Speed-Accuracy Tradeoffs for Detecting Sign Language Content in Video Sharing Sites

Frank M. Shipman, Satyakiran Duggina, Caio D.D. Monteiro, Ricardo Gutierrez-Osuna
Department of Computer Science and Engineering
Texas A&M University
College Station, TX 77843-3112
1-979-862-3216
shipman@cse.tamu.edu

ABSTRACT
Sign language is the primary medium of communication for many people who are deaf or hard of hearing. Members of this community access online sign language (SL) content posted on video sharing sites to stay informed. Unfortunately, locating SL videos can be difficult since the text-based search on video sharing sites is based on metadata rather than on the video content. Low cost or real-time video classification techniques would be invaluable for improving access to this content. Our prior work developed a technique to identify SL content based on video features alone but is computationally expensive. Here we describe and evaluate three optimization strategies that have the potential to reduce the computation time without overly impacting precision and recall. Two optimizations reduce the cost of face-detection, whereas the third focuses on analyzing shorter segments of the video. Our results identify a combination of these techniques that yields a 96% reduction in computation time while losing only 1% in F1 score. To further reduce computation, we additionally explore a keyframe-based approach that achieves comparable recall but lower precision than the above techniques, making it appropriate as an early filter in a staged classifier.

CCS Concepts
• **Information systems** → Multimedia and multimodal retrieval
• **Social and professional topics** → Assistive technologies

Keywords
Sign language detection; signal processing; computer vision.

1. INTRODUCTION
Many applications rely on identifying moments where communication takes place in a recording. For spoken languages, this involves voice detection in an audio signal. For sign languages (SLs), this involves detecting sign language in a video signal. This paper describes and evaluates optimizations for SL detection algorithms aimed at reducing computation without severely impacting accuracy.

Sign language is the medium of communication for many people who are deaf or hard of hearing. With the rising popularity of video sharing sites like YouTube and Vimeo, the volume of SL content available is steadily growing. The SL community often shares these videos through email or posts in social media, passing direct links to the content. However, when an information consumer, rather than an information provider, wants to locate SL content on a particular topic, they must rely on the existence of metadata that identifies both the topic and language used in the video. Unfortunately, studies have shown that metadata is frequently applied inconsistently [5]. Studies of metadata-based access to SL videos on particular topics have found precision rates of 43% [8]. As a result, many in the SL community do not search for SL content. Automatic SL detection would help this situation.

We have been exploring how to identify SL video in video sharing sites. Our earliest work explored the relative value of a set of video features [6] on detecting SL content, but was limited to videos containing a single person facing the camera. In later work [4], we relaxed the constraints to enable detection in videos including multiple visible people and improved the recall but with considerable computational cost. In this later approach, likely signing activity in each frame of a video is modeled using polar coordinates (angle and distance) in a polar motion profile (PMP) which relies on background modeling and subtraction [10] and face detection using Haar features [9]. These combine to create a computationally intensive process. Reduction in computation time is needed to scale the solution to the quantity of videos uploaded to video sharing sites. Once a fast approach to detecting sign language in videos is available, more computationally expensive techniques can be applied to detect which sign language it is.

In this work, we propose techniques to reduce the amount of computation needed to generate PMPs for a video. We evaluated the impact of using alternate face detectors, varying the length of the video segments analyzed, and detecting faces for sampled video frames on the precision and recall of the classifier. These results lead to a recommended combination of these optimizations, which is analyzed in terms of both computation and accuracy. Finally, we explore the potential for computing PMPs from sampled frames without generating a background model – an approach that further reduces computation but comes with a greater reduction in accuracy.

2. 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 American Sign Language into written words. Such a capability would be useful for those not in the SL community to understand the videos in SL and would also

ASSETS'17, Oct. 29–Nov. 1, 2017, Baltimore, MD, USA.

DOI: http://dx.doi.org/10.1145/3132525.3132559

enable search over the content of SL. However, this is a hard problem and not currently applicable to the inconsistent quality of shared videos. Most current research on SL transcription recognizes signs or parts of signs in constrained contexts such as with limited vocabularies, limited signing speed, and/or limitations on the signer's position and the background.

Instead of transcribing SL, our focus is more modest: detecting SL. Related to this problem, Cherniavsky et al. [1] developed an activity detection technique to reduce the bandwidth of mobile video communication when the user is making gestures. But this approach was not meant to differentiate between sign language and other forms of human gesturing and hand/arm motion.

To address this problem, our early work used off-the-shelf video analysis techniques to develop five features that were expected to discriminate between SL videos and other types of videos [6]. The results showed that a measure of the symmetry of movement relative to face was the best feature for classifying SL videos. Expanding on this result, we developed polar motion profiles (PMPs) to model quantity of motion relative to the faces of potential signers [4], enabling videos with multiple people visible simultaneously. This later approach relies on accurate face detection and uses multiple face detectors based on Haar-like features in parallel. Here we explore techniques to speed up the generation of PMPs without negatively affecting the outcome.

Once a video is believed to include sign language, techniques are needed to identify which sign language it is. Gebre et al. [2, 3] tackled the problem of distinguishing between specific sign languages based on video features with a corpus of professionally produced SL videos. We have shown that PMPs can be used to similarly distinguish between certain pairs of sign languages although the accuracy of results are considerably lower for the type of videos found on video sharing sites than they are for the professionally produced SL videos [7].

3. POLAR MOTION PROFILES

Polar motion profiles (PMPs) are a translation and scale invariant measure of the amount of activity, computed on a polar coordinate system centered on each face. In this technique, PMP needs prior information extracted by both face detection and background subtraction modules. Figure 1 shows the steps involved on the computation of a PMP for a given video.

Face detection information is obtained using an ensemble of the individual face detectors provided by the openCV library (*Default, Alt, Alt2, Alt Tree, Profile.*) In this approach, a majority vote is used to determine the face locations. Based on the location of the detected faces, regions of interest (ROI) are defined in every frame of the video. Each ROI is computed to be large enough to capture the arm and hand movements for the person detected. Activity inside each ROI is identified using foreground-background separation based on the adaptive Gaussian mixture model described in [10].

Given the ROI and the foreground pixels, PMPs are computed to represent the activity in a video. The resulting PMP is a feature vector with 460 elements, 360 representing each angular coordinate and 100 elements for radial coordinates. Equation 1 details the computation for the angular coordinate (θ) features; a similar equation is used to obtain the radial coordinate features.

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

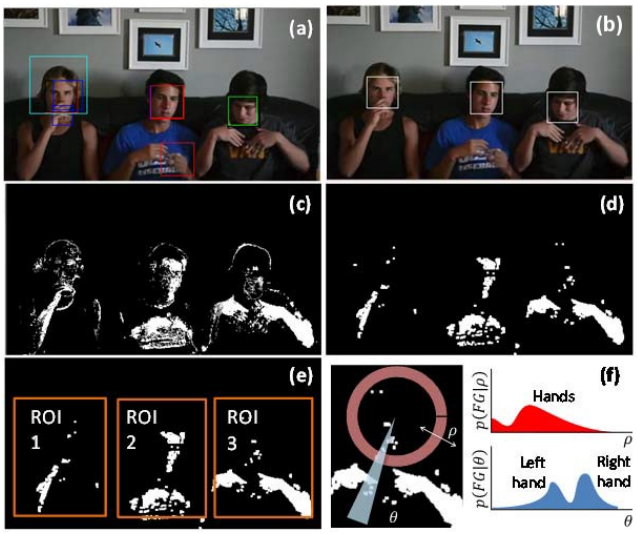

Figure 1. (a) Faces identified by each face detector. (b) Faces from the ensemble of the detectors. (c) Foreground (FG) pixels returned by the background subtraction. (d) Refined FG after morphological de-noising. (e) Regions of interest (ROI) defined for each face detected in the frame. (f) Computation of PMPs for a video frame.

where *R(t)* is the number of ROIs at frame *t*, *T* is the total number of frames in the video, and $PMP_r(\theta,t)$ is the ratio of foreground pixels (*FG*) to the sum of foreground pixels and background pixels (*BG*) at polar coordinate θ for region *r* and frame *t*.

$$PMP_r(\theta,t) = \frac{FG_r\ (\theta,t)}{FG_r(\theta,t) + BG_r(\theta,t)} \qquad (2)$$

The end result is the probability of finding a foreground pixel in the portion of the ROI represented by the element of the PMP feature vector. Principal Components Analysis (PCA) is applied to map the 460 feature elements into six features, which are then passed to a support vector machine (SVM) classifier.

4. OPTIMIZATIONS

There are a variety of approaches available to reduce the computation time. Each of these techniques may detrimentally affect classification accuracy. We initially explored (1) reducing face detection computation time by using a single face detector instead of the ensemble approach, (2) applying face detection to a reduced number of frames, and (3) reducing the length of the video segments used for classification. The initial goal was to match the performance of background subtraction, which 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, then the lag to generate PMPs can be minimized since the two efforts can be executed in parallel.

4.1 Face Detection Computation Time

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 false positive faces may not significantly affect the performance of the classifier, allowing the use of a single face detector to reduce

computation time. To support this hypothesis, we replaced the ensemble of face detectors with each of the individual face detectors, and measured precision, recall, and computation time.

A second approach to reducing computation for face detection examines how performing face detection on every Nth frame affects SL detection. We had reason to believe that sub-sampling would have limited effects since signers' faces and bodies tend to be relatively stationary in many of the SL videos on video sharing sites. The ROIs for the frames between sampled frames were the last computed ROI – that is the ROIs are in static positions for the frames in between frames where face detection is computed.

4.2 Shorter Video Segments

A one minute segment of the original video was used to perform classification in our earlier studies. This means that face detection and background subtraction was performed on each frame in that segment. 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 segment affects SL detection not only informs the design of optimized SL detectors but helps answer to what degree fine-grained diarization (i.e., discriminating segments of a video that include SL from those that do not) can be achieved.

4.3 Recommended Approach

Using the results from the above assessments, we identify a recommended configuration that substantially lowers computation time while not sacrificing precision and recall. We report the computation time and accuracy of this configuration.

5. EVALUATION

The dataset used for assessing the current work is the same as in [4]. It was collected from online video sharing sites and each video was manually labeled as SL or Non-SL video. This corpus contains 111 SL videos and 116 non-SL videos considered related to the SL videos in the corpus by the video sharing site – that is the non-SL videos were returned as results for queries including terms like "sign language" or were recommended as related to known SL videos. The videos in this corpus are not limited to videos with a single potential signer facing the camera as in our earliest evaluations [6, 8] As such, this dataset closely resembles the set of videos that need to be classified in real-world scenarios.

For all experiments, we compare the recall (a measure of false negatives) and precision (a measure of false positives) achieved by our new approaches with the earlier ensemble approach. The F1 score is the harmonic mean of precision and recall. Each value reported is the average of 50 iterations, with each iteration dividing the dataset into training and testing samples randomly.

5.1 Time to Process a Minute of Video

We ran the five individual face detectors in openCV and the ensemble technique on the videos from the dataset to determine their computational time requirements. The length of each video segment was chosen to be one minute and the resolution 240p. The results are shown in Table 1. The focus here is on the relative time requirements as the computation was performed on a quad-core 3.5 GHz Windows PC with 8 GB of memory so may not be indicative of performance on a modern server.

Table 1. Average time for face detectors for 1 minute video.

Ensemble	Default	Alt	Alt 2	Alt Tree	Profile
896 s	94 s	161 s	130 s	107 s	174 s

Using a single face detector instead of the 5-detector ensemble can reduce face-detection time by a factor of 5-10, depending on the particular selection. The time required for the ensemble also indicates that running all five face detectors in parallel takes more time than the sum of time when running them individually.

5.2 Using Single Face Detector

While using a single face detector is much faster, such a choice may negatively affect the accuracy of results. To answer this question, we evaluated classifiers based on each of the five individual face detectors and a classifier that used the 5-detector ensemble. Figure 2 presents the F1 score as a function of the dataset size. As shown, the ensemble is the best face detection technique to be used when training with a limited number of samples and performs well overall. As the number of training samples increase, classifiers with frontal face detectors employing a cascade of stage classifiers and adaptive boosting i.e., alt and alt2, performed best. Overall, the range of F1 scores shows that using a single face detector instead of the ensemble detector does not substantially impede SL detection.

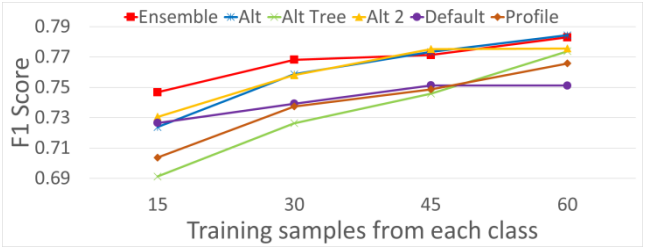

Figure 2. F1 scores for face detectors and training set sizes

Taking both accuracy and computation time into consideration, we chose the alt2 frontal face detector over the alternatives for our recommended configuration.

5.3 Sampling Frames for Face Detection

As already mentioned, the body and head of signers in SL videos tend to be relatively stationary. Hence, instead of detecting faces at each frame, we tested sub-sampling frames at regular intervals and detecting faces at only those frames. The frame rate of the videos in the dataset is 30 frames per second. We tested the effect of sampling intervals ranging from 1 (each frame) to 120 (one frame every 4 seconds) for each of the face detectors. The alt and alt2 face detectors consistently performed better than the other individual face detectors. Figure 3 shows that the ensemble and alt2 face detectors performance was relatively stable up to sampling every 20th frame, at which point the performance gradually decreased and became inconsistent.

This indicates that a sub-sampling rate of 1/20 significantly reduces computation time without losing much accuracy. Additionally, the slow degradation in F1 score as the number of frames sampled is reduced indicates the potential for frame sampling at much longer intervals, as is explored in Section 6.

5.4 Processing Shorter Video Segments

Shortening the length of video segments for feature extraction and classification provides two advantages: first, the computation time for feature extraction can be substantially reduced; second, it enables the identification of shorter segments of SL content in videos (i.e. diarization). We evaluated classifiers with the individual face detectors to find how they performed relative to the ensemble classifier with shorter segments of videos. To select the shorter segment, we took the segment at the center of the first-

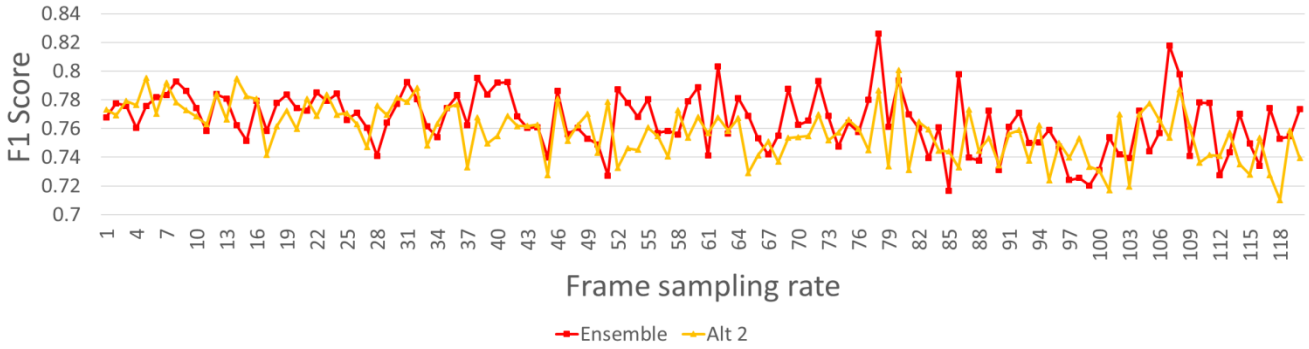

Figure 3. F1 scores for applying face detection to sampled frames. The erratic nature of the results are likely due to the deterministically sampled frames being more or less representative of the video as the starting frame was fixed for each iteration.

minute segment of the full video. For training the classifier, we used 50 samples from each of the SL and non-SL corpus.

Figure 4 shows the F1 scores for the face detectors as the segment lengths vary. The performance degrades as the segment length is decreased. The results reaffirm the selection of alt2 as an appropriate choice for our recommended configuration, but do not identify a shorter segment length for the recommended model discussed next.

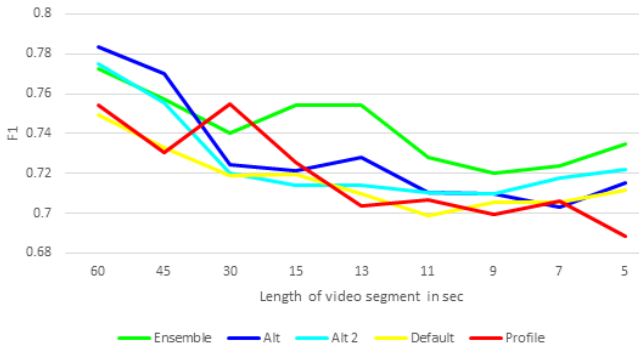

Figure 4. F1 scores for different segment lengths.

5.5 A Recommended SL Detection Approach

Based on the above findings, 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 a sub-sample rate of 1/20 with 60 training samples from each of the SL and non-SL corpus. Table 2 presents the results. Results were about 2% lower on precision, had the same recall, and a 1% lower F1 score when compared with the original approach. Yet these optimizations reduced computation time by 96% (from 896 seconds to 31 seconds) for the one minute segments of video.

Table 2. Evaluation of recommended approach

	Karappa et al. Approach	Recommended approach
Average face detection time	896 sec	31 sec
Precision	85 %	83 %
Recall	71 %	71 %
F1 Score	78 %	77 %

Our recommended approach 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. KEYFRAME-BASED SL DETECTION

Given the huge volume of videos that are uploaded to video sharing sites every day, the optimizations above improve the applicability of SL detection to this context but do not solve the problem. What is needed is a pre-processing stage that identifies videos that warrant such analysis. Towards this end, and based on the results from frame sampling for face detection, we developed a keyframe-based SL detector that uses the same PMP features as before but computes these features for a small set of frames within the video rather than computing them for each frame in the video. Because this approach does not examine every frame, foreground pixels must be identified by alternative means.

As with the above approaches, the keyframe-based approach relies on face detection to identify a region of interest. Within each region of interest, the foreground model is generated via frame subtraction with the prior frame – i.e. the prior frame is used as the background model. More specifically, foreground pixels in a keyframe are the result of applying a median filter to the two greyscaled frames, subtracting the frames, then thresholding the results. Finally, image opening is applied to remove noise among the identified foreground pixels. The resulting foreground image is then used to generate PMPs which are passed to the SVM for classification.

The advantage of a keyframe-only approach when compared to the above techniques is that only a small number of frames have to be considered. The limitation of using such an approach is that, without a dynamic background model and without information about the intermediate position of the potential signers' arms, the foreground identified is likely to include change within the field of view not associated with hand/arm motions. Changes in body position, lighting, etc. could overwhelm the foreground data associated with human gestures, creating too much noise for accurate SL detection.

To characterize the performance of keyframe-based SL detection, performance of the resulting classifier was examined as the number of keyframes selected from the video varied. These results were examined for both 30 second and 60 second video segments. The keyframes included in the analysis were evenly distributed in the video segment. Thus, if 5 frames are chosen from a 60 second video segment, the frames are chosen at the 0, 15, 30, 45 and 60 second points for generating the PMPs.

The results in terms of F1 scores using the same corpus of SL and non-SL videos as before is shown in Figure 5. The F1 scores of

the classifier are clearly better with the longer (60 second) source videos. Since the longer video for the same number of frames doubles the time between examined frames, this indicates that changes in background are not the main cause of incorrect classifications for this approach. Overall performance in terms of the F1 score leveled off when at least 10 keyframes were selected for analysis at about 71% for the longer videos.

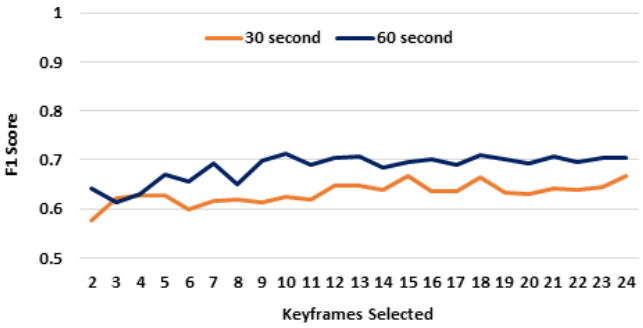

Figure 5. SL vs. Non-SL F1 score for 30 and 60 second videos.

When the results for this approach are compared to the ensemble approach described in [4], the keyframe-based approach achieves similar or slightly better recall but at a considerable loss in precision. This is shown in Table 3.

Table 3. Comparison of keyframe and Karappa approach.

	Karappa et al. Approach	Keyframe Approach
Precision	85 %	69 %
Recall	71 %	74 %
F1 Score	78 %	71 %

While the performance of the keyframe-based SL detector is lower in terms of precision and F1 score, it reduces the number of frames extracted and analyzed from 1800 to 10. This makes the classifier much faster. Indeed, in our studies with this approach, feature generation is reduced to such a point that frame extraction can become a bottleneck depending on how the video is encoded.

In terms of performance within a staged classifier, given the relatively strong recall, there is the potential to include a frame-based classifier as an initial filter. Videos identified as likely including SL content by this filter could then be more carefully analyzed by a second, more powerful SL detector.

7. DISCUSSION AND FUTURE WORK

We report on the possibility of reducing the computational time involved in feature extraction 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 could take more than 10 minutes to compute for a video of one-minute length, using the ensemble approach. We were able to bring down the computation time for face detection to at or below 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 system performance when the ensemble of face detectors is replaced with individual face detectors. Then we focused on detecting faces on sampled frames of the videos rather than for each frame. Finally, we focused on shortening the length of video segments analyzed. The recommended configuration obtained was close to the performance of the original model and reduced the computation time in the face detection module by 96%.

Further exploration of a frame-based classifier removed the need for background modeling. While this resulted in a considerable loss in precision, the frame-based classifier had a relatively high recall with computation costs at the level required for frame extraction. This combination makes it a good candidate as an early filter in a staged classifier. We are investigating whether a such a staged classifier can both improve accuracy and reduce the average computation time for our current single stage classifier.

The above results show that PMP-based classifiers perform well on the types of videos uploaded to video sharing sites, but rely on background subtraction to identify signing activity. The current approach 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. Another class of video where the current approach may fail are videos that have both SL and non-SL segments. For such videos, the classification of the video as containing sign language is needed. At the same time, the system must also identify the SL segments of the video for signers looking for accessible content (i.e. diarization.)

8. REFERENCES

[1] Cherniavsky, N., Ladner, R., Riskin, E. 2008. Activity detection in conversational sign language video for mobile telecommunication. *IEEE Int. Conf. on Automatic Face and Gesture Recognition*, 1–6.

[2] Gebre, B., Wittenburg, P. and Heskes, T. 2013. Automatic sign language identification. *2013 IEEE Int. Conf. on Image Processing*, 2626–2630.

[3] Gebre, G., Crasborn, O., Wittenburg, P., Drude, S. and Heskes, T. 2014. Unsupervised feature learning for visual sign language identification. *Proc. of Annual Meeting of the Association for Computational Linguistics*, 370–376.

[4] Karappa, V., Monteiro, C., Shipman, F. and Gutierrez-Osuna, R. 2014. Detection of sign-language content in video through polar motion profiles. *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 1290–1294.

[5] Marshall, C. 2009. No Bull , No Spin : A comparison of tags with other forms of user metadata. *Proc. of the ACM/IEEE-CS Joint Conf. on Digital Libraries (JCDL '09)*, 241–250.

[6] Monteiro, C., Gutierrez-Osuna, R. and Shipman, F. 2012. Design and evaluation of classifier for identifying sign language videos in video sharing sites. *Proc. of the ACM Conf. on Computers and Accessibility*. (2012), 191–198.

[7] Monteiro, C., Mathew , C., Gutierrez-Osuna, R. and Shipman, F. 2016. Detecting and Identifying Sign Languages through Visual Features, *Proceedings of IEEE International Symposium on Multimedia 2016*, 2016, pp. 287-290.

[8] Shipman, F., Gutierrez-Osuna, R. and Monteiro, C. 2014. Identifying Sign Language Videos in Video Sharing Sites. *ACM Transactions on Accessible Computing*. 5, 4 (Mar. 2014), 1–14.

[9] Viola, P. and Jones, M. 2001. Rapid object detection using a boosted cascade of simple features. *Computer Vision and Pattern Recognition, CVPR 2001*, I--511.

[10] Zivkovic, Z. 2004. Improved adaptive Gaussian mixture model for background subtraction. *Proc. of the 17th Int. Conf. on Pattern Recognition,* 28–31.