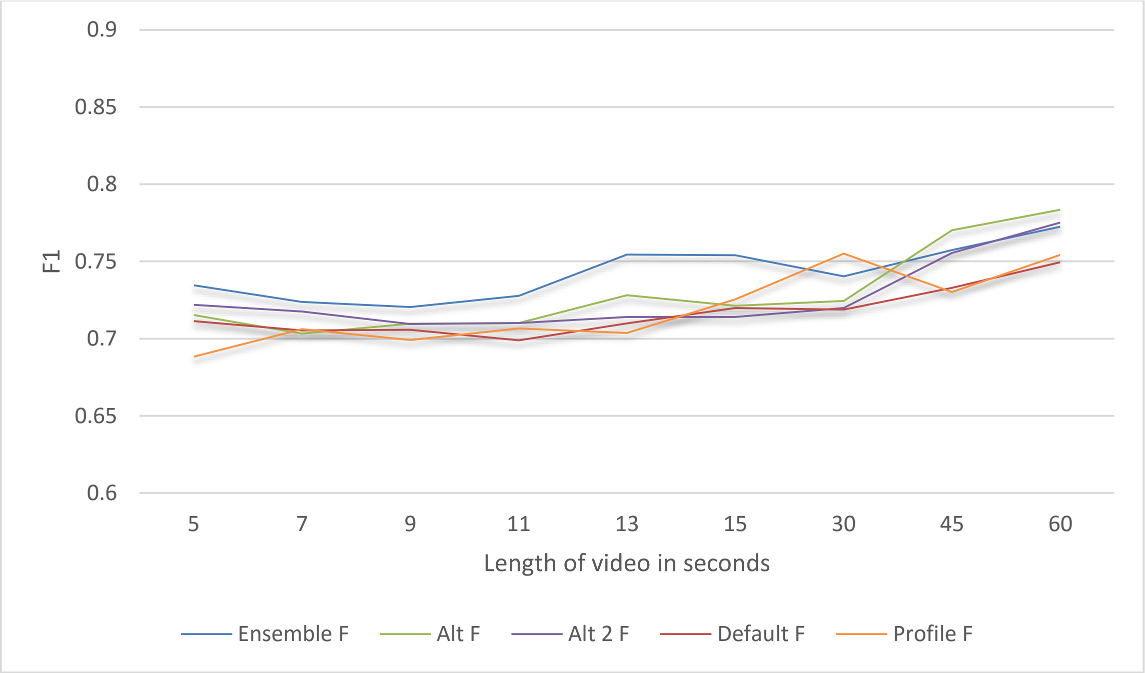

Figure 19 F1 obtained for shorter video segments in dataset B

5.4 Evaluation of Performance of Classifier by Sampling Frames for Face Detection

As already mentioned, the body and head of signers in sign language video content tend to be relatively stationary. Hence, instead of detecting faces in every frame that are used to create unique ROIs for each frame, we tested sampling frames at regular intervals and detected faces in only those frames. The frame rate of the videos in our corpus is 30 frames per second. We tested the effect of sampling rates ranging from 1 (each frame) to 120 (one frame every 4 seconds) for each of the six face detectors.

Figures 20, 21, 22, 23, 24, 25 show the precision, recall and F1 scores achieved for various sampling rates. Other than for alt tree and profile face detectors, applying face detection on sampled frames had only a small negative effect. Consistent with the above findings, classifiers with alt and alt2 frontal face detectors achieved comparable recall and precision to the ensemble of face detectors. The classifier with default face detector achieved better recall in dataset A but performed worse in dataset B. The classifiers with the ensemble of face detectors and the alt and alt2 frontal face detectors maintained their precision and recall until an approximate sampling rate of 20, at which point the performance very gradually decreased and was not consistent.

The sampling rate of 20 would improve the face detection computation time by approximately 20 times without losing much of the precision and recall obtained by detecting faces in every frame. In the next subsection, we will combine this recommendation with the above selection of the alt2 face detector and compare that combination to the original model and discuss the gain in computation time and the impact on precision and recall.

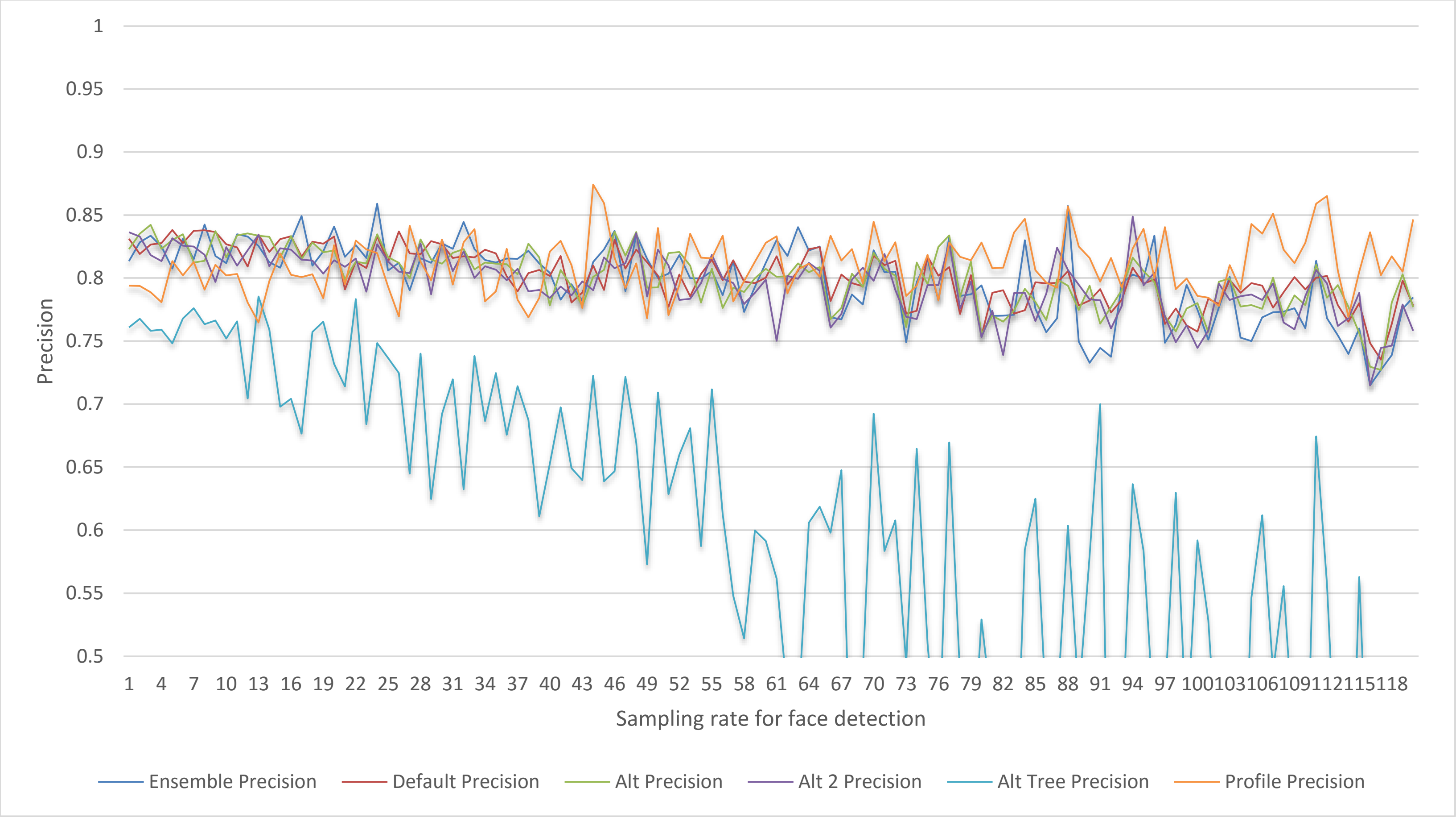

Figure 20 Precision obtained when faces detected at intervals on dataset A

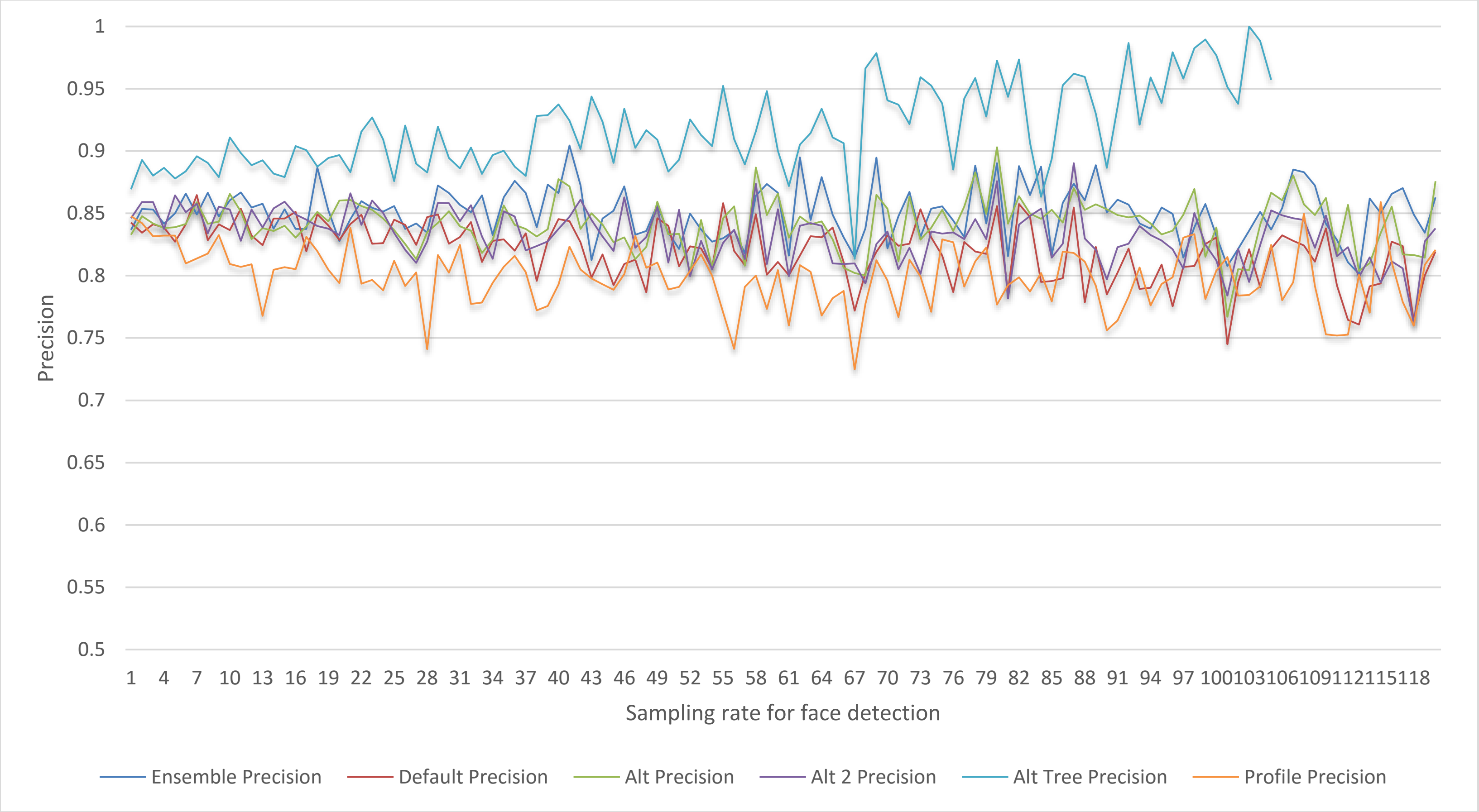

Figure 21 Precision obtained when faces detected at intervals on dataset B

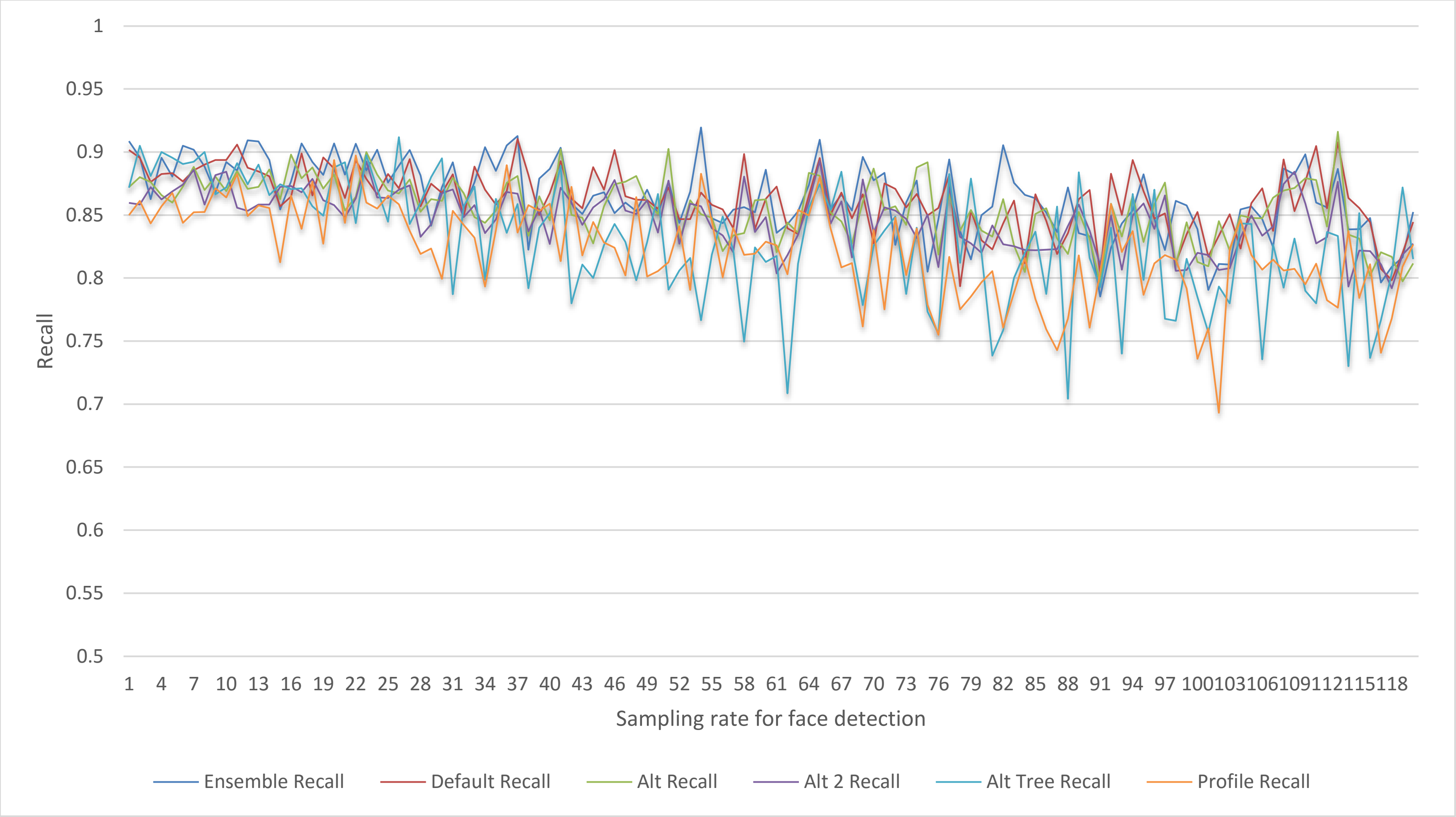

Figure 22 Recall obtained when faces detected at intervals on dataset A

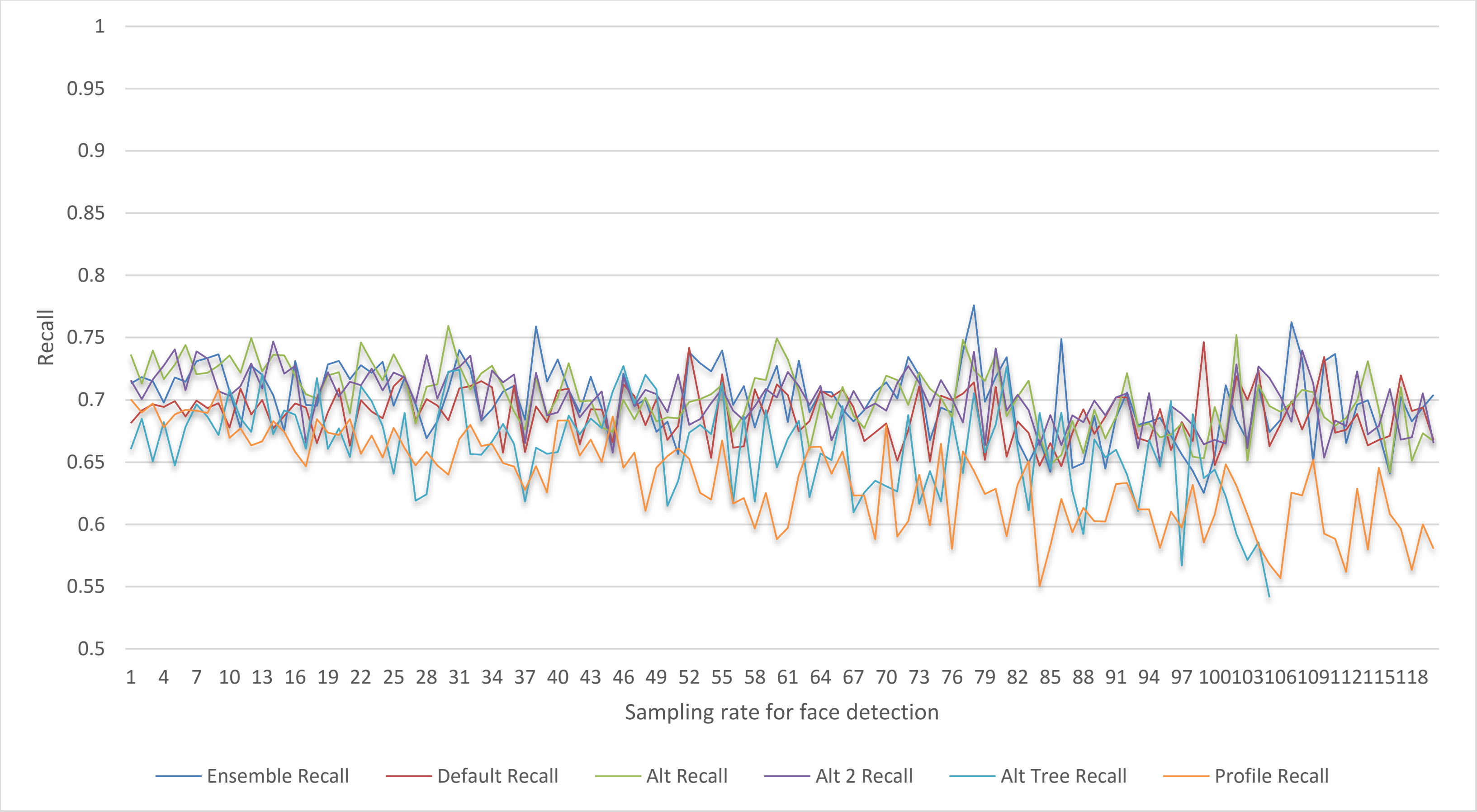

Figure 23 Recall obtained when faces detected at intervals on dataset B

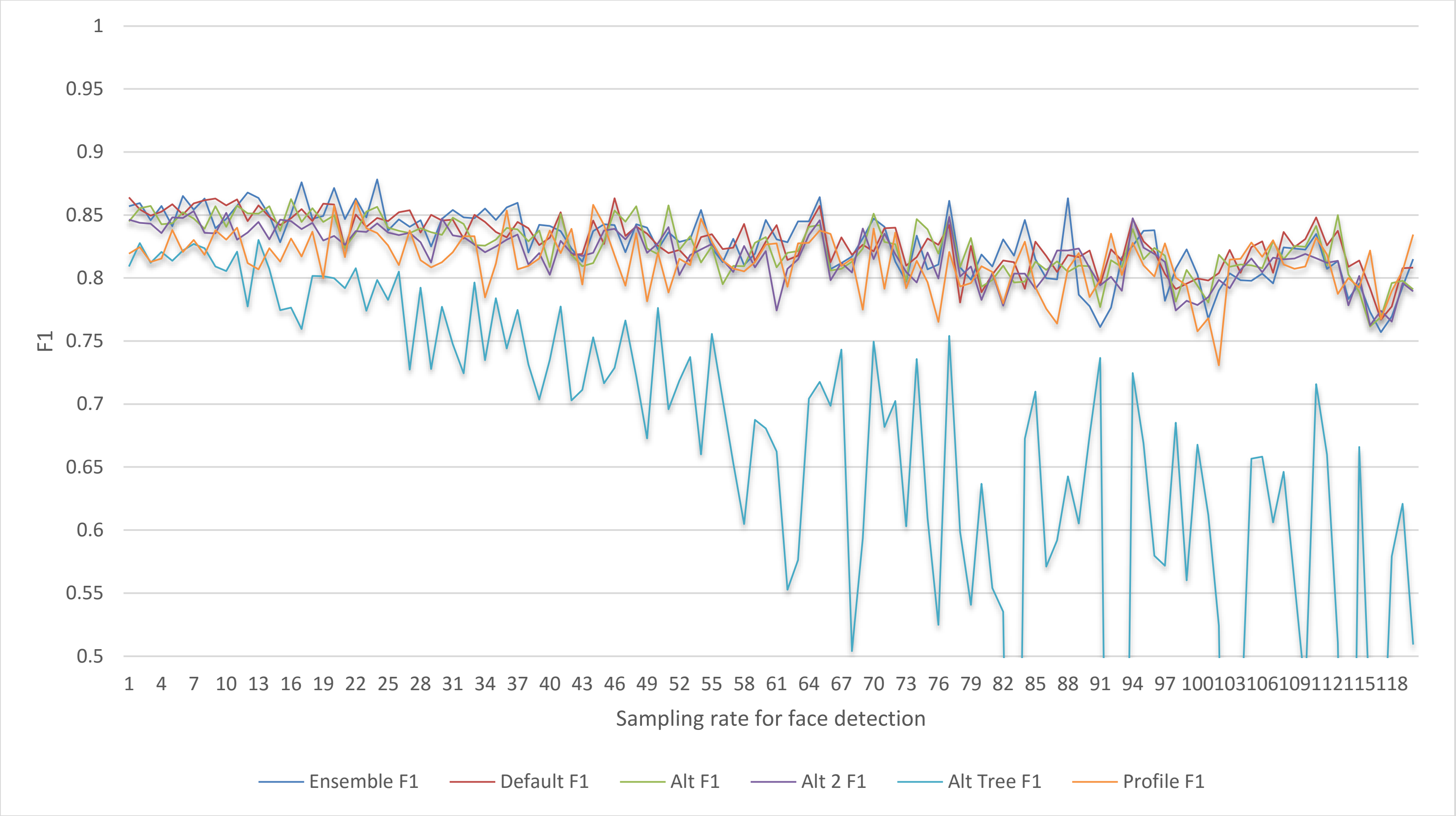

Figure 24 F1 obtained when faces detected at intervals on dataset A

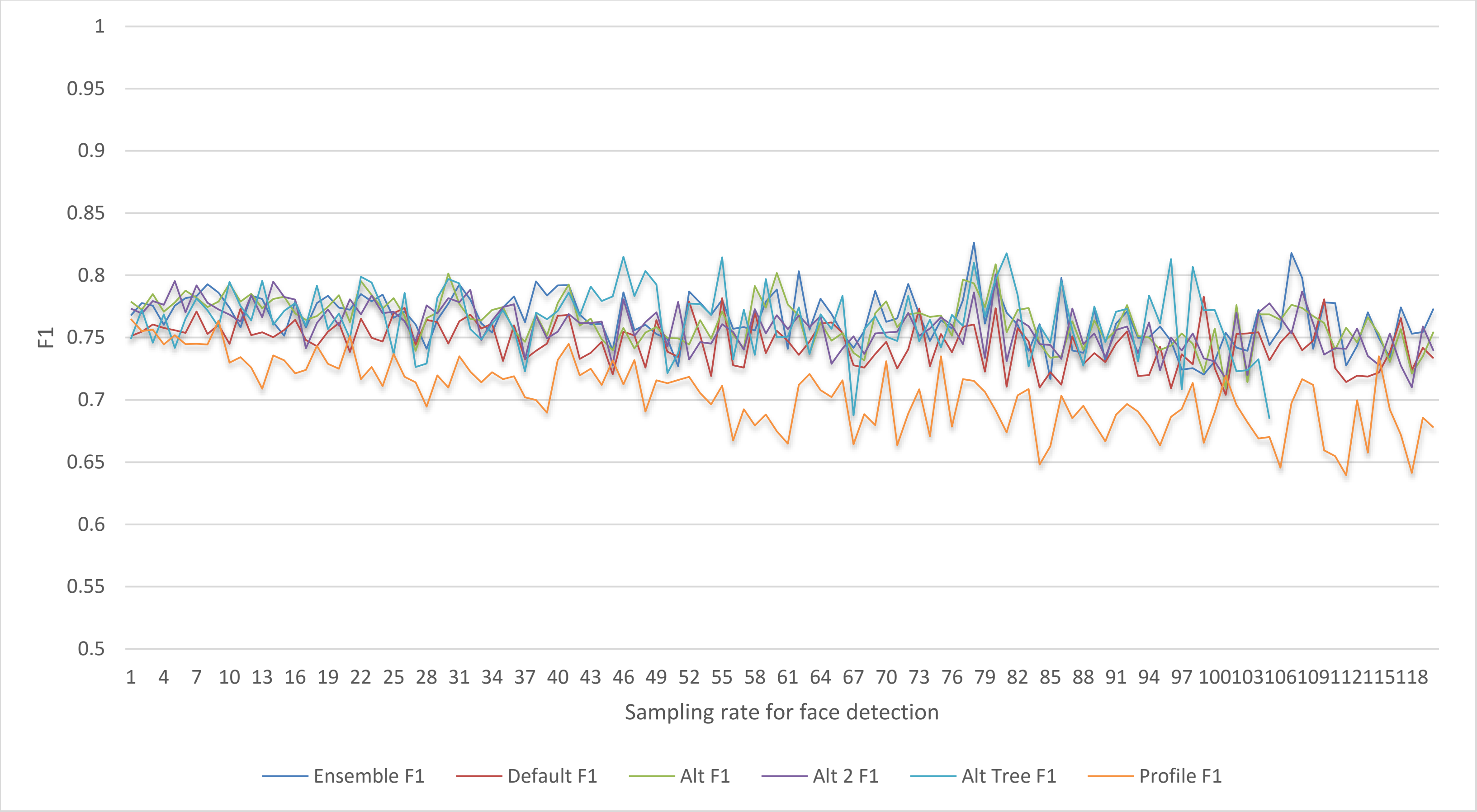

Figure 25 F1 obtained when faces detected at intervals on dataset B

5.5 Comparison of Recommended Model to Original Model

Based on the above findings, it can been summed that frontal face detectors built with a cascade of stage classifiers with gentle adaptive boosting of the weak classifiers are the best performing face detectors for effective classification of sign language videos. It should be noted that only discrete and gentle adaboost cascades are provided in the OpenCV library. Hence, we do not have data regarding the impact of other two boosting techniques.

We chose alt2 frontal face detector as the face detector to compare against the original voting scheme. The other design choices considered are detecting faces at sample rate of 20 with 60 training samples from each of the sign language and non-sign language corpus.

Table 1 Evaluation of recommended configuration on dataset A

Factor	Original model	Sampling model
Average face detection time for a minute video	649 sec	10 sec
Precision	83.55 %	80.98 %
Recall	89.78 %	86.45 %
F1 score	86.46 %	83.47 %

The evaluation of recommended configuration on dataset A can be seen in Table 1. A classifier with the alt2 frontal face detector applied on every 20th frame on videos in dataset A had 3% less precision, recall, and F1 score than did the original voting scheme.

Yet it reduced computation time from 649 seconds to 10 seconds for the one minute segments of video.

Table 2 Evaluation of recommended configuration on dataset B

Factor	Original model	Sampling model
Average face detection time for a minute video	896 sec	31 sec
Precision	85.39 %	83.48 %
Recall	71.29%	71.36 %
F1 score	77.54 %	76.69 %

The same combination of classifier and sampling showed even closer accuracy to the ensemble method when applied on dataset B, which resembles the real world data corpus. Here the computation time was reduced from 896 seconds to 31 seconds for processing one minute segments of video that can be seen in Table 2. This is close to a 30-fold reduction.

Our recommended model did not explore how segment length would affect computation time and accuracy when combined with sampling. Shortening the segment lengths tended to have a more significant impact on performance but is clearly crucial for diarization, a topic for future work.

6. CONCLUSION

6.1 Discussion

This thesis reports on the possibility of reducing the computation time involved in feature extraction time when detecting sign language video. Polar Motion Profiles depend on face detection and background subtraction and the generation of PMPs has to wait until data from both are computed. Although background subtraction is real time, face detection almost takes 10 minutes for detection in a video of one-minute length. Our work was able to bring down the computation time in face detection to the time requirement for background subtraction without greatly impacting precision and recall.

We focused on three approaches to reduce the time in the face detection module. First we assessed the impact on precision and recall when the ensemble of face detectors is replaced with individual face detectors. Then we focused on shortening the length of video segments analyzed. Finally, we focused on changing the sampling rate of the videos which currently stands at each frame. Polar Motion Profiles by their nature were able to detect sign language videos even when faces are detected in frames at regular intervals rather than every frame. The recommended configuration obtained from the three approaches was close to the performance of the original model but reduced the computation time in the face detection module by a factor of 30 for the higher resolution videos in dataset B which are more representative of what we would expect in practice.

The relatively competent performance and great reduction in computation time makes the recommended configuration an ideal canditate for future PMP generation.

6.2 Future Work

There are a wide range of extensions to the current and prior work on sign language detection. A video can have both sign-language and non-sign language content. In this case, the classification of the video as containing sign language is needed. And at the same time, being able to identify the segments of the video containing sign language is important signers those looking for accessible to them. This thesis explored how classification based on shorter segments affected accuracy. Applying this detection approach at intervals across the whole video could imprecisely perform such segmentation and would considerably increase the computation required for each video.

Generalizing this issue, there is a need for techniques to preclassify, or triage, videos based on how much further processing and analysis is required. Videos that are of a single continuous signer do not need segmentation as discussed above. Similarly, videos of landscapes, cats, etc. are clearly not in sign language and, if they could be identified as such with minimal processing, would greatly increase the ability to process the huge quantities of video being uploaded.

This thesis shows that the current classifier can maintain precision at shorter video segments, but background subtraction has to be continuous. This means the current appraoch will not work on videos that are edited to include short segments with different backgrounds. For such videos, alternative techniques for identifying hand motion are needed.

We currently evaluated the system to distinguish sign language videos from non-sign language videos. Sign languages evolved independent from one another and hence just like vocal languages, there needs to be a way for a community to access videos in their particular sign language. The current techniques should be evaluated to find out if they can classify videos based on the sign language being used or if they can distinguish between sets of sign languages.

With the optimizations described in this thesis, the current classifier can perform all the operations in real-time and is ready to be integrated into a digital library system. The classifier when coupled with real information tasks and new content being produced every day will identify more complex scenarios of use that should be taken into consideration to make the classifier more robust.

REFERENCES

1. Brouwer, B. (2014, December 1). YouTube Now Sees 300 Hours Of Video Uploaded Every Minute. Retrieved October 01, 2015, from http://www.tubefilter.com/2014/12/01/youtube-300-hours-video-per-minute/

2. Lienhart, R., Liang, L., & Kuranov, A. (2003, July). A detector tree of boosted classifiers for real-time object detection and tracking. In *Multimedia and Expo, 2003. ICME'03. Proceedings. 2003 International Conference on* (Vol. 2, pp. II-277). IEEE.

3. Monteiro, C., Gutierrez-Osuna, R., & Shipman, F. (2012). Design and evaluation of classifier for identifying sign language videos in video sharing sites. Proceedings of the 14th International ACM SIGACCESS Conference on Computers and Accessibility - ASSETS '12.

4. 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.

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

6. Starner, T., & Pentland, A. (1997). Real-time american sign language recognition from video using hidden markov models. In *Motion-Based Recognition* (pp. 227-243). Springer Netherlands.

7. Assan, M., & Grobel, K. (1998). Video-based sign language recognition using hidden markov models. In *Gesture and Sign Language in Human-Computer Interaction* (pp. 97-109). Springer Berlin Heidelberg.

8. Liang, R. H., & Ouhyoung, M. (1998, April). A real-time continuous gesture recognition system for sign language. In *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on* (pp. 558-567). IEEE.

9. Bauer, B., & Kraiss, K. F. (2002). Video-based sign recognition using self-organizing subunits. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on* (Vol. 2, pp. 434-437). IEEE.

10. Hienz, H., Grobel, K., & Offner, G. (1996, October). Real-time hand-arm motion analysis using a single video camera. In *Automatic Face and Gesture Recognition, 1996., Proceedings of the Second International Conference on* (pp. 323-327). IEEE.

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

12. Somers, G., & Whyte, R. N. (2003, September). Hand posture matching for irish sign language interpretation. In *Proceedings of the 1st international symposium on Information and communication technologies* (pp. 439-444). Trinity College Dublin.

13. Dimov, D., Marinov, A., & Zlateva, N. (2007, June). CBIR approach to the recognition of a sign language alphabet. In *Proceedings of the 2007 international conference on Computer systems and technologies* (p. 96). ACM.

14. Potamias, M., & Athitsos, V. (2008, July). Nearest neighbor search methods for handshape recognition. In *Proceedings of the 1st international conference on PErvasive Technologies Related to Assistive Environments* (p. 30). ACM.

15. Cherniavsky, N., Ladner, R. E., & Riskin, E. A. (2008, September). Activity detection in conversational sign language video for mobile telecommunication. In *FG* (pp. 1-6).

16. Viola, P., & Jones, M. (2001). 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* (Vol. 1, pp. I-511). IEEE.

17. Lienhart, R., & Maydt, J. (2002). An extended set of haar-like features for rapid object detection. In *Image Processing. 2002. Proceedings. 2002 International Conference on* (Vol. 1, pp. I-900). IEEE.

18. Itseez/opencv. (n.d.). Retrieved October 01, 2015, from https://github.com/Itseez/opencv/tree/master/data/haarcascades

19. Friedman, N., & Russell, S. (1997, August). Image segmentation in video sequences: A probabilistic approach. In *Proceedings of the Thirteenth conference on Uncertainty in artificial intelligence* (pp. 175-181). Morgan Kaufmann Publishers Inc..

20. Zivkovic, Z. (2004, August). Improved adaptive Gaussian mixture model for background subtraction. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on* (Vol. 2, pp. 28-31). IEEE.

21. Motion Analysis and Object Tracking. (n.d.). Retrieved October 01, 2015, from http://docs.opencv.org/modules/video/doc/motion_analysis_and_object_tracking.html#backgroundsubtractormog2