DEVELOPING A SIGN LANGUAGE VIDEO COLLECTION VIA METADATA AND VIDEO CLASSIFIERS

A Dissertation

by

CAIO DUARTE DINIZ MONTEIRO

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

DOCTOR OF PHILOSOPHY

Chair of Committee,	Frank M. Shipman III
Co-Chair of Committee,	Ricardo Gutierrez-Osuna
Committee Members,	Richard Furuta
	Ergun Akleman
Head of Department,	Dilma Da Silva

August 2018

Major Subject: Computer Science

ABSTRACT

Video sharing sites have become a central tool for the storage and dissemination of sign language content. Sign language videos have many purposes, including sharing experiences or opinions, teaching and practicing a sign language, etc. However, due to limitations of term-based search, these videos can be hard to locate. This results in a diminished value of these sites for the deaf or hard-of-hearing community. As a result, members of the community frequently engage in a push-style delivery of content, sharing direct links to sign language videos with other members of the sign language community. To address this problem, we propose the Sign Language Digital Library (SLaDL).

SLaDL is composed of two main sub-systems, a crawler that collects potential videos for inclusion into the digital library corpus, and an automatic classification system that detects and identifies sign language presence in the crawled videos. These components attempt to filter out videos that do not include sign language from the collection and to organize sign language videos based on different languages. This dissertation explores individual and combined components of the classification system. The components form a cascade of multimodal classifiers aimed at achieving high accuracy when classifying potential videos while minimizing the computational effort.

A web application coordinates the execution of these two subsystems and enables user interaction (browsing and searching) with the library corpus. Since the collection of the digital library is automatically curated by the cascading classifier, the number of irrelevant results is expected to be drastically lower when compared to general-purpose video sharing sites.

The evaluation involved a series of experiments focused on specific components of the system, and on analyzing how to best configure SLaDL. In the first set of experiments, we investigated three different crawling approaches, assessing how they compared in terms of both finding a large quantity of sign language videos and expanding the variety of videos in the collection. Secondly, we evaluated the performance of different approaches to multimodal classification in terms of precision, recall, F1 score, and computational costs. Lastly, we incorporated the best multimodal approach into cascading classifiers to reduce computation while preserving accuracy. We experimented with four different cascading configurations and analyzed their performance for the detection and identification of signed content. Given our findings of each experiment, we proposed the set up for an instantiation of SLaDL.

DEDICATION

To my beloved family, my wife, my daughter, and our dog. To my parents and brother.

ACKNOWLEDGEMENTS

First of all, I would like to thank Dr. Frank Shipman. He was an amazing advisor all the time, always available to guide and support me whenever needed. Doing a doctorate is certainly a hard task, but his calming and constant presence made this whole experience much more enjoyable.

I would also like to thank my co-advisor, Dr. Ricardo Gutierrez-Osuna. Like Dr. Shipman, he helped me out since my first contact with this project as an undergraduate student back in 2011 and always provided me valuable feedback.

Thank you as well to my committee members, Dr. Richard Furuta and Dr. Ergun Akleman, for their support throughout the course of this research.

Thanks also to the friends that I made here in College Station and the wonderful lab mates I had over the years, Sam, Josh 1.0, Gabe, Josh 2.0, Satya, Christy Maria, and Aishwarya.

A special thanks to my wife for the immense support and love during all these years. She was the safety net that caught me when I failed and the encouragement I needed to always keep going. Also, thanks to my little daughter, she is too young right now to understand it, but she is the reason I give my total effort every day to be not only a better professional/student but a better person as a whole. Concluding the acknowledgments for the ones who share a home with me, thanks to Kobe, his wiggling tail and clumsy hugs every time I opened the door after a long day of work always reinvigorated my spirit and put a smile on my face.

Finally, I would like to thank my whole family, the family I have since I was born and the one I gained after marriage. To each one of you my gratitude and appreciation for all the love and care, all of these years.

CONTRIBUTORS AND FUNDING SOURCES

This work was supported by a dissertation committee consisting of Professors Frank Shipman (advisor), Ricardo Gutierrez-Osuna (co-advisor), and Richard Furuta of the Department of Computer Science and Engineering, and Professor Ergun Akleman of the Department of Visualization.

The PMP algorithm and one of the datasets used in the experiments were part of Virendra Karappa’s MS thesis work, presented in 2014, of the Department of Computer Science and Engineering. All other work conducted for the dissertation was completed by the student independently.

Graduate study was in part supported by a scholarship from CAPES foundation.

NOMENCLATURE

ASL	American Sign Language
BSL	British Sign Language
LDA	Latent Dirichlet Allocation
LSF	French Sign Language
LSM	Mexican Sign Language
NMF	Nonnegative Matrix Factorization
PMP	Polar Motion Profile
SL	Sign Language
SLaDL	Sign Language Digital Library
SVM	Support Vector Machine
TF-IDF	Term Frequency – Inverse Document Frequency

TABLE OF CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

Page

1. INTRODUCTION

The increasing popularity of video sharing sites like YouTube and Vimeo, coupled with cheap video recording hardware, have enabled content creation by virtually everyone with internet access. Generally, people looking to consume content from these video sharing sites have two ways of finding the desired content. They can follow direct pointers to specific resources, regardless of how they obtained these pointers, or they can browse and search through the video sharing site collection seeking for content that matches their needs. The latter approach is difficult for certain types of content needs. This work explores supporting location of video content accessible in Sign Language (SL).

In the U.S, approximately 0.5% of the population is deaf or hard-of-hearing, using the definition that they can at most understand utterances shouted at their best ear [1]. Mainly for those who grew up with sign language as their primary language, the surrounding written language is not a proper substitute, as a previous study has shown that half of the deaf or hard-of-hearing 17-18 years old population had a lower reading level than the typical fourth grade student [2]. Therefore, for a considerable portion of the SL community, being able to easily locate and access signed content is extremely important.

Searching on video sharing sites works by the user providing a textual query that is then compared and matched against video characteristics represented as metadata. Apart from the inherent problem of retrieval based on user-provided tagging and metadata [3], finding relevant content in sign language is made additionally difficult because metadata must indicate not only topic, but language as well. Prior work has examined the extent of this problem. In an early study examining this issue, Shipman et al. have found that just about 46% of returned videos from queries combining SL terms (e.g. ASL, “sign language”) with specific topics were actually on topic and in sign language [4]. A later study by Karappa found that just 13% of the results were on topic and in sign language when considering more varied and less time-constrained topics [5]. It is important to note that in both studies, American Sign Language (ASL) was the only language analyzed. As with verbal and written languages, there are many different types of sign languages, with little to no mutual intelligibility between them. Also, there is no one-to-one mapping between the sign language of a population and their surrounding written language. For example, ASL and British Sign Language (BSL) are both used in countries where English is the official written language, yet they are independent sign languages, each one with its own vocabulary, syntax, and grammar [6].

In this work, we propose a framework to address the accessibility problem for sign language content. We have designed a system capable of maintaining a collection of Sign Language content, taking the form of a portal [7], the Sign Language Digital Library (SLaDL). The size and rate of growth of video sharing sites like YouTube make it infeasible to simply process their entire corpus. Because of that, our system uses a guided crawler mechanism to locate potential new SL content. Candidate videos are identified by the crawler from two sources, user-generated query results and related videos from SL videos already present in the digital library (e.g. videos from the same channel, videos suggested by YouTube, videos from subscribed channels, etc.).

Adding a candidate video to the collection is dependent on its automatic classification as a sign language video. Due to the large quantity of videos to be considered, computational load should be minimized to avoid bottlenecks. Therefore, our framework uses a staged classifier based on a combination of metadata and video content features. This staged approach aims to avoid full video content processing whenever possible, applying computationally cheaper techniques initially and reserving more expensive techniques to a much smaller set of potential SL videos.

Features extracted from metadata are often much more efficient to compute than visual features. Thus, early stages of our classifier focus on these sets of features, considering metadata both as a structured and unstructured resource. With structured metadata we consider not only which text can be found in the metadata, but also where they are located. This structure allows us to learn decision rules inferred from previously processed video data. Given a set of decision rules, classification of a new video can be very efficient. A good example of this approach is the case of a channel devoted to sign language content. After processing several videos from this channel, the system would then be able to infer a decision rule and automatically classify other potential SL videos from this channel as being in sign language, without needing to perform any feature transformation.

When structured metadata does not provide sufficient evidence for a final decision, we consider textual fields of metadata, using text analysis to obtain a feature representation of video metadata and then applying the same classification algorithm used in our video based computations.

For the cases when metadata features alone are not able to provide an accurate classification for the video, visual features are then computed. In prior work, Karappa proposed the extraction of Polar Motion Profiles (PMP) to classify videos between sign language and non-sign language [8]. Further work by Monteiro has shown that these features have also the potential to distinguish among different types of sign language [9]. However, extracting these features from video is computationally intensive. To cope with that, we incorporated techniques to reduce the average video processing time [10] into our staged pipeline, processing just a few keyframes of the video at first, and moving to more computationally demanding tasks if needed.

We explored three research questions in our work, each one related to a main component of our framework: how different crawling strategies impact the variety and size of the SLaDL corpus; how different parameters of the staged classifier impact the average time required to process a video; and which combination of video and metadata features yields to better results when detecting and identifying SL videos.

It is important to note that while the work proposed in this dissertation explores the classification of sign language videos, the techniques used in the framework are expected to be generalizable to other domains, making the framework a blueprint for instantiating community-oriented video collections.

The rest of this work is organized as follows. section 2 presents a review of related work. This is followed by a discussion of the problems and issues tackled by this project in section 3. section 4 describes our work and evolution of the SL classifier, and in section 5 we explain the current SLaDL system architecture and the components of the cascading classifier framework. The experiments and our investigated research questions are discussed in section 6, followed by the results and our findings in section 7. In the last section we draw our conclusions and cast light into future work opportunities in this project.

2. RELATED WORK

Our proposed framework for the development of community-oriented collections, more specifically a sign language collection, was built upon several research areas: sign language video processing; multimodal video categorization; crawling mechanisms; and community-oriented collections.

2.1 Sign Language Video Processing

Video analysis and processing has been used to tackle different problems within the sign language domain. A great amount of work has been done on the task of translating sign language content, enabling communication between signers and non-signers. Transcribing SL content into textual language is a very difficult problem, especially when considering real-world pre-recorded videos with no constraints on the signer and a wide range of recording qualities and scenarios. In one of the earliest studies of SL transcription. Starner [11] used Hidden Markov Models (HMM) to classify 40 different words, aside from the small vocabulary, another limitation of their work was that each individual signer required a new model to be trained. Another common technique is to use image similarity measures to identify hand postures and then identify gestures. In [12], a dual camera setup is used to capture video and then hand postures are identified matching silhouettes against a database of 3D rendered models. This work has also focused on specific words and added the constraint of recording using two cameras. Further work have been done based on handshape image similarities [13], [14]. Besides being limited to small vocabularies, all these approaches have focused on one aspect of sign language communication, handshape, while in reality, each sign is comprised by five major components: handshape, position, palm orientation, trajectories, and facial expression. Towards that end, Caridakis et al. [15] proposed a framework combining many of these features, although it appears it was never instantiated or evaluated. While recognizing words alone is already a challenging task, it does not bridge the gap in communication between the SL community and non-signers. Therefore, some work has focused on recognizing not only words, but entire sentences [16], [17]. When considering sentences, researchers do not discuss whether the proposed techniques require constraints on the speed of expression. More generally, the majority of the SL translation efforts make use of SL video that includes signing at a slower speed than what is normal for fluent signers.

More related to our work is the task of detecting or identifying sign language in video. In [18], an activity detection algorithm is used to reduce bandwidth consumption on sign language mobile telecommunication. However, this work does not focus on distinguishing sign language gestures from other types of gestures. More recently, Gebre et al. [19], [20] have tackled the problem of identifying specific types of sign language. While their method was focused solely on computed video features (e.g. skin detection, hu moments [21], etc.), positive reported results were obtained when classifying videos from the Dicta-Sign [22] corpus, a professionally produced dataset targeted to be used in sign language research with very high video quality, a static background, and relatively slow signing. Thus, it is unclear to what extent their approach is feasible to be used with user-generated content. We are not aware of any work, other than our previous efforts, that discuss sign language detection and identification when dealing with unconstrained pre-recorded videos of the type found within video sharing sites like YouTube.

While many of the above techniques do not solve the problem we address in our work, namely that of aiding members of the SL community to locate SL content, they can be applied in other contexts. For example, they can support research focusing on aspects of sign language learning [23], [24] and on the ability to support signer to non-signer communication [25], [26].

2.2 Multimodal video categorization

In this work, we explored the potential for multimodal categorization in the context of creating an SL video collection. Multimodal classifiers can be applied to a range of problems and have been used to support video summarization and image classification [27], [28]. Due to the inherent difficulty of extracting semantics purely from video signals (this can be a hard task even for human subjects), video features are usually coupled with some other media type into a multimodal classifier [29]. One of the most common approach is to combine the computed video features with textual features (e.g. title, description, tags, etc.) into a single feature space and perform the classification task on this new space, this technique is often called *early fusion* [30]. Use of early fusion has the advantage of representing the samples into a single feature space and requiring a single classification effort, but this comes at the expense of generalizability and reusability. It is unlikely that a particular feature set combination generate for one problem will work with the same amount of success for other problems. To overcome this limitation, researchers started to experiment with *late fusion* [31]. This approach combines the assessments from different modes in the semantics space rather than in a feature space; it works by combining the learned labels or concepts from each mode into a final interpretation. This higher level of abstraction makes late fusion strategies more reusable. In [32], the authors compared the performance of early and late fusion techniques when semantically classifying videos combining visual and textual features. They found that while most of the categories had a higher accuracy using late fusion, for the cases when early fusion worked better, the difference between the techniques was higher. This further strengthens the tradeoff between generalizability and performance found in these two techniques.

Independent of the combination approach used, most of the related work have treated textual features as an unstructured resource [29], [33]. However, when treating textual data as a raw resource much of the information can be lost and ambiguities can arise (e.g. the same terms can have very different meanings depending on the metadata field that they are presented). To avoid this, some prior work treats each field of metadata separately or combines sub-groups of them. In [34], metadata and visual features are late fused to classify videos in different categories.

2.3 Crawling mechanisms

Because of the dynamic and ever-growing nature of web resources, different crawling mechanisms have been developed to support web based information retrieval systems. These same characteristics can also be found on video sharing sites collections, making it a domain where crawling can be really useful [35]. Crawlers were first developed to retrieve web pages present in the Internet, maintaining an index with the resources found that could then be used by other applications, such as search engines [36]. Generally speaking, there are two types of crawling techniques: (1) General-purpose crawlers, where the goal is to perform an exhaustive exploration on the search space and be as comprehensive as possible; and (2) Heuristic-based crawlers, where the crawlers tries to locate new resources not blindly, but guided by some heuristic, usually scoring the resources found [37]. Considering the problem we address in this dissertation, the latter is more suited to tackle our needs, due to the small size of relevant resources when compared to the whole collection.

Heuristic-based crawlers can range from simple approaches using a naïve best-first based on textual distance metrics, to more complex ones that includes other types of information like depth bound [38], resources taxonomy [39], and context [40]. Common to these approaches is the use of an evaluation metric to rank unvisited resources and decide their visiting order.

2.4 Community-oriented collections

Digital libraries, such as the Digital Library for Earth System Education (DLESE) [41] have aimed to support specific communities through components and subsystems for generating and curating resources in the collection. User involvement into the collection maintenance can take the form of different actions. Users should be allowed or responsible for direct or indirect inclusion of new resources into the digital library. Community members can locate and register new data providers for the collection and they can curate and maintain the collection through feedback mechanisms or direct manipulation. While this dissertation will not explore community feedback directly, the design of the classifier aims to enable such feedback in the future.

2.5 Summary

While there is plenty of work on sign language video processing, most of the efforts have focused on the issue of sign language translation. Little attention has been given to improving sign language content accessibility, the issue we are trying to tackle in this dissertation. To improve SL video accessibility, we make use of multimodal classifiers. These types of classifiers have been thoroughly discussed in the literature but no previous work has focused on the domain we are investigating. Additionally, we explore alternative applications of multimodal features within a staged classifier, aimed at reducing computational overhead.

The proposed system takes the form of an ever-growing collection of SL videos, and while previous works have analyzed ways for user involvement into collection maintenance, our initial system relies on the users indirectly including new videos into the collection and the use of automated classification to validate potential videos.

3. PROBLEM STATEMENTS AND ISSUES

As previously discussed in the Related Work section, many of the current research efforts related to SL video are focused on the task of transcribing sign language, with very few of them aimed at improving the accessibility of existing SL content. Also, most SL video research is not concerned with the issues of efficiency or video quality, important aspects when considering application to existing videos posted on video sharing sites.

3.1 Accessing Sign Language content

First of all, it is important to define what it means for a video to be a SL video. For the scope of our system, a video is considered to be in sign language when most of its content is intelligible to people who sign in the language found in the video. Videos where sign language happens incidentally (e.g. a news video talking about sign language) are not useful for the community and thus, not the current target of our work.

Seeking sign language content on the web is not a simple task, as a matter of fact, most members of the deaf and hard-of-hearing community do not even bother trying to locate new content. Instead, they rely on direct pointers passed through ad hoc mechanisms (e.g., instant messaging, email, social networks, etc.). To quantify this problem, we conducted a study to analyze how hard it is to locate videos in sign language for particular topics in video sharing sites [4]. For the study, we focused on finding ASL videos on YouTube with topics from the top ten news queries of 2011 in Yahoo! [42]. Top 20 results for each query were then manually coded as on topic/not on topic and in SL/not in SL. Table 1 presents the results. As is observable, out of the 110 videos returned by the queries only 46% were both on topic and in sign language, while 8% of the videos where totally unrelated, being not in sign language and about other topics.

Table 1. Number and percent of results in/not in sign language and on/off topic based on 2011 top ten news queries on Yahoo!.

	In SL	Not in SL	Total
On Topic	50 (46%)	27 (25%)	77 (70%)
Not on Topic	24 (22%)	9 (8%)	33 (30%)
Total	74 (67%)	36 (33%)	110 (100%)

In a follow-up study [5] using broader and less time limited topics, the results again were not satisfactory. While the percentage of returned videos in sign language was greater, the precision when considering videos to be in both sign language and on topic was drastically lower; 13% compared to the previous 46%. This indicates that there is a clear difficulty when trying to locate sign language resources concerning particular topics on general purpose video sharing sites. This limitation arises from the inherent ambiguity of query terms (e.g. ASL has different meaning in different domains), the overall quality of user provided metadata [3], and how video sharing sites identify and rank videos in response to textual queries.

3.2 Types of Sign Language videos posted on Video Sharing Sites

YouTube and other video sharing sites are idiosyncratic when considering the types of SL videos that one can expect in these platforms. For example, videos depicting signed translations of music became quite successful, encouraging content creation of this kind. Nowadays, it is pretty common to find signed versions for popular songs. Different types of videos present different challenges for the classification task, below are some typical videos that are popular on video sharing sites and their unique characteristics.

- **Educational videos –** This type of videos include any sort of teaching or learning SL. Usually, videos in this category have an above average metadata, making it easier to locate them through search queries. Due to its educational purposes, signed utterances are usually performed at a much slower speed than a natural SL conversation and with bigger gaps between one sign and the next, making it harder for video features to identify the language. On the other hand, backgrounds in these videos are reasonably static, avoiding noisy artifacts in the video processing.

- **Storytelling –** Videos where usually one signer tells a story to the audience. For these videos, metadata quality can wildly vary between different posting users. While the background of the video often is static, depending on the recording location, background activity might be present. Video quality also varies a lot in this type of videos, given that it is more of a homemade type of video than a professional one.
- **Video log (Vlog) –** This type shares much of the same characteristics of storytelling videos. Similar to the previous category, usually one person is responsible for all the signing, however, instead of telling a story, vlog videos are centered around signers' previous experiences or opinions. They are a more personal expression, indicating that recording and background qualities can drastically change from one poster to another. Partial or incomplete metadata is also very common for this type of video.
- **Music interpreting –** As previously mentioned, for this type of videos, one or more signers transcribe the lyrics of a music to a specific sign language, using their bodies to also express the music rhythm and flow as accurate as possible. This artistic type of video presents a set of distinguishing characteristics when compared to the previous types. Even though many of these videos are personal expressions, overall recording quality is usually greater than vlogs or storytelling videos. In this type, videos usually contain more than one take (indicating sudden background changes) and signers move around a lot following the rhythm of the music. These two aspects make it hard to process these

videos based solely on video features. Due to all the movement and background changes, noisy artifacts are certainly a problem when processing the video. Luckily, metadata in this type of video is usually fairly complete and accurate. Given the effort required to produce this type of content, it is expected that the posters would carefully annotate the metadata of the video when uploading it to video sharing sites.

- **Picture-in-picture (PIP)/Bilingual videos –** This category includes videos where there is any sort of real-time captioning in sign language. Using traditional search queries, this is probably the hardest type of video to search for. Metadata for these videos seldom indicate the presence of SL and usually focus only on the main subject/topic being discussed in the video (e.g. a presidential speech over a health care bill that includes a SL translation, might not mention sign language in any of its metadata). For these videos, visual features seem like a good option; the portion of the screen containing signing activity is usually controlled and with a static and contrasting background, making it easier for a classifier to identify a SL within that region, although with fewer pixels devoted to the signing.
- **News –** Videos in this category have contemporary news being delivered primarily through sign language, different from PIP videos where SL was not the main language of the video. This is a category where videos are more professionally produced, thus metadata is usually of a higher quality. Analogous to an oral news show, videos in this category have takes in different scenarios and lots of other information can be also present in the screen (e.g. supporting graphics like charts, pictures, logo, etc.). This combination makes it hard for visual-based classification to work consistently for these

videos. Thus, like with music interpreting videos, metadata should be more valuable when identifying videos in this category.

3.3 Efficient Sign Language Video Processing

Because much of the research in the area are aimed towards SL translation, a problem far from being solved, efficiency has not been much of a concern. Currently, researchers are focused on techniques to achieve better results (greater recognition accuracy, less constrained environments, etc.) without worrying too much about computational costs.

According to [43], from 2013 to 2014 there was a 200% growth in the number of uploaded videos on YouTube, with roughly 3600 new videos posted per minute by 2014. Even if we had no increase in the number of uploaded videos in the subsequent years, this gives the system millions of videos to be processed daily. Therefore, it is important for the techniques used in our proposed framework to efficiently process each new video, enabling the SL collection to keep up with new content posted on video sharing sites as much as possible.

3.4 Identify different types of sign language

Analogous to the problem of finding a text resource in a particular language, when locating a SL resource, the signed language of the content also matters. Sign Languages have evolved in part independently of the surrounding written language, thus they do not have a one-to-one mapping to any written form [6]. If we consider the example of the signed languages in the United States and England, while English is the written and spoken language on both places, their respective sign languages have totally different origins and there is little mutual intelligibility among signers of these two languages. Figure 1 shows an example of fingerspelling differences between ASL and BSL. Many videos posted on video sharing sites are tagged with generic terms like “sign language” or just partially describe the topic and/or language used, making it impossible for text-based queries to identify the specific language presented on the video. Hence, for the development of a SL community-oriented collection, it is necessary not only to detect sign language content in videos, but also to identify the SL language used in each video.

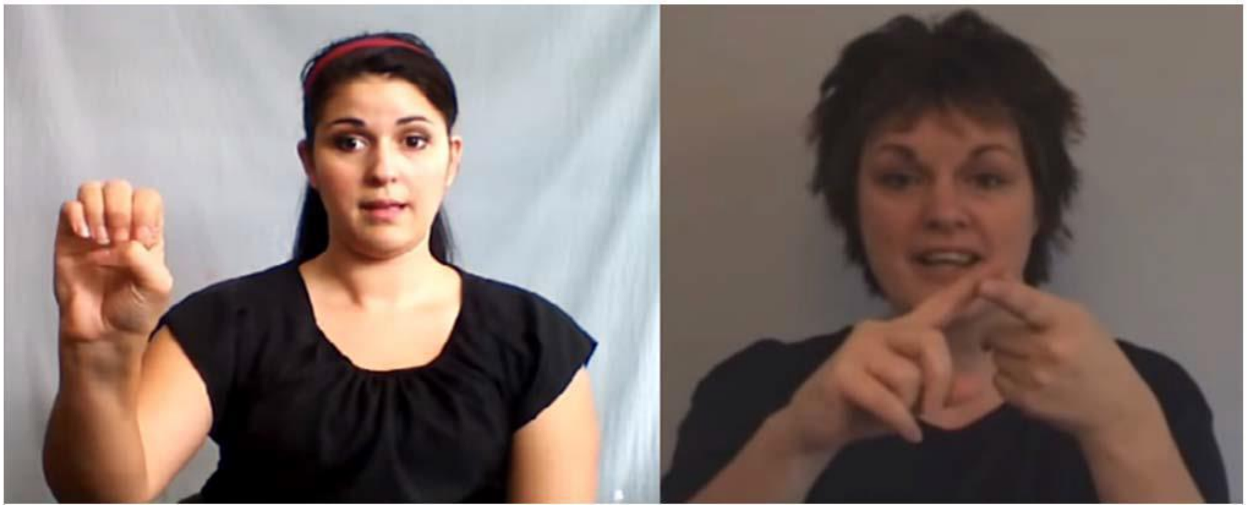


Figure 1. Letter E signed in ASL (left) and BSL (right).

4. SIGN LANGUAGE DETECTION AND IDENTIFICATION

Our proposed Sign Language Digital Library is only made possible with the presence of robust and efficient SL classifiers, capable of both detecting the presence of sign language, and identifying which SL is being used in videos. In this section, we discuss our prior work on SL classification and contributions to our original approach, work that served as an inspiration for our current cascading framework.

4.1 SL detection through visual features

In our very first effort to classify SL videos posted online, we focused on the problem of detecting sign language (i.e. does most of the video contain some sort of sign language?). Our classification was based on two main parts: (1) video processing, to extract information we thought would be valuable for our task; (2) classifier training, given a set of manually encoded videos, train a classifier using the features extracted in (1) as input.

Video processing consisted of, given an input video, converting this video to a 5-D feature space. This was achieved combining background-foreground separation and face detection in the frames of the video. Due to the wide variety of videos posted on video sharing sites, the background model needed to be dynamic in order to accommodate changes in video (e.g. lightning, scenery, etc.). Our dynamic background was computed as a running average of previous frames of the video. In this approach, the parameter α controls the influence of older frames in the current background model, where a high value of α means a model where just the most abrupt changes are detected as foreground, and a low value indicates a model where even slight changes in pixel intensity are treated as foreground activity. So, given a particular pixel of a video frame at time *t*, the corresponding background pixel at time *t* is compute as follows:

$$BP(t) = (1 - \alpha)BP(t-1) + \alpha P(t) \quad (1)$$

where *P(t)* is the grayscale pixel intensity at the current frame. For our approach, α = 0.04, seemed to be a good compromise between properly capturing signer's movement and ignoring irrelevant changes in the background. With a computed background value for each pixel of the frame, we subtract the grayscale value of this same pixel in the next frame - if the absolute difference between them is greater than a threshold, 45 in our case, the pixel is considered a foreground pixel. Small body movements and changes in the background can result in noise being identified as foreground activity. To filter these out, a morphological opening (erosion, followed by dilation) is applied in the foreground pixels, resulting in a cleaner final foreground image. Ideally, just the signers' arms and hands are considered foreground. Figure 2 displays the process of computing a foreground frame given a frame of the video.

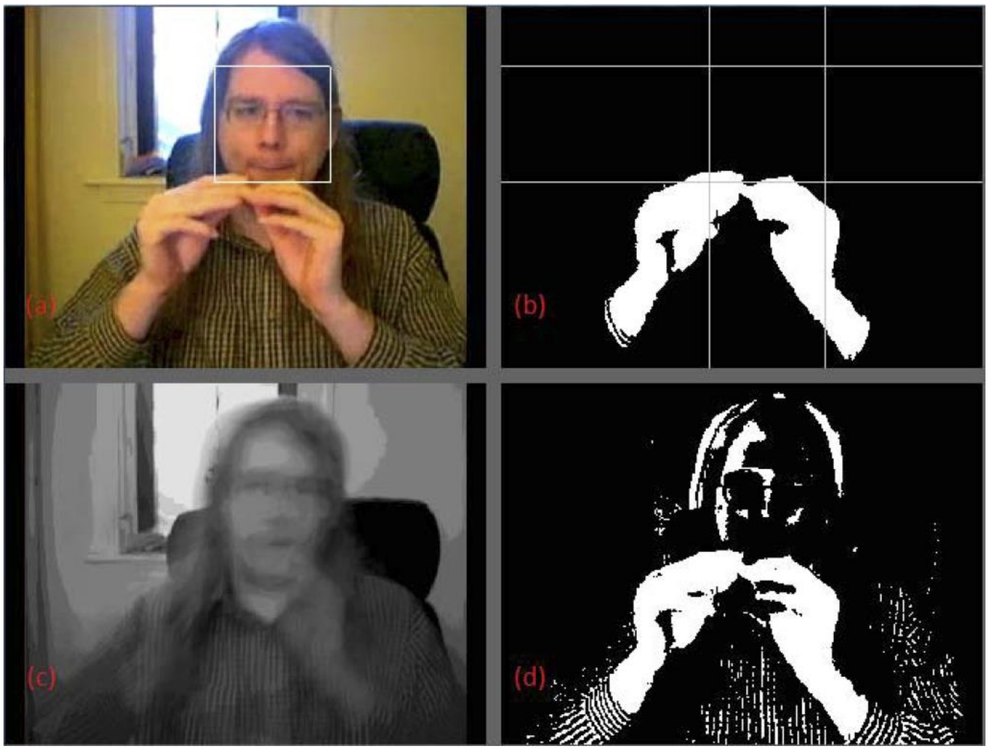


Figure 2. Steps involved in performing background-foreground separation. (a) is the incoming frame of the video; (b) is the final compute foreground; (c) is the background model used for thresholding the incoming frame; and (d) is the foreground image after thresholding but before the morphological opening.

Simultaneous to background-foreground separation, we also detect faces in the frames using Haar-like features [44]. The white box displayed at Figure 2 (a) indicates the face detected at the given frame. With foreground activity and face locations determined, we can then compute our five visual features:

- Total amount of activity per video (VF1) – This is measured as the average proportion of pixels marked as foreground is each frame of the video.
- Spread of activity (VF2) – The percentage of how many pixels of the video have been marked as foreground in at least one frame.
- Continuity of Motion (VF3) – This is measured as the average percentage of new foreground pixels in each frame (i.e. a foreground pixel that was not considered foreground in the previous frame).
- Symmetry of motion (VF4) – Average of the proportion of foreground pixels that are in a symmetric position relative to the center of the signer's face.
- Amount of non-facial movement (VF5) – Measures the average proportion of pixels outside the face area that are marked as foreground pixels in each frame.

With these 5 visual features, we hope to capture three important traits of sign language communication: (1) the higher quantity of arms and hands movements when comparing to other types of gesturing; (2) the more constant flow of motion present in sign language communication; and (3) the location patterns observable in different sign languages. Once the five features are computed for some samples, we could then manually encode them and train a supervised classifier.

Our classification task was to determine if a video was in SL or not, thus a binary classifier would suffice. After experimenting with a few different classifiers (Hidden Markov

Models, K-nearest neighbors, and Support Vector Machine), we found out that a Support Vector Machine (SVM) [45] yielded the best results. To evaluate our approach, we hand selected a collection of 192 videos, 98 SL videos (70 ASL and 28 BSL) and 94 non-SL videos, collected from YouTube. The set of non-SL videos was collected aiming to select likely false-positives (i.e. videos where the background is somewhat stable and there is a person gesturing a lot throughout the video duration). For each video in the collection, we processed a one-minute segment at the middle of the video and extracted the five visual features previously described. We then trained a SVM on part of the dataset and used the rest of it for testing, the results reported below are the average of 1000 random training and testing splits for each context assessed. To quantify the quality of the classifier we use the metrics of precision (percentage of SL classifications that were actually correct), recall (percentage of how many SL videos were classified correctly), and F1 score (the harmonic mean of precision and recall). Table 2 presents the results as we vary the number of randomly selected training samples per class; videos not included in the training set were then used for testing. As is observable, there is a slight increase in the performance as we increase the training set size, while precision irregularly varies within a 3% band, recall has a 4% improvement. This indicates that the classifier works well even with a small training set. Interpreting these results for our broader purpose of detecting SL videos posted in video sharing sites, it means that four of every five videos predicted to be in SL were really SL videos, and nine out of ten SL videos present in the testing set were properly detected.

Table 2. Results for different training set sizes.

#Videos/Class	Precision	Recall	F1
15	81.73%	86.47%	0.84
30	83.62%	88.11%	0.85
45	80.67%	91.00%	0.85
60	82.21%	90.83%	0.86

Given the encouraging result, we decided to investigate the relative value of each visual feature through a series of experiments using feature subsets. First, we estimated the impact on the classifier performance when one of the features is removed. As shown in Table 3, removing symmetry of motion (VF4) has the biggest impact on performance, with a 9% drop on precision and 12% drop on recall, when compared to the classifier trained with all five features. On the other hand, total amount of activity (VF1) appeared to be the least important feature, with its removal leading to a dip of just 1.3% in precision and 0.2% in recall. To confirm these results, we analyzed the performance of each feature individually, training a classifier with just one feature at a time, Table 4 presents the results. Once again, symmetry of movement seemed to have the biggest importance, exceeding by far all other individual features, and even their combination. This clearly indicates that SL communication has a distinguishable pattern of gestures location and this finding inspired further development performed on visual features.

Table 3. Results when all but one feature is used for classification with a training set size of 15 videos per class.

Video feature removed	Precision	Recall	F1 Score
VF1	80.36%	86.25%	0.83
VF2	78.34%	85.41%	0.82
VF3	78.90%	83.62%	0.81
VF4	72.80%	74.30%	0.74
VF5	78.86%	85.60%	0.82

Table 4. Results when just one feature is used for classification with a training set size of 15 videos per class.

Video feature	Precision	Recall	F1 Score
VF1	70.48%	60.14%	0.65
VF2	73.57%	53.26%	0.62
VF3	65.65%	64.03%	0.65
VF4	75.95%	83.69%	0.80
VF5	56.31%	49.52%	0.53

4.2 Improving videos visual feature representation

The original five feature approach previously described had two major drawbacks, it was not able to deal with videos where more than one person was signing, and the simple background model was not able to properly separate background and foreground for more complex videos. Aiming to improve these two shortcomings and considering the importance of symmetry of motion, Virendra Karappa proposed a new feature space to represent the videos, called Polar Motion Profiles (PMP) [5]. PMP is a scale and translation invariant representation that measures the expected signing activity in a polar coordinate system (ρ, θ) centered on a signer's face. Similar to the previous approach, faces are detected using Haar-like features, however, instead of using a single detector, an ensemble of five face detector was used, with majority voting deciding the final result. Parallel to face detection, background-foreground separation was performed using an adaptive Gaussian Mixture Model [46]. This model was more robust than the running average previously used, and could more accurately filter out background activity from the foreground frames. Given both foreground activity and the face locations determined by the ensemble approach, regions of interest (ROI) were then computed in a manner that each ROI would represent the possible signing space of a person (the extent to which the person's arms could stretch). Foreground pixels inside and ROI would then be used to compute PMP. Equation 2 shows the expression for computing the angular coordinate (θ), a similar equation is used for the radial coordinate (ρ).

$$PMP(\theta) = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{R(t)}\sum_{r=1}^{R(t)} PMP_r(\theta, t) \qquad (2)$$

where $R(t)$ is the number of ROIs at the current frame t, T is the amount of frames in the video segment being processed, and $PMP_r(\theta, t)$ is the ratio of foreground pixels (FG) to the sum of all pixels, foreground and background (*BG*), at polar coordinate θ for region r and frame t, as expressed below:

$$PMP_r(\theta, t) = \frac{FG_r(\theta, t)}{FG_r(\theta, t) + BG_r(\theta, t)} \tag{3}$$

This results in a PMP feature vector with 460 elements for each ROI of each frame; the first 360 elements represent each angular coordinate and remaining 100 the radial coordinates. All the PMPs are then averaged and principal component analysis (PCA) is applied to get the final feature representation of the video. Experiments with the available datasets at the time determined that using the first six components of PCA were a good compromise between space complexity and variance representation.

This new feature representation enabled the classification of videos with more than one signer, as well as a better overall performance when there was background noise present in the video. However, the use of an ensemble and the more complex background computation severely impacted the time required to process a video, thus, making this approach not suitable for being used as the automatic classifier of our sign language digital library.

4.3 Reducing computational costs for generating Polar Motion Profiles

To get a sense of how much computational effort is required to process a video, the PMP based approach proposed in [8] takes approximately six times the processed segment duration to compute all the necessary features to be used by the SVM classifier, with the biggest part of it being spent on face detection and generation of the polar motion profiles. While longer segments require more computational effort, Karappa’s approach fixed the segment length to a maximum of one minute. This sets an upper-bound on the time necessary to process a single video, but limits finer-grained analysis necessary in some contexts, such as diarization when a video includes segments of both SL and non-SL content.

While it was a more computationally demanding technique, using PMP to represent the videos had its advantages, it removed the single signer limitation from the SL classifier and allowed it to properly detect SL in more varied types of videos. Because of that, Satyakiran Duggina investigated methods to reduce the computational load and their impact on the accuracy metrics [10]. He suggested three optimizations to the PMP approach that were able to greatly reduce the computational costs, while keeping the accuracy values still high. His first optimization suggestion was to discard the ensemble of face detector and use the single one with the best performance; this lead him to adopt the detector Alt2 as the single one. This sole change was responsible for reducing the average time spent to detect the faces of a video from 896 to 130 seconds, while the F1 score remained pretty much the same when at least 45 videos were used for training the classifier.

The second suggested alteration was to compute the face location just at a given sampling interval. This suggestion was made under the assumption that in a short period of time (i.e. less than a second) face location would change minimally, and could be considered anchored with no real penalty to accuracy. In this approach, using a sampling rate of *n*, at every n^{th} frame, face locations were to be computed and fixed for the next *n* frames. Background-foreground separation would still be performed for every frame. Using the single face detector Alt2 and a sampling rate of 20 frames, the average computational cost to process a one-minute video segment dropped to 31 seconds, almost 30 times lower than the original 896 seconds. All this computational time saved came at the expense of less than 1% in F1 score, which made it a great trade-off.

The last suggested approach was to process just a few equidistant frames and completely get rid of the Gaussian background model, simply computing foreground activity based on the absolute difference between a frame and the previous one. Duggina’s experiments found that at about 10 keyframes the performance of the classifier started to stabilize. Thus, if we consider the original approach where 1800 frames needed to be processed for a one-minute segment (30 frames/second), reducing the processing to just 10 frames is such a great improvement that the bottleneck of the system is now on retrieving the selected video frames than actually processing them. While this change had a bigger hit into the overall classifier performance (F1 score dropped from 78% to 71%), recall remained at the level of the original approach, indicating that a strategy like this could have some use as a first-step filter to quickly discard obvious non-SL videos.

5. A SIGN LANGUAGE DIGITAL LIBRARY VIA CASCADING METADATA AND VIDEO CLASSIFIERS

This dissertation is part of a broader effort to develop an ever-growing community-oriented collection of sign language videos, called the Sign Language Digital Library (SLaDL). SLaDL aims to gather and filter SL contents from general purpose video sharing websites, making it a central place where members of the sign language community can find accessible resources. In this section we describe the major components of this system as a whole, as well as the parts explored in this dissertation.

5.1 Identifying different types of SL

Analogous to spoken languages, there are many different kinds of sign languages, with little to no mutual intelligibility between them. Therefore, for improving accessibility of sign language content, the automatic classifier must be able to not only detect the presence of sign language in a video, but also distinguish what language is being signed.

Our goal was to primarily verify if visual features could be used in situations where metadata can be ambiguous, for example, differentiating videos in ASL or BSL. Both sign languages are used in English speaking regions, but their respective sign languages are considerably different; in fact, studies have shown that just about 30% of the signs in these two languages are shared [47]. In a scenario like this, visual features might prove to be more useful than metadata features, unless the video is accurately tagged for the kind of sign language used.

As an initial experiment to validate the PMP approach to identify SLs, we used the research dataset Dicta-Sign [22]. This dataset contains professionally recorded videos with static backgrounds for 4 European sign languages (British, German, Greek, and French sign languages). Videos were recorded as a conversation between two signers and while there was not a sign-by-sign script, the topics and general guidelines for the conversations were the same in all languages.

Given the absence of American Sign Language at the Dicta-Sign corpus, we used the French Sign Language (LSF) as a proxy for it. Both languages, ASL and LSF, have evolved from the Old French Sign Language, and share a lot of common signs and structures between them. One of the biggest differences between ASL/LSF and BSL is their fingerspelling. While ASL and LSF have a one-handed alphabet where the letters are signed with the signer’s dominant hand, BSL uses a two-handed version. Thus, we expect a clear difference in the patterns of signing between these two languages.

To evaluate the classifier, we collected 128 videos from Dicta-Sign corpus, half of them in LSF and the other half in BSL. Results from 1000 iterations were computed and averaged. At each iteration half of the videos for each class were used for training and the rest for testing. Table 5 presents the results as we increase the length of the video segment being analyzed. Even with segments as short as 10 seconds, the classifier has no problem distinguishing between the two languages, achieving a 93% F1 score. If we increase the processed segment length to the usual 1 minute, the F1 score goes up to 97.5%.

With the encouraging results using the Dicta-Sign corpus, we moved on to test the classifier in a scenario that would resemble real-world more closely. We collected a corpus of 95 BSL videos from YouTube and compared them against 100 of the ASL videos used in [8].

Before executing the classifier, we analyzed if the patterns of signing between the classes would be different enough for classification to succeed. To do so, we have generated two activity maps, one for the ASL collection, and one for the BSL corpus. To compute the activity map for each collection the following equation was used on the foreground videos:

$$I(x,y) = \frac{1}{\sum_{n=1}^{N}\sum_{t=1}^{T} R(t)} \sum_{n=1}^{N}\sum_{t=1}^{T}\sum_{r=1}^{R(t)} i(x,y) \quad (4)$$

where N is the number of vides in the collection, T is the total amount of frames for the *n-th* video, *R(t)* is the number of ROIs at frame *t*, *i(x, y)* is the pixel value (since this is the foreground video, the value is either 255 or 0) on the given region *r* at the coordinate (x, y), and *I(x, y)* is the resultant average pixel intensity in the activity map. Figure 3 presents the activity maps for the two languages. As expected, although similar, there is a clear distinction between the signing activity in the videos from these two languages. ASL has a broader signing area and most of the activity happens on the right-hand side of the signer, on the other hand, BSL videos have a more restricted signing area and most of the signing activity happens on the top central part of the signer's torso. This is the type of difference that we were expecting given the fingerspelling distinction between these languages.

Table 5. Precision, Recall, and F1 measures for BSL vs LSF classification.

Video length	Precision	Recall	F1 score
10 s	94.7%	91.7%	93.0%
30 s	96.7%	93.5%	95.0%
60 s	99.6%	95.6%	97.5%

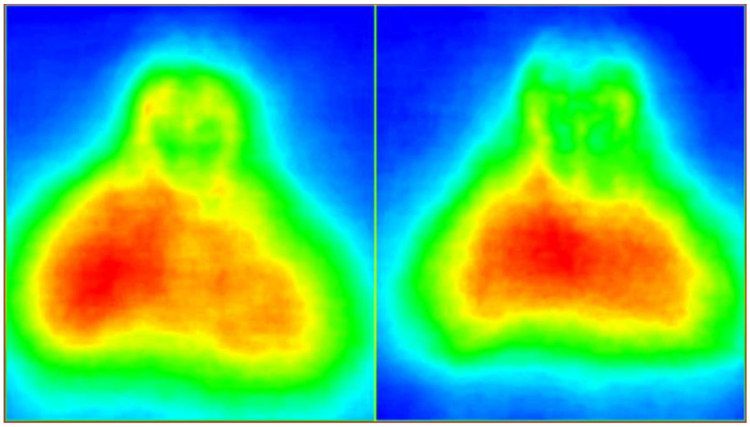


Figure 3. Activity maps for ASL (left) and BSL (right).

With the activity maps highlighting the differences on the datasets, we then run the classifier to verify the actual classification task performance. Again, we experimented with different video segment lengths, and the reported results are the average of 1000 iterations where in each iteration each class had 60 videos randomly selected for training, and the rest was used for testing. Figure 4 presents the results. Like with the Dicta-Sign corpus, increasing the segment length generally leads to better results. However, we can observe that while the results are still encouraging, it is nowhere near the values obtained on the Dicta-Sign corpus (70% compared to 97%). This can be mostly explained by the nature of the videos present in each dataset. While Dicta-Sign corpus is composed of professionally recorded videos, with basically no noise present at the background, the set of collected BSL videos along with the ASL videos used in this second experiment are much more representative of real-world cases. This kind of video presents a much bigger challenge to the classifier due to the inevitable presence of noisy artifacts in the computed PMPs.

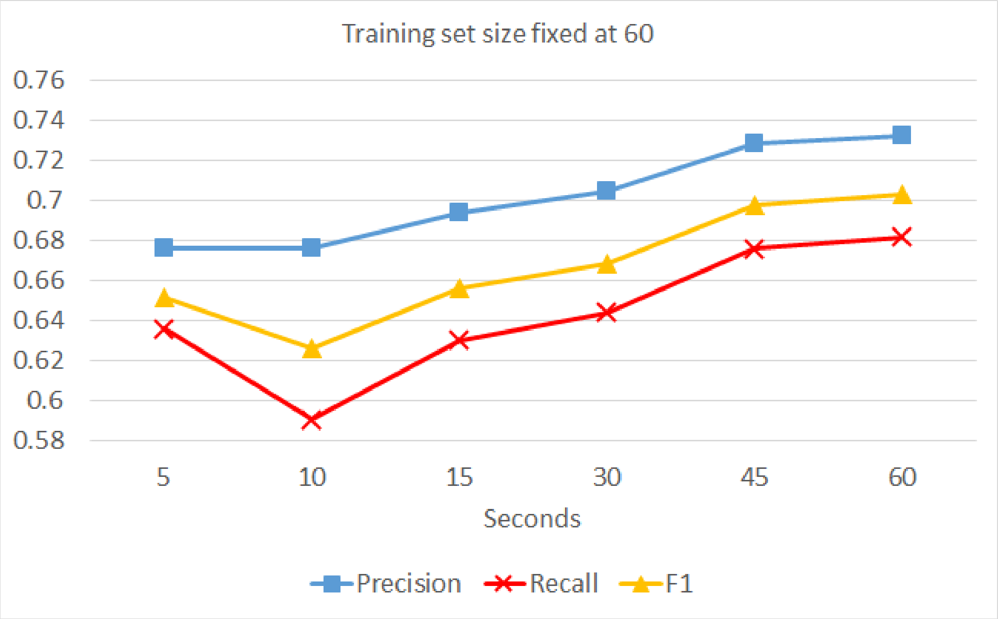


Figure 4. Precision, Recall, and F1 score for ASL vs BSL classification.

5.2 SLaDL architecture

We envision the SLaDL system as a web portal where members of the sign language community can go to locate content on their desired sign language and topics of interest. The focus of the system is not on playback or storage services (these capabilities are already provided by the general purpose video sharing sites), but on how to locate and augment these videos with metadata pertinent to the SL community, analogous to systems like TalkMiner [48].

A challenge for maintaining an up-to-date collection in SLaDL is the number of videos that have to be processed and the relatively small corpus of SL videos when compared to video sharing sites' entire collections. These characteristics makes it unfeasible to simply scan through these collections classifying each video. Hence, we utilize a snowball approach to crawl potential videos to be inserted into the SLaDL corpus. Figure 5 presents the general design of the system.

In this architecture, the crawler is fed from two different sources. The first one is the set of known SL videos; these are the videos that have already been automatically classified and marked as one of the sign languages supported by the system, where related videos are going to be marked as potential sign language videos and enqueued for processing. The other source for the crawler are the results from user queries. These queries are augmented with terms related to sign language, and the returned videos are included in the dataset of potential SL videos. Videos marked as a potential SL video by the crawler are then dispatched at some point to the automatic classifier, which will determine if this video will be inserted in the digital library corpus or not.

While user-involvement is a common practice on the development of digital library collections [41], [49] and we expect it to be a major component on the final version of SLaDL, in this first instantiation we have focused on the interactions of the cascading classifiers and the crawler. Therefore, the only user action currently capable to indirectly influence the corpus of the digital library, is to guide the crawls via search queries.

Overall, our proposed architecture for SLaDL should form a solid guideline for the development of community-oriented collections, with the general framework being able to be used by different automatic classifiers and explored in many other domains. Figure 6 presents the detailed architecture of the current SLaDL implementation. The backend of the portal was developed in C# and the front-end (Application Layer) with ASP.Net MVC and JavaScript.

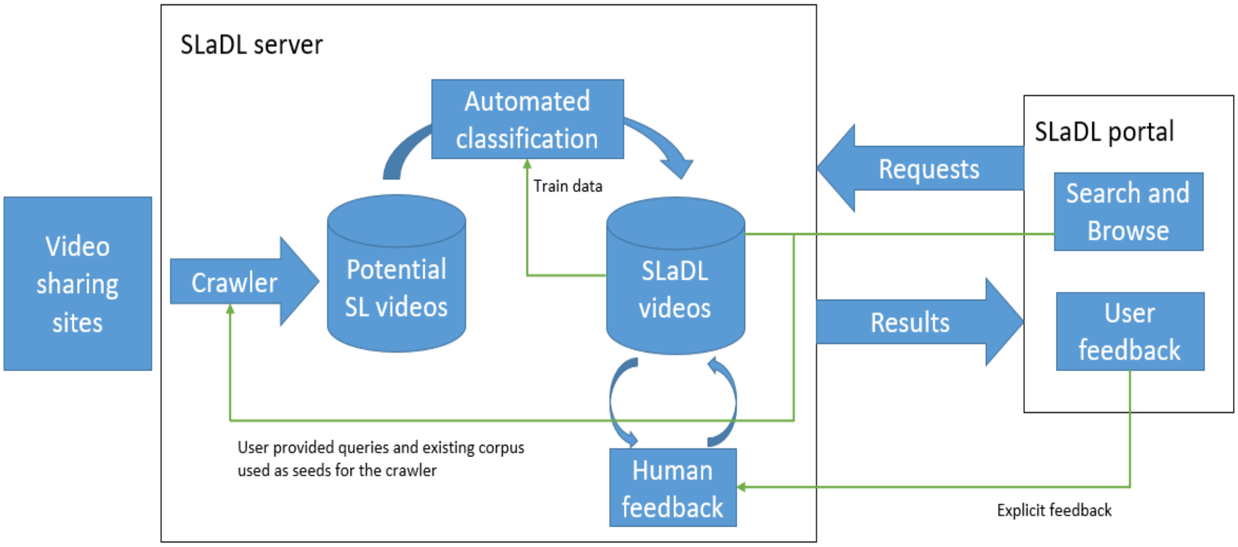


Figure 5. Overall design of SLaDL.

The Services Layer makes the bridge between the application layer and the data access layer, it contains all the business logic of the system and the domain models. It is also responsible for the background task that periodically runs the classifier and crawler managers.

The crawler manager is responsible for connecting to video sharing sites APIs, obtaining all the required video information for processing (i.e. metadata and video stream). In our current instantiation the crawler is able to fetch videos from YouTube. The automatic classifier manager is responsible for processing the top potential videos identified by the crawler, it triggers the execution of video processing in C++ and the cascading classifier in MATLAB, using inter-process communication. Data access layer is responsible for the exchange of data between the services and the data persistence layer. Data is persisted on the MS SQL relational database, and as previously mentioned, we do not store the actual videos, since this would duplicate services already provided and demand large quantities of storage. Instead, we just keep all the required metadata to access and retrieve the video later, and the extracted video features necessary for classification and reclassification.

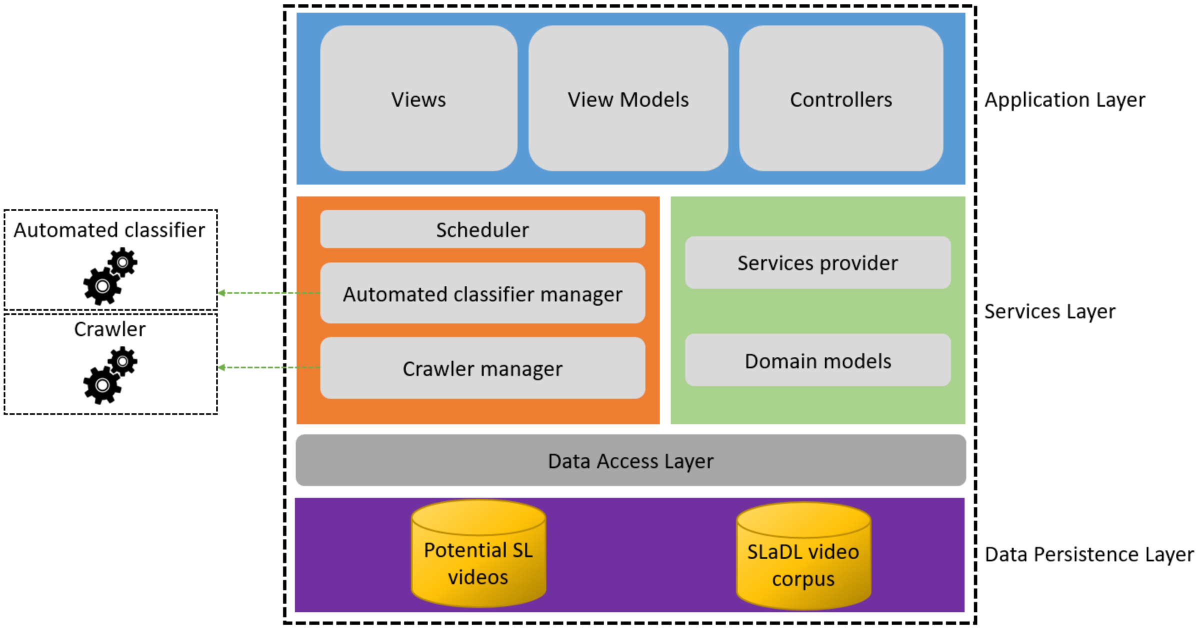
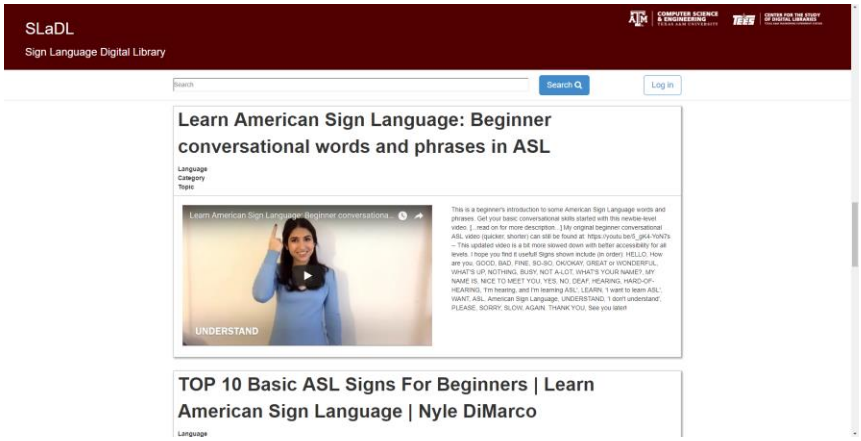


Figure 6. Architecture of current SLaDL implementation.

5.3 SLaDL user interface

In its current version, the Sign Language Digital Library mainly allows the users to browse and search videos to watch. Figure 7 displays the library home where the user can start looking for content. In this view queries are appended by the general term “sign language”, and results are provided for more than one sign language. If the user is logged into the system then additional options are available, as shown in Figure 8. For a logged user the system will also append the queries with terms related to the specific sign languages of interest to the user and results in other languages will be filtered out. Logged users can bookmark videos for future reference, annotate videos with tags related to language and topic and access video recommendations. Video recommendations are based on popular tags used by the user (i.e. if the user has bookmarked a lot of videos with the music-signing tag, we are going to recommend other videos that have been tagged as music-signing by other users).

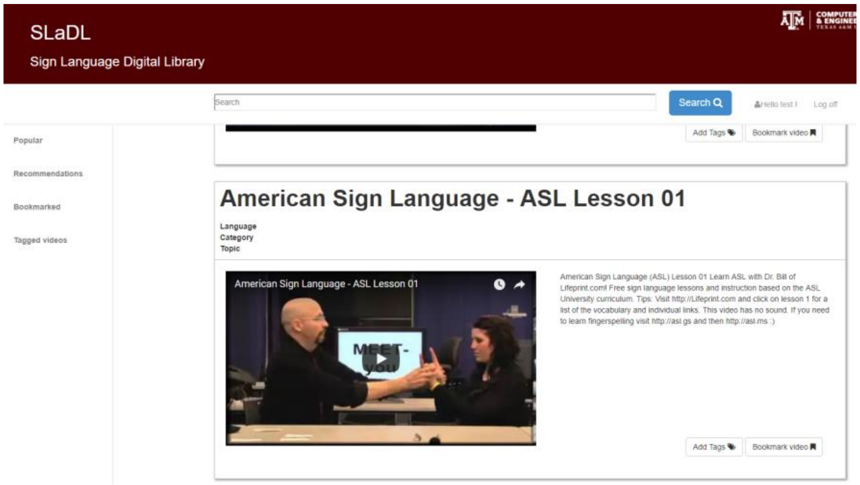

SLaDL
Sign Language Digital Library
Search
Search
Log in
Language
Category
Topic
Learn American Sign Language: Beginner conversationa...
UNDERSTAND
Language

Learn American Sign Language: Beginner conversational words and phrases in ASL

This is a beginner's introduction to some American Sign Language words and phrases. Get your basic conversational skills started with this newbie-level video. [...read on for more description...] My original beginner conversational ASL video (quicker, shorter) can still be found at: https://youtu.be/5_gK4-YoN7s -- This updated video is a bit more slowed down with better accessibility for all levels. I hope you find it useful! Signs shown include (in order): HELLO, How are you, GOOD, BAD, FINE, SO-SO, OK/OKAY, GREAT or WONDERFUL, WHAT'S UP, NOTHING, BUSY, NOT A-LOT, WHAT'S YOUR NAME?, MY NAME IS, NICE TO MEET YOU, YES, NO, DEAF, HEARING, HARD-OF-HEARING, 'I'm hearing, and I'm learning ASL', LEARN, 'I want to learn ASL', WANT, ASL, American Sign Language, UNDERSTAND, 'I don't understand', PLEASE, SORRY, SLOW, AGAIN, THANK YOU, See you later!

TOP 10 Basic ASL Signs For Beginners | Learn American Sign Language | Nyle DiMarco

Figure 7. Homepage for the Sign Language Digital Library.



American Sign Language - ASL Lesson 01

American Sign Language (ASL) Lesson 01 Learn ASL with Dr. Bill of Lifeprint.com! Free sign language lessons and instruction based on the ASL University curriculum. Tips: Visit http://Lifeprint.com and click on lesson 1 for a list of the vocabulary and individual links. This video has no sound. If you need to learn fingerspelling visit http://asl.gs and then http://asl.ms :)

Figure 8. Additional options for a logged-in user.

5.4 Candidate videos crawler

Considering the magnitude and rate of growth of video sharing sites' collections, it is not possible for us to classify every video they host. We need an efficient crawler that can fetch videos that are likely to contain sign language, reducing computation wasted on processing non-SL videos.

Our crawler is guided by two principles that help it to increase its SL to non-SL videos ratio. First, we use user queries results as seeds to our crawler; thus, our seeds are videos that are most likely to contain sign language based on metadata. Second, we use a focused crawling mechanism to expand the crawler frontier. In this approach, videos discovered by the crawler are ranked according to a classification score and a priority queue is used to always expand the frontier from the video with the highest score.

Our focused crawler uses a SVM trained with metadata features to assign a likelihood of a video being in SL, thus this SVM is trained only for SL detection, as opposed to the classifier SVM that also needs to handle identification. Metadata features are represented in the same manner as the identification SVM discussed in the next sub-section. Therefore, for each new video reached by the crawler, we extract metadata features and perform a classification for it. The likelihood obtained in the trained SVM is then used as the score for the focused crawler (i.e. a video with a high likelihood might be a good indicator of a video worth further processing).

For every video that we select from the frontier, the video with the highest likelihood at that given time, we explore three sources for finding potential new videos: (1) other videos from the same channel; (2) videos from subscribed channels; and (3) related videos returned by YouTube. All these potential videos are ranked based on their metadata as well, and could be selected at later iterations of the crawler if they have a score high enough.

At the crawling phase we have also embedded the first stage of our cascading classifier. Since we are already extracting metadata for ranking the candidate videos, we use this metadata for the decision rule classification. If the video cannot be properly classified by the rule-based system then we will keep it on the priority queue of potential videos until it is selected for processing by the rest of the cascading.

5.5 Multimodal Sign Language videos classification

At the core of our framework is the automatic classification system. We need a classifier robust enough to deal with the variety of videos present in video sharing sites, but also one that can classify new videos with as little computational effort as possible.

As a first step towards this goal we incorporate metadata features in our video representation. Our intuition is that textual features can provide complementary information to visual features, mainly when considering the task of identifying certain types of sign language. Consider Figure 9 and the task to distinguish between these three types of SL (ASL, BSL, and Mexican Sign Language - LSM). Visual features might be better suited to differentiate between ASL and BSL, given that these two sign languages have different origins and alphabets that are visually distinctive, while metadata for both languages will be presented in English, making it difficult for metadata feature to work unless the video is accurately tagged for the specific language. On the other hand, textual features are probably more valuable for distinguishing between ASL and LSM; these two sign languages have originated from Old French Sign Language and thus have some overlap between them, but metadata for the videos in these languages are expected to be considerably different, given that ASL videos metadata is most probably in English and LSM videos metadata is likely in Spanish.

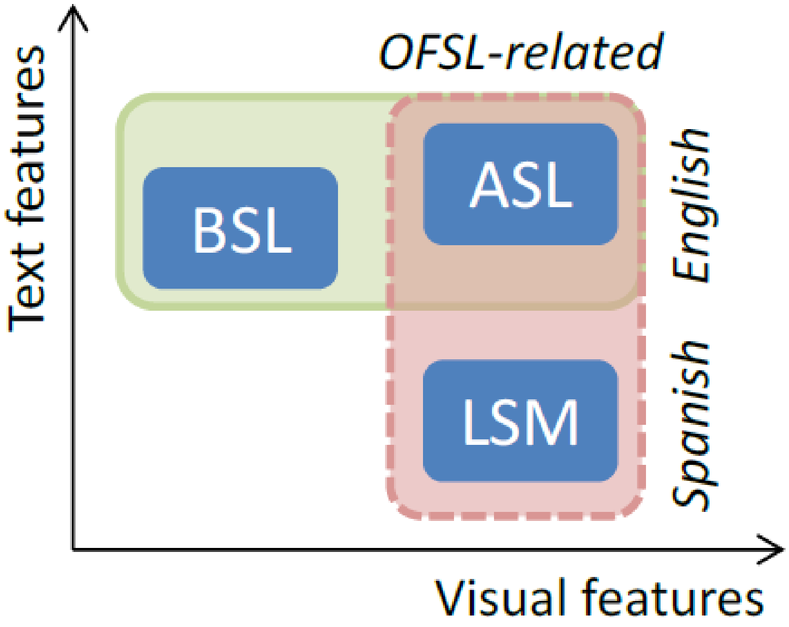


Figure 9. Similarity between ASL, BSL, and LSM when considering textual features or visual features.

Extracting metadata and computing textual features is also a much faster process than computing visual features like PMP. Therefore, depending on how we use them, metadata features might be able to make new videos classification faster. To decide on how to represent video metadata in the feature space we have analyzed the performance of three well-known techniques for document feature extraction: (1) Term frequency-inverse document frequency (TF-IDF); (2) Latent Dirichlet Allocation (LDA); and (3) Non-negative matrix factorization (NMF). Results for our analysis and the decided feature representation are discussed at the Results section.

TF-IDF is an established method in information retrieval for weighting and ranking resources [50]. It has also been used successfully to represent documents in clustering and classification problems. In this approach, resources of a collection are represented as a vector of numerical values assigned to each word in the collection. For each term present in the corpus, its

TF-IDF value for a given document is obtained by multiplying the number of times it appears on the document with the inverse of its document frequency, which is the count of documents in the collection that have the same term present at least once. Using the inverse of document frequency aims to reduce the importance of common terms that appears throughout the collection and thus provide little discriminative information.

LDA is commonly used as a topic modeling tool. It is a hierarchical Bayesian model that represents each document of the corpus as a mixture of K topics. The K topics are inferred from the whole corpus, where each topic represents a distribution of word occurrences, with the words with higher probabilities indicating the subject of the particular topic. Topic probabilities for a document are then assigned using the word frequencies of that document. In our analysis we have used the online version of LDA described in [51].

NMF is also commonly used for topic modeling, although it was originally developed as a dimensionality reduction technique. It works by approximately factorizing a matrix A into two non-negative matrices W and H, where W has K columns and H has K rows. There are many possible algorithms to compute the approximate factorization $A \approx WH$, we used the approach proposed in [52]. Interpreting the NMF solution as a topic modeling problem, we can interpret the K columns of matrix W as topics and each column of matrix H as the topic distribution for each document in the collection. Another common way to interpret a solution for NMF is as a clustering problem, where W contains the clusters centroids and H has the document-cluster assignments.

Once a feature representation is decided for the metadata, we still have to combine these textual features with our current video features. In general, there are two broad categories of algorithms to combine multimodal features, early fusion and late fusion. For early fusion, feature representations from each media type are combined prior to training or testing of the classifier, so the fusion happens at the feature representation stage. This approach has the advantage that no additional classifiers need to be trained.

For late fusion, each media type has its classification done independently and the results are merged somehow into a final decision. Therefore, it is said that fusion happens at the semantic space. Usually, fusion at the semantic space takes two general forms: (1) outputs from unimodal classifiers are used as a new feature space; or (2) some quantitative metric is applied on unimodal outputs and the final label is decided based on the metric result. In our analysis we have tried both of these approaches. As previously discussed in the Related Work section, late fusion of multimodal features seems to improve accuracy when compared to early fusion in many classification tasks. Therefore, the overhead of training more than one classifier might be worthwhile, mainly considering that the process for training a classifier using textual features is considerably less computationally demanding than using visual features, thus the extra work represents just a fraction of the work that needs to be done for generating the video features.

5.6. Cascading classifiers

Given the performance of multimodal classifiers discussed in the Results section and the video-based optimizations proposed in [43], we explored a cascading of classifiers. This approach allows us to benefit from the video optimizations while reducing the performance impact on the accuracy of the classification. The intuition behind the cascading classifiers is that videos that are easy to determine as having or not having one particular sign language can be quickly classified by low-cost classifiers, such as the keyframes or frame sampling approaches, reducing the number of videos that require a full video segment feature extraction.

Because we are using less information in earlier stages of the cascading pipeline, it is important to just make a final assessment of a video class if there is enough confidence in the prediction by the current cascading stage. To simulate the prediction confidence, we will use a variant of the traditional SVM described in [53]. In the results section we analyze the effect of different threshold values for the prediction confidence. Low threshold values yield to videos being classified faster on average, but this comes at the expense of more false positive and false negative classifications. A higher threshold means a better performance of the cascading pipeline in terms of accuracy, but can severely degrade the average computational time required to process a video, up to the point where the overhead of the cascading pipeline can make the average time longer than the original full video segment approach, rendering useless the use of such a pipeline.

We have experimented with two versions of the cascading classifiers. In both of them, the first stage is a rule-based classification system and it is embedded at the crawling mechanism. Considering how similar the performance of the frame sampling approach described in [43] is to the full PMP method [5], in one version of our cascading classifier we have used 4 stages (Rule-based, Keyframes + Metadata, Frame sampling + Metadata, Video Segment + Metadata) while in the other one we have removed the last stage and completely avoided doing full video segment processing.

All the stages of the pipeline have the same decision criteria, if both classifiers agree on the resultant class and the prediction confidence is greater than the threshold, than the video is classified as the particular class and no further processing is required. If any of these two validations fail, the video is then forwarded to the next step in the pipeline, if the current stage is already the last stage of the pipeline, then the video is assigned a final class regardless of classifiers agreement or prediction confidence.

It is important to note that the metadata features used in stages 2 to 4 are the same, thus, textual features have to be extracted just once per video and the computational time to do so is negligible when compared to visual features. The same is not true for visual features, where each stage uses more visual information than the previous one, therefore, requiring more computational effort. Considering this, we can estimate the time required to classify a video on the cascading pipeline as the following:

- For videos classified at stage 1: Classification done at crawling time, no extra computational required.
- For videos classified at stage 2: Time to perform keyframes computation (when using the keyframes approach, both face detection and background-foreground separation are done sequentially) plus time spent generating PMPs.
- For videos classified at stage 3: Time to perform background-foreground separation (Face detection in the sampled frames is done in parallel to background subtraction and is the fastest process between the two), plus time spent generating PMPs, plus time spent at stage 2.
- For videos classified at stage 4: Time to perform face detection in every frame of the video (face detection at every frame is more costly than background subtraction, thus it will dictate the time spent at this step), plus time spent generating PMPs, plus time spent at the previous 2 stages.

Figure 10 presents the overview of the cascading classifiers when using the four stages. Computational effort increases as we move forward on the pipeline. For the version where we removed full video segment processing, the pipeline does not have the last stage and the Metadata + Frame sampling stage makes its decision regardless of prediction confidence or classifiers agreement, Figure 11 presents this scenario. Each step of the cascading pipeline is further described below.

5.6.1 Rule-based classification

The first and cheaper stage of the classification pipeline is the rule-based classification. This stage happens along with the crawler mechanisms, which means that as soon as the crawler reaches new videos they are classified by the rule-based system. The rule-based classifier contains a set of active rules that are compared to the metadata of a given video. If the video metadata fits into any of the active rules, the video is then labeled with the respective rule output.

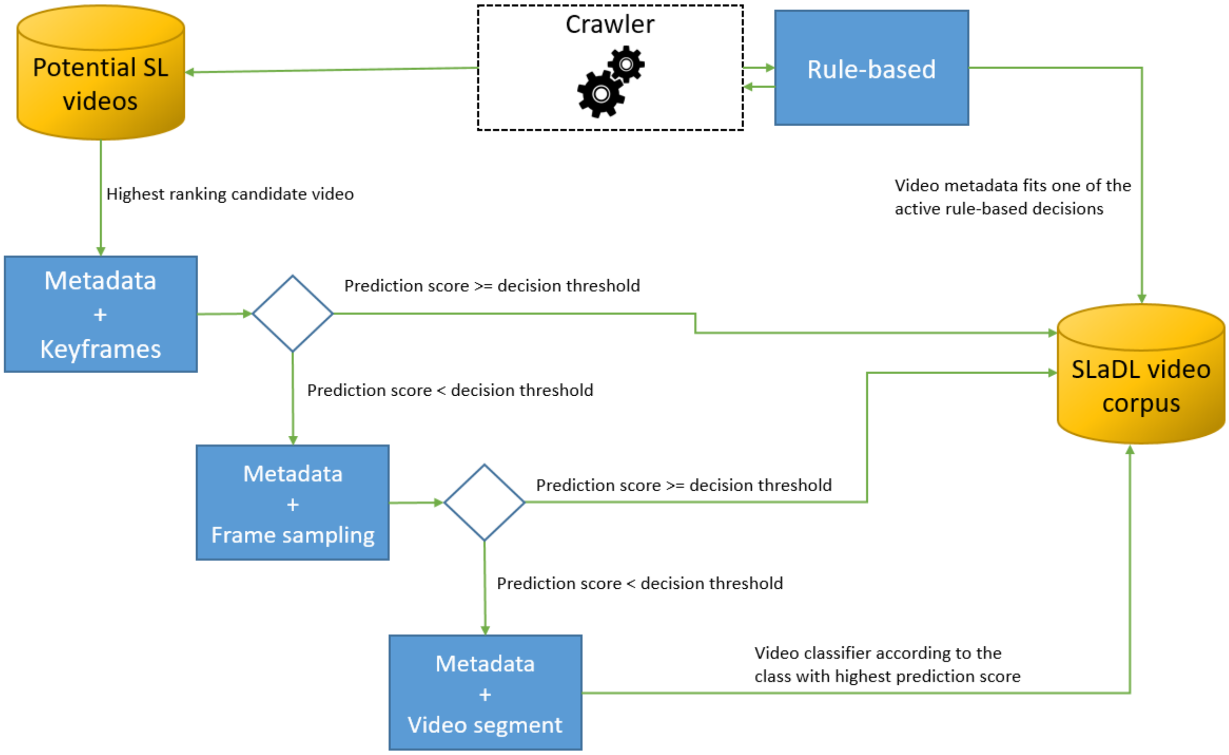

Potential SL videos
Crawler
Rule-based
Highest ranking candidate video
Video metadata fits one of the active rule-based decisions
Metadata + Keyframes
Prediction score >= decision threshold
Prediction score < decision threshold
SLaDL video corpus
Metadata + Frame sampling
Prediction score >= decision threshold
Prediction score < decision threshold
Metadata + Video segment
Video classifier according to the class with highest prediction score

Figure 10. Overview of the cascading pipeline when all four stages are used.

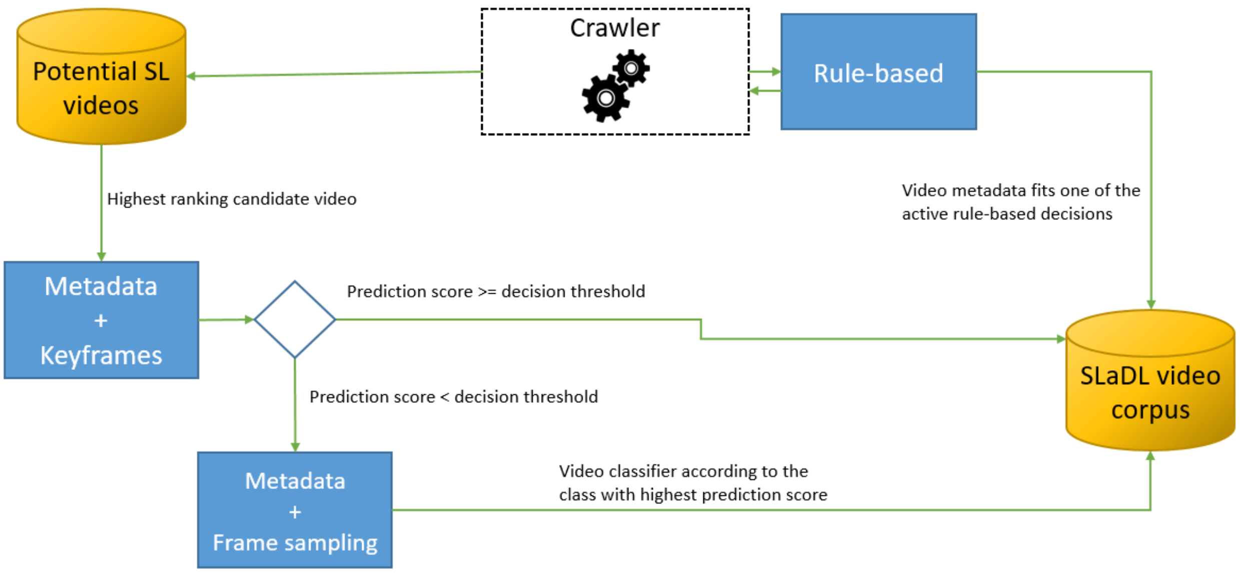

Crawler
Potential SL videos
Rule-based
Highest ranking candidate video
Video metadata fits one of the active rule-based decisions
Metadata + Keyframes
Prediction score >= decision threshold
SLaDL video corpus
Prediction score < decision threshold
Metadata + Frame sampling
Video classifier according to the class with highest prediction score

Figure 11. Overview of the cascading pipeline when video segment processing is disabled.

The output of a rule can be either one of the supported sign languages in the system, or an indication that there is no sign language present on the video. In the event that a potential video metadata does not fit into any of the active rules, the video remains into the crawler priority queue for further processing by the pipeline.

To obtain our set of active rules, we use the C4.5 algorithm to build Decision Trees [54]. Due to the nature of this algorithm, we convert the video metadata values to a codebook representation that is then used for learning the decision tree. Because title and description can have any arbitrary pieces of text, we left these fields of metadata out of our tree representation, focusing on metadata fields that are more suitable for a codebook transformation. Therefore, our decision tree, and consequently rule set, is built using the following fields of metadata:

- Channel URL - Again, given how we represent the metadata using a codebook, it is better to identify the channel by its URL than its name (which could have a different number of words);

- Video category – Each video uploaded to YouTube has a category indicated by the user who posted the content. The user can select the category out of a list of pre-determined categories available;
- First keywords – Besides the category, the user posting a video has also the chance to provide a list of keywords (or tags) to identify her video. Opposed to the category, the user can freely type as many keywords as she wishes. So, considering this we have limited our attention to the first three keywords indicated by the user, as we hope them to be more representative of the video's content than keywords added later on the list.

For learning the decision tree, we use a set of pre-classified videos; these videos were previously labeled as having one particular type of sign language or no sign language at all. At the end of its training phase, we have a decision tree that provides a classification for all the videos present in the training set in a manner that reduces the expected error on it. However, this is the first stage of our pipeline and we are using very little information of each video to try and make the classification. Thus, no matter how accurate our decision tree is in our training set, we still have plenty more information available to us to make a possible better-informed decision. That means that unless we are pretty confident about a specific rule, we should not rush a final classification at this stage.

Considering this, after the initial Decision Tree was learned from the training data, we perform some post-processing to disable rules that did not work well in the training set. From the Decision Tree, we traverse all of its paths from root to leaves to obtain a rule set. From the obtained rule set, we test each training video against all the rules, obtaining the accuracy for each rule in the set. If a given rule has an accuracy below the accepted threshold, it is then discarded and not used in the final active rule set. Figure 12 presents the subtree formed from a decision tree trained for the SL vs Non-SL problem containing only the rules above the set threshold, which in the depicted example was 100%.

A special case when classifying new videos in our rule-based algorithm is when at least one of its metadata field does not have a corresponding transformation in our codebook (i.e. the particular keyword, category, or channel was never seen in the training set). For these cases it is not possible to obtain a rule-based classification and the video must be kept into the crawler priority queue.

At any point it is possible to retrain a new rule set for the rule-based classification. Ideally, retraining would make use of all the videos previously classified by the SLaDL system or a subset of them, however, given the current absence of a human-feedback mechanism to clean the results of the automatic classifiers, this strategy can bring a lot of noise to the training set. Once the SLaDL system has such human-feedback, retraining for the rule based classifier and all the other classifiers should be completely automated, with no necessary human interference beyond feedback on individual videos.

5.6.2 Keyframes and Metadata classifier

If a video is not classified by the rule-based system, it will wait in the crawler priority queue until its turn of being processed by the rest of the cascading pipeline. In this stage, new metadata features are extracted as well as visual features based on the keyframes approach previously discussed.

Metadata features extracted at this stage are the reused for the latter stages of the pipeline, and are computed from the title, description, and keywords of a video. Different from the rule-based system, at this stage we use all the keywords provided to a video, not only the first few.

For the video features we use the keyframes approach described in [10]. We use 10 equally spaced frames of the original one-minute video segment and perform direct pixel-by-pixel subtraction between adjacent keyframes to identify potential signing activity. If subtraction absolute value is higher than the set threshold we mark the corresponding pixel as a foreground pixel. For each keyframe we also compute faces locations using Haar like features [44]. Once face location and foreground activity are computed, we then extract the polar motion profile (PMP) as the final visual feature representation of the video.

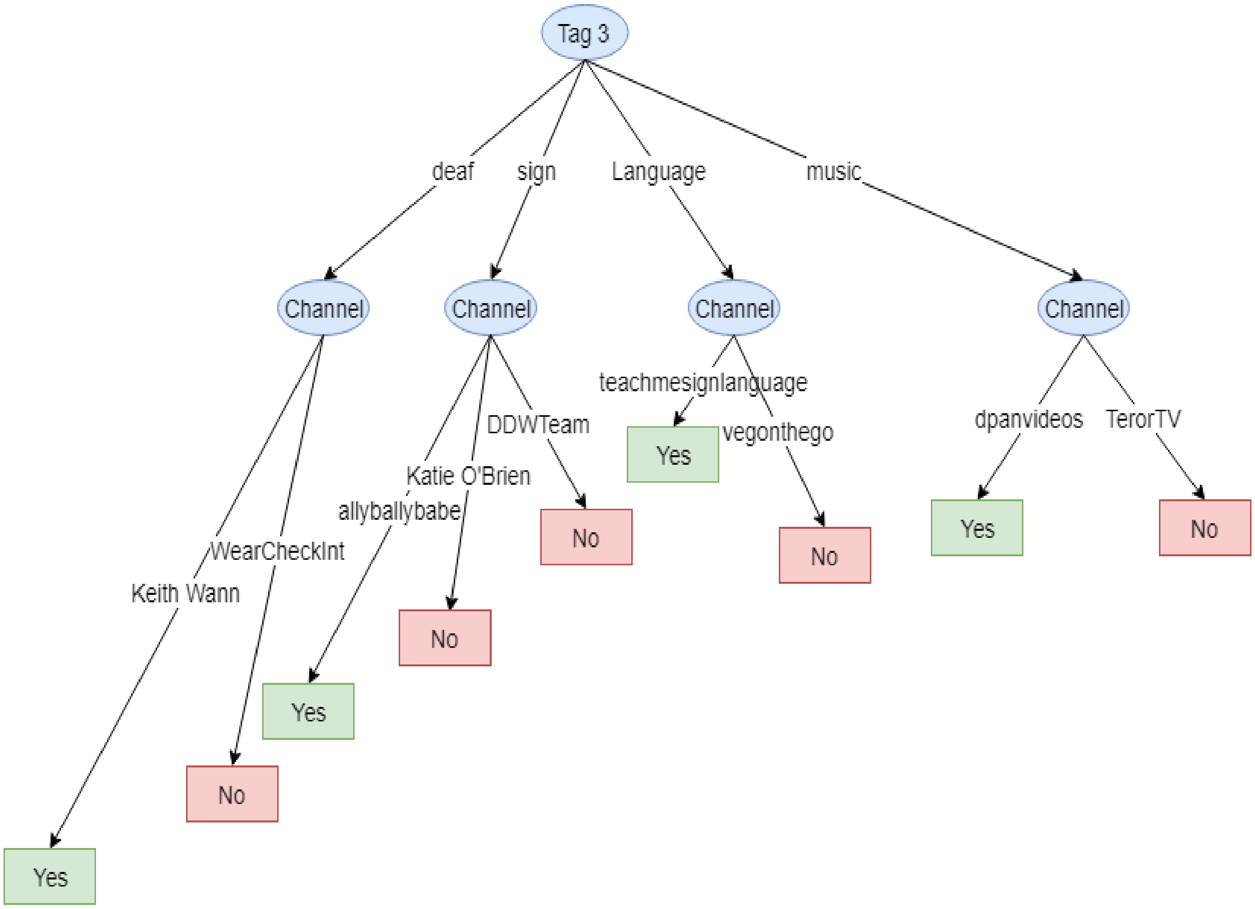


Figure 12. Part of a decision tree trained for classifying videos as SL or Non-SL containing the active rule set.

This is an extremely cheap stage, since we are reducing the whole video analysis to 10 frames. Considering a video with 30 frames per second (fps), we are computing 180 times less frames than the original PMP approach where a complete one-minute segment is processed.

With both visual and textual features extracted, we are ready to classify the new video using the trained SVM(s) for this stage. SVMs are traditionally used for solving binary problems. However, due to the fact that our problem is not only detecting but also identifying the sign language present in the video, we need to use a SVM implementation that is capable for handling more than two classes. For our problem we used the one-against-one method described in [55], where given *k* possible classes, *k(k - 1)/2* binary SVMs are trained to represent each possible pair in the set, and majority voting is used to determine the resultant class of a new sample. This same SVM configuration is used to train the SVMs for the latter stages of the pipeline.

For a classification performed at this stage be accepted as the final class of a video, its prediction confidence needs to be higher or equal than the set classification threshold, otherwise the video will be forwarded to the next step in the pipeline.

5.6.3 Frame sampling and Metadata classifier

At this stage we use the same metadata features from the previous stage and combine them with PMP features extracted from processing videos with the frame sampling approach. When using frame sampling the two steps of video processing (locate faces and detect foreground activity) are done in parallel. Frame sampling is performed just by the locate face algorithm, meaning that background-foreground separation is still performed on the whole one-minute video segment as in the original PMP approach. For our pipeline, we have used a sampling rate of 20 frames (i.e. every 20th frame we compute face locations and these are fixed for the next 19 frames). In such a scenario, face detection is not the bottleneck of the video system anymore and the processing time is dictated by the background subtraction, which can be performed roughly in real time.

Once visual features are extracted at this stage, we combine them with the metadata already available and perform the classification on the trained SVM(s) of this stage. When using the 4 stages pipeline, we apply the same conditional test of the previous step to verify if the classification result should be final. If full video segment processing is disabled, then the resulting class of this stage will be considered final.

5.6.4 Video segment and Metadata classification

When enabled, this is the last stage of the pipeline and the most computationally demanding. It is important to note as well that processing time for visual features extraction at this stage largely surpass the average video length presented on YouTube (about 5 minutes) [43], this means that with the absence of a cascading approach it would be really hard for the SLaDL system to expand its corpus at an acceptable rate. Therefore, our goal is to keep the number of videos classified in this stage at the minimum possible. In fact, we expect that tuning the prediction confidence threshold, the majority of the videos can be classified at earlier stages, thus, mitigating a possible bottleneck by the heavy computational load of this stage.

At this stage, metadata features previously computed are combined with PMP features computed over a continuous one-minute segment extracted from the middle of the video. This is the same approach proposed in [8], but using a single Haar-like face detector, instead of the ensemble version. Since this is the last stage of the pipeline, the output of the trained SVM(s) for the new sample is going to be always final.

6. EXPERIMENTS

In this section we present the research questions investigated in this work and the experimental setups used to verify these questions. We have conducted experiments to try and answer the questions related to each main component of our system: (1) the crawler; (2) a multimodal classifier; and (3) the cascading approach. Table 6 presents the research questions.

Table 6. Research questions explored in this work.

Research question	Evaluation method
RQ1: How do different techniques for crawling new videos influence the resulting digital library corpus with respect to the size and variety of it?	To identify a reasonable approach for the crawler, we will compare the performance of different crawling strategies. Results from the crawler will manually examined and coded through a qualitative analysis.
RQ2: Which metadata and video features combination technique works best for the detection of sign language content?	With the intent to improve the classification system performance, we will compare different combination strategies for the metadata and video features. A quantitative analysis will determine the best approach to be used in the cascading pipeline.
RQ3: How do different prediction confidence thresholds affect the average computational load required to process a video?	The staged classifier aims to reduce the computational load required to process a video. We will evaluate the tradeoff between accuracy and computational load according to different values of confidence threshold.

6.1 Crawler configurations

To explore RQ1 we analyzed the videos encountered by the crawler under three different configurations: (1) Breadth-first crawling; (2) Depth-first crawling; and (3) Focused crawling. For each one of these configurations we have started the crawler with some starting seeds (i.e. an initial frontier) and collected new videos until a quota of 300 news videos were found. The initial frontier is made of a set of 6 videos randomly selected from our ASL vs. Non-ASL dataset. Manual encoding of returned videos was then performed and videos were grouped as containing sign language or not having sign language. Our crawler is capable of using subscribed channels or related videos from YouTube as its sources of expansion, however, given the unlimited number of videos that a channel can have, we limited our analysis to expansion though related videos only.

In the Breadth-first crawling approach, for each video present in the frontier, we collect five of the top related videos and include them into a queue; this will be the expansion step for this particular video. Once we are done expanding one element in our frontier, we get the first video waiting in the queue and perform the expansion process on it. We keep doing this until the quota of 300 videos is reached. While in this experiment we are manually encoding the video, if an automated classifier was used, it would classify the video at the moment that frontier expansion was happening on it. The reason we are using manual encoding, is to make sure that our analysis is not affected by misclassifications of the cascading classifiers.

When using Depth-first crawling, the overall process is very similar to the breadth-first approach, the difference is that instead of using a queue to hold the videos that are going to be explored we use a stack, so instead of evenly traversing the search space expanding one level at a time, we exhaust all expansions originated from a given video before looking at other expansions. Again, we will repeat the expansion process until we reach the 300 videos quota.

The last approach used for the crawling is based on an heuristic called focused crawling [39]. In this method, instead of having a fixed algorithm dictating the order in which the frontier is going to be expanded, a ranking mechanism is used to dynamically determine the order at each expansion step. There are many ways to rank the elements in the frontier, and this task is greatly domain-dependent. For our system, we trained a SVM classifier using only textual features to decide if a video was in SL or not (e.g. a text-based SL detection classifier). Then, for each video that we have in our frontier, we have also a ranking score that is determined as the given posterior probability obtained from the SVM of this particular video being a SL video. A priority queue is then used to select the next video in which expansion is going to happen. Five of the related videos are all assigned a ranking score using the trained SVM and these videos are incorporated into the frontier and priority queue. Again, this iteration will occur until the required number of videos was collected. With this approach, we hope that the frontier will expand in directions that are more promising in terms of finding new SL videos, and the ranking score is used as the indicator of how good that direction is.

To compare each one of the three approaches, we compute the percentage of relevant videos found by the crawler, and the variety of SL videos found, in terms of different channels and video categories found.

6.2 Multimodal fusion techniques

As we previously discussed there are multiple ways for combining multimodal features into a single classification output. Thus, to explore RQ2 we have performed the classification task of detecting sign language using different combination approaches and several ways to represent textual features. For each particular configuration, we run 1000 iterations of the classification task using random sampling on each iteration to separate our corpus into training and testing. We use then the metrics of precision, recall, and F1 score to compare each approach. All the conducted experiments used a subset of the ASL vs. Non-ASL dataset presented in [8], this subset contained all of the videos from the original dataset that were still online on YouTube, and thus, could have metadata collected. This resulted in a dataset with 99 SL videos and 106 non-SL videos.

The first step necessary for doing the multimodal classification was to decide how to represent videos' metadata as textual features. So, we performed a comparison of the classification task when using each one of the three candidates feature representations (TF-IDF, LDA, NMF). To ensure a fair comparison between them, and also a baseline comparison with video features, we set the feature vector size to six for all three techniques. For TF-IDF, the final feature vector was obtained by applying PCA to the full TF-IDF matrix, this was the same dimensionality reduction procedure used in the original PMP approach. For LDA and NMF, we set their corresponding *K* values (which indicates the number of topics to extract in each approach) to six. The LDA features were computed from raw term frequency counts, and NMF from the full TF-IDF matrix.

After experimenting with the three textual feature representations in the first step, the two best, TF-IDF and NMF, were used in the different fusion approaches tested. For early fusion two configurations were tested. The first consisted of a multimodal feature vector with 12 dimensions, and the other one of a multimodal feature vector with 6 dimensions. In the first strategy, we simply merged the unimodal feature vectors, so the resulting vector would contain 6 visual features (PMP), and 6 textual features (TF-IDF or NMF). The increased number of dimensions might cause problems of over-fitting the classifier, thus, in our second strategy, we compressed the multimodal feature vector down to six dimensions. This was done reducing PMP and TF-IDF features down to three dimensions using PCA, while for NMF the *k* value was set to three. The resultant multimodal feature vector would then contain three visual features and three textual features.

Lastly, we have experimented with two late fusion approaches. In the first one, we trained two unimodal SVMs (one with visual features and one with textual features) for the classification task, then the posterior probabilities obtained from these SVMs were used as the basis of a semantic feature space. For our experiment, each unimodal SVM outputs two probabilities, representing the posterior probability that a given video is a SL video or a non-SL video. A third SVM is then trained on this 4-dimensional feature space; two dimensions from the textual SVM probabilities and two dimensions from the visual SVM probabilities. The final classification of a test video is then the class with the highest posterior probability in the semantic SVM. Figure 13 displays an overview of this process.

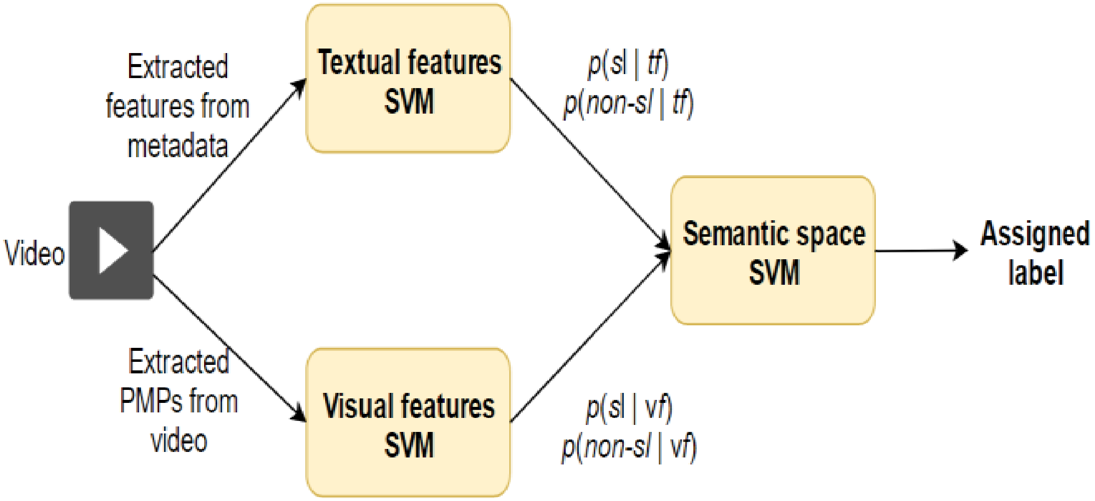


Figure 13. Overview of the classification process when using a semantic SVM, *tf* and *vf* represent textual and visual features, respectively.

For the second late fusion approach, we used the product rule of posterior probabilities. In this configuration, the posterior probabilities from each mode and class are multiplied, and the video is assigned to the class which yielded the highest multiplication value as shown in Equation 5:

$$\omega = \underset{k=sl,non-sl}{\arg\max}\ (p(\omega_k \mid x_t)p(\omega_k \mid x_v)) \tag{5}$$

where $p(\omega_k \mid x_t)$ is the posterior probability of sample *x* belonging to class *k* based on textual features, and $p(\omega_k \mid x_v)$ is the posterior probability of this sample belonging to class k based on visual features. This approach has the advantage that no extra SVM has to be trained other than the unimodal ones.

Once all the experiments were done we were then able to select the best set up of textual features and fusion technique to be used in the cascading classifiers.

6.3 Cascading classifiers configurations

In our last set of experiments, we wanted to assess how much computational time we could save while still maintaining the accuracy of the classifier. To this end we have explored several confidence prediction thresholds and two different arrangements for the cascading pipeline, one with full video segment as the last stage and another one where the frame sampling is the last stage. For all the cascading experiments, we have disabled the rule-based classifier, given that it is embedded in the crawler and that our current collections are not large enough to properly evaluate the rule-based system. In addition to the ASL vs. Non-ASL dataset that was used in the previous experiments we have also experimented with a three-class dataset. This new dataset was built simply adding a collection of 92 BSL videos to the previous ASL vs. Non-ASL dataset. The goal of experimenting with this dataset was to evaluate the performance of the cascading classifiers when dealing with multiple languages at the same time.

In this three-class scenario we have also experimented with using two sequential cascades, where the first one would focus solely on detecting sign language, while the second one would focus on identification. Therefore, in our three-class dataset all of the videos would go through the first cascading, and then just the ones classified as a SL video would go through the second cascading, that would differentiate between ASL and BSL. Table 7 summarizes all the cascading classifiers examined.

Table 7. Summary of all combinations of cascading evaluated.

	Single cascading	**Sequential cascading**
3 stages cascading	Configuration A	Configuration C
2 stages cascading	Configuration B	Configuration D

7. RESULTS

In this section we report and discuss all the results from the experiments described in Section 6. To ensure a fair analysis of computational times, all the experiments were performed on the same machine, a desktop computer equipped with a quad-core AMD A10-7800 processor and 8 GB of RAM. At the end of the analysis of all sub-components of the SLaDL system, we define the details of our final instantiation of this digital library.

7.1 Crawling strategies

Crawling video sharing sites collections to find potential new videos for the SLaDL corpus is a vital part of our proposed system. Given the rate of growth of video sharing sites and the time required to process a video, it is important for the crawler to bring as few irrelevant videos as possible, that way, more computational effort can be spent on classifying resources that will be part of the corpus.

As previously explained, we have two main goals for our crawler, it should find good candidate videos (i.e. videos that are in sign language), and also diversify its results, so at any given point in time, our SLaDL corpus could be treated as a subset of all sign language videos posted on video sharing sites, including the expected variety and distribution of video types and topics. To evaluate alternative designs, we created two initial scenarios to analyze each of the three crawling strategies. In the first one (positive seeds), all of the 6 initial seed videos are going to be sign language videos, more specifically, ASL videos. For the other scenario (mixed seeds), our initial frontier is comprised of good videos and bad videos (e.g. videos not in sign language), half from each class. In this scenario we aim to emulate a real-world scenario, where not all videos in our frontier are going to be useful resources.

It is important to note that, because of the way that the related videos list from YouTube works, we have configured all the crawlers to never expand directly to other videos of the same channel, as doing so leads to a drastic decrease in the crawled videos' variety in all tested strategies.

7.1.1 Positive seeds

Running the crawler with positive seeds, the focused crawler and BFS were able to find a significant amount of SL videos. However, the DFS approach performed poorly, finding only two SL videos early on in the crawling process and then losing its track for the remainder of the execution. Table 8 details the aggregate results for the first 300 videos selected via each strategy. For BFS and DFS, some collected videos apparently had permission restrictions, and thus, could not be coded. The focused crawler found the highest number of SL videos, with more than 96% percent of returned videos being a SL video. BFS did not find as many SL videos, with about 47% of the results being SL videos, but its channel diversity was greater than the focused crawler, collecting videos from 239 channels as opposed to the 183 from the focused crawler.

Concerning SL identification, we can observe that majority of the videos are in ASL, and this is expected given that the initial seeds were all in ASL as well. Also, a large portion of found videos had no indications at all in their metadata of what language was being signed in the video. The focused crawler contained more than 60 videos without sign language identification, and the BFS crawler contained about 35 of these cases. Among the other sign languages present in the crawler videos were LSM and BSL for the focused crawler and BSL and Filipino Sign Language (FSL) for the BFS.

Table 8. Number of channels, SL and Non-SL videos returned by each crawling strategy when the initial frontier has just SL videos.

	Focused Crawler	**BFS**	**DFS**
Channels	183	239	185
SL videos	290	142	2
Non-SL videos	10	156	295

Considering the limited number of videos coded in this analysis, it is important for us to investigate the rate of growth of SL videos for each strategy, as we want a strategy that can provide a steady flow of good candidate videos to our automatic classifier. Figure 14 presents the number of crawled SL videos as we progressed with the crawling execution. We can observe that for the whole crawler execution, the focused crawler keeps its rate of growth stable, almost always finding a good seed video due to the usage of a ranking score. While BFS is also able to find a reasonable amount of SL videos, we can see that for certain periods of time the crawler simply stops finding new SL videos; these areas are the areas when the related videos of a non-SL video are being explored.

To conclude, we examined the distribution of YouTube categories presented in the video and the distribution of SL video types according to the proposed classification in section 3.2. Table 9 presents the results. For both strategies, educational videos made the largest portion of crawled SL videos. This can be partially explained because these videos tend to have a higher metadata quality than the other types of SL video, this interpretation being further strengthened by the higher dominance of this category/type in the focused crawler. Additionally, it might also be the case that this type of videos is really the majority of posted videos, which would not be surprising, considering their appeal to non-signers wanting to learn a SL and the indirect

Table 9. Distribution of categories/types for the crawlers when using positive seeds.

	Focused Crawler	BFS Crawler
YouTube categories		
Films & Animation	2.1%	0.7%
Music	3.8%	5.7%
Travel & Events	0.3%	1.4%
People & Blogs	23.2%	22%
Comedy	0.3%	0.7%
Entertainment	3.8%	8.5%
News & Politics	0.3%	0.7%
Howto & Style	5.9%	2.8%
Education	58.1%	47.5%
Science & Technology	0.7%	0%
Nonprofits & Activism	1.4%	7.8%
Movies	0%	2.1%
Sign Language video type		
Education	71%	51.4%
Storytelling	3.1%	7.7%
Vlog	11%	24.6%
Music interpretation	13.1%	10.5%
PIP/Bilingual	0.3%	0%
News	0.7%	4.9%
Other	0.7%	0.7%

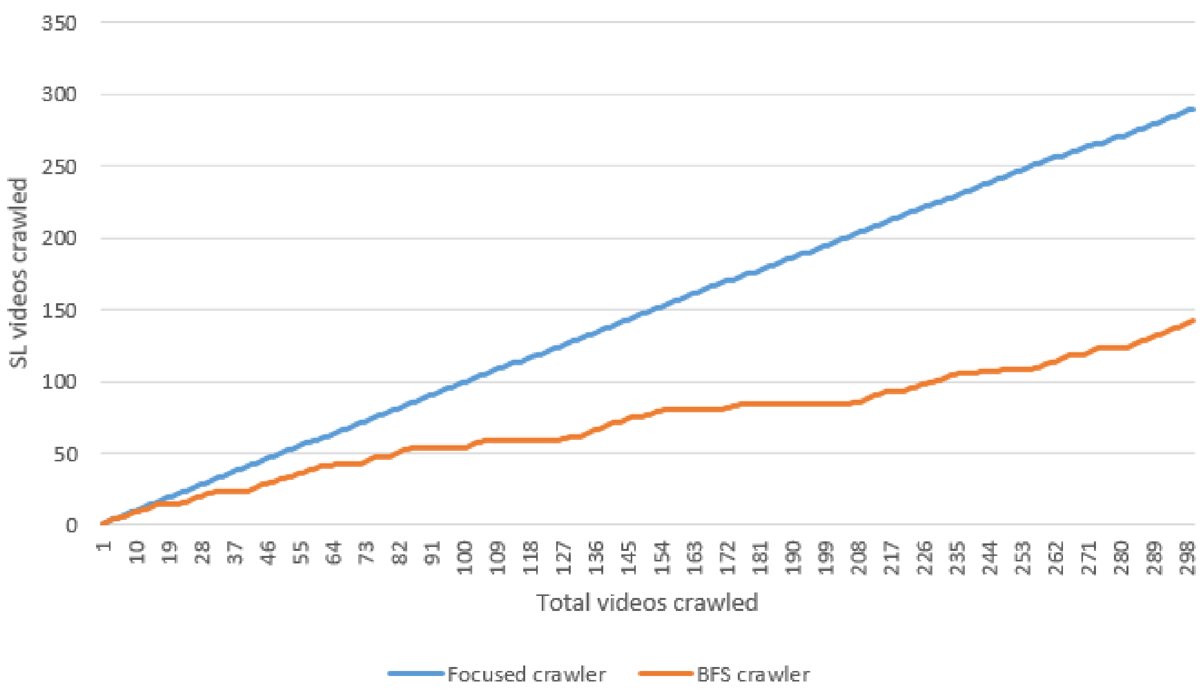


Figure 14. Progression of SL videos found by the crawlers throughout their execution when using positive seeds.

commercial value of these videos. It is also worth to note, that while the BFS crawler returned fewer SL videos, it was able to achieve a better balance on the variety of channels and categories/types.

7.1.2 Mixed seeds

In our second experimental scenario, the initial seeds of the crawler contain both SL and non-SL videos, a scenario that better represents what we would expect in a deployed system. Table 10 details the aggregate results. Once again, DFS performed poorly; this time it was not able to find a single SL video due to the fact that its first explored video was a non-SL video. BFS still managed to find almost a third of the total videos as SL videos, but the focused crawler had arguably the best performance, managing to find 295 SL videos out of the total 300 videos collected. This shows that the use of the ranking score can prevent the crawler from exploring less promising paths.

Table 10. Number of channels, SL and Non-SL videos returned by each crawling strategy when the initial frontier has both SL and non-SL videos.

	Focused Crawler	**BFS**	**DFS**
Channels	226	250	193
SL videos	295	94	0
Non-SL videos	4	206	300

Again, a majority of crawled SL videos were in ASL, with 2 videos being identified as BSL in the focused crawler and 1 in the BFS crawler. For this scenario, no other sign language could be properly identified through metadata in both strategies. However, a lot of videos had no indication of the sign language used, mainly for the focused crawler, which got 114 SL videos with no language identification. Examining these videos revealed that a large portion of them were part of a campaign to raise awareness of sign language communication and its importance that happened around 2016. The campaign encouraged everyone who knew any kind of sign language to share experiences about why they started signing, so many of these videos had metadata focused on the campaign, identified by the keyword #whyIsign, and no further identification of the sign language used.

Taking a look on the rate of growth of the focused crawler and BFS for the mixed seeds, we can observe that as the crawler progresses, there are more and more areas where the crawled videos are not relevant, as shown in Figure 15. Again, to conclude, we analyzed the distribution of categories and SL video types in the set of crawled SL videos. Education videos were the largest part of the crawled set once again, however, for the focused crawler strategy, a lot of results were also of the Vlog SL video type, mainly due to the previously mentioned videos for the #whyIsign campaign. Results are presented in Table 11.

As a note for the categories and SL video types distributions, since for these experiments we relied in the YouTube related videos feature, we are limited in the variety of videos we can reach. If for any reason, there are any biases in YouTube's algorithm that identifies related videos, they would manifest in our analysis.

Summarizing, when we take into account our two main concerns for the crawler (rate of growth and results variety), the focused crawler seems like the most promising approach. Its rate of collection growth was much more stable than the BFS approach, and the use of metadata features for ranking the candidate videos did not severely impair variety when compared to BFS results. However, such a strategy will certainly benefit categories and/or posting users that make

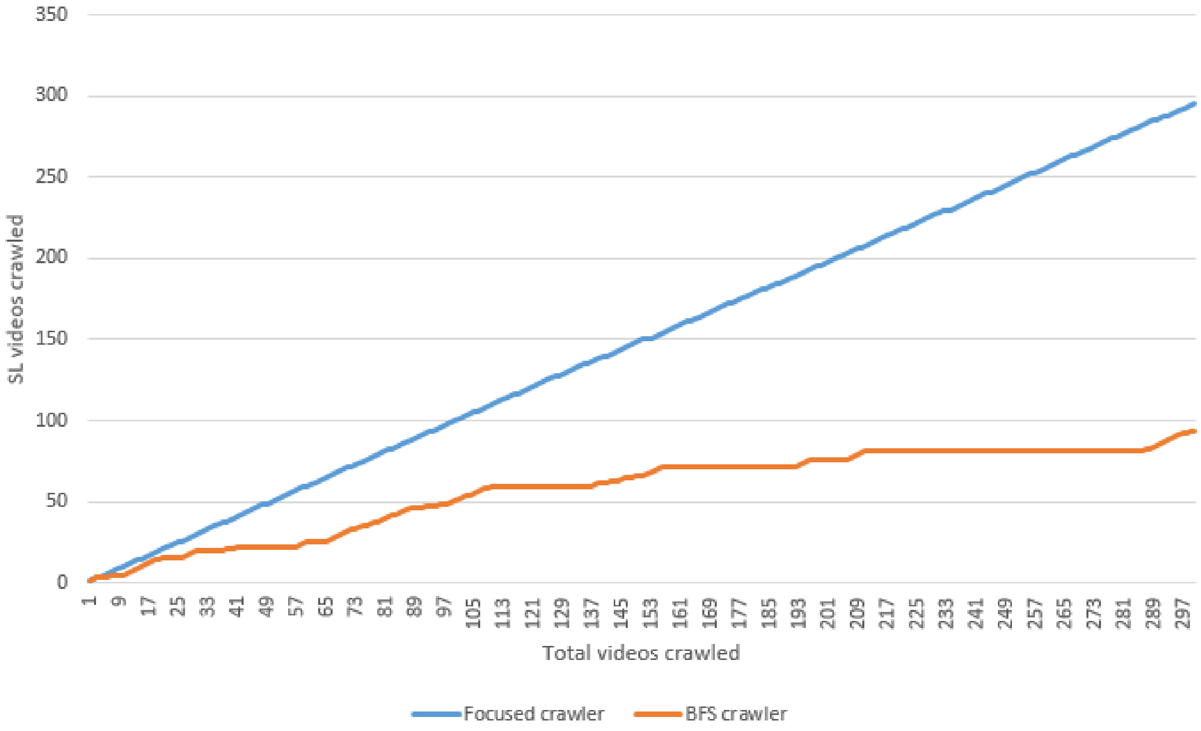


Figure 15. Progression of SL videos found by the crawlers throughout their execution when using mixed seeds.

Table 11. Distribution of categories/types for the crawlers when using mixed seeds

	Focused Crawler	BFS Crawler
YouTube categories		
Films & Animation	0%	0%
Music	3.4%	2.2%
Travel & Events	0%	2.2%
People & Blogs	31.5%	7.5%
Comedy	5.1%	3.2%
Entertainment	7.4%	11.8%
News & Politics	1.7%	0%
Howto & Style	3.1%	3.2%
Education	39.7%	48.4%
Science & Technology	0.3%	0%
Nonprofits & Activism	2.4%	1.1%
Movies	5.4%	1.1%
Sign Language video type		
Education	42%	38.3%
Storytelling	7.1%	8.5%
Vlog	44.1%	9.6%
Music interpretation	5.4%	39.4%
PIP/Bilingual	0.3%	1.1%
News	1%	2.2%
Other	0%	0%

at least some use of metadata annotation. As currently implemented, the DFS strategy is deemed basically useless for our task, given the ratio of non-SL videos to SL videos present in video sharing sites; once a bad candidate is explored it is extremely unlikely that the remaining of its path produce any good candidates. This shortcoming might be alleviated with the implementation of some early-stop criteria.

7.2 Multimodal classification

We began our analysis on the best way for combining visual and textual features by deciding which of the three textual features representation (TF-IDF, LDA, or NMF) performed best and should then be combined with the visual PMP features. Table 12, Table 13, and Table 14 presents the results in terms of precision, recall, and F1 score for each approach. It is interesting to note that LDA performance was much lower than the other two methods, mainly due to its low precision.

For the goal of developing our sign language digital library, it is important not only that the classifier finds sign language videos correctly (Precision), but we must be careful with false negatives, i.e. discarding SL videos due to misclassification (Recall). Therefore, to better visualize the overall performance of each method we took a look at their F1 scores. Figure 16 presents the average F1 score of each technique for the four different training set sizes used. As is observable, increasing the training set size steadily improved the performance in all techniques, particularly the TF-IDF approach, which at the largest training set size, performed almost as good as the NMF approach.

Table 12. Precision, Recall, and F1 score when representing textual features using LDA for different training set sizes.

	Precision	Recall	F1 score
15 training samples	50.62%	55.39%	51.41%
30 training samples	51.39%	60.61%	54.30%
45 training samples	51.45%	63.54%	55.80%
60 training samples	50.63%	66.75%	56.63%

Table 13. Precision, Recall, and F1 score when representing textual features using TF-IDF for different training set sizes.

	Precision	Recall	F1 score
15 training samples	82.68%	49.17%	57.51%
30 training samples	86.32%	57.23%	67.92%
45 training samples	86.44%	63.98%	72.98%
60 training samples	86.01%	68.06%	75.56%

Table 14. Precision, Recall, and F1 score when representing textual features using NMF for different training set sizes.

	Precision	Recall	F1 score
15 training samples	76.99%	71.92%	73.82%
30 training samples	79.39%	73.29%	75.85%
45 training samples	80.28%	74.38%	76.89%
60 training samples	80.81%	74.32%	77.09%

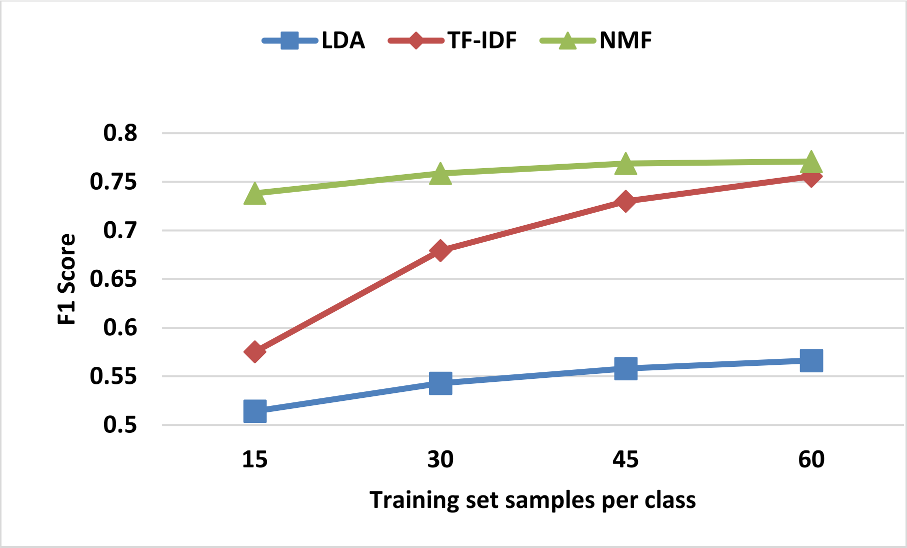


Figure 16. Average F1 score for each technique over different training set sizes.

When using PMP for the same classification task at the same dataset, trained with 60 samples from each class, it obtained 82.9% precision, 67.8% recall, and 74.2% F1. Thus, two of the proposed metadata features had a better performance than visual features. However, videos in our dataset were manually selected through a combination of user queries and related videos on YouTube. Consequently, many of these videos present in the corpus had been located via metadata at the first place, and therefore, the quality of metadata in our collection is probably higher than what can be expected throughout the whole corpus of YouTube. Additionally, the videos representing the non-SL class in our dataset were first collected with the purpose of analyzing the strength of PMP features, thus, many of them can be considered visually similar to SL videos.

Given the low performance of LDA, we discarded it as a candidate for being used in the fusion techniques, also, we limited the fusion experiments to 60 training samples per class. In the first early fusion configuration, we simply combined the computed textual and visual features of a video in a single feature vector with 12 dimensions. Table 15 presents the results.

Although just slightly, this strategy is able to improve the overall performance of the classifier. Following the same pattern from unimodal results, when using TF-IDF features in the combination we got a higher precision, while when using NMF features we got a higher recall and F1 score. Interestingly, PMP features alone had a much higher precision than recall, but in both cases, the combination results had a lower precision and higher recall than any of the unimodal features involved. This is particularly surprisingly, for the combination of TF-IDF and PMP, where the two unimodal approaches had considerably low recall.

Next, we wanted to assess how this combination approach would fare in comparison with the unimodal classifiers. To ensure a fair comparison, we compressed the multimodal feature vectors down to 6 dimensions. Table 16 presents the results.

Table 15. Average precision, recall, and F1 score for multimodal feature vectors with 12 dimensions.

	TF-IDF + PMP	**NMF + PMP**
Precision	**79.3%**	76.1%
Recall	77%	**82.7%**
F1 score	77.4%	**78.6%**

Table 16. Average precision, recall, and F1 score for multimodal feature vectors with 6 dimensions.

	TF-IDF + PMP	**NMF + PMP**
Precision	**84.7%**	83%
Recall	73.2%	**75.9%**
F1	78.1%	**78.9%**

The use of six dimensions improved both multimodal results when compared to the 12 dimensions version. Interestingly, precision at this time was maintained reasonably high, while recall suffered a small hit when compared to the 12 dimensions results but were still higher than when using unimodal features. Again, the combination of NMF and PMP lead to the highest recall and F1 score overall. When comparing it with the unimodal features involved (NMF for textual and PMP for visual), this fusion was also particularly successful, increasing the performance for both precision and recall.

While results for both six and twelve dimensions were really close one to another, in a scenario like this we should favor the representation with fewer dimensions, since it brings two advantages when compared to the bigger feature vector: (1) it takes less space in the memory to store these features, which are necessary for retraining the classification models of the SLaDL periodically; (2) while the training corpus is not very large, using more dimensions can increase the risk of overfitting our models. Therefore, if using early fusion for combining textual and visual features in our final instantiation, we would opt for the 6 dimensions representation.

We also experimented with two scenarios of late fusion, where the combination happens at the semantic space. For the first approach, we trained a semantic SVM responsible for deciding the class of new samples. Table 17 presents the results for this fusion approach. As shown, recall for both textual features representation increased at the expense of precision, when compared with the early fusion results. Nevertheless, overall F1 scores were higher than early fusion in both cases. The shortcoming of using such approach is that instead of training a single SVM as is the case with early fusion, we have to train *m+1* SVMs, where *m* is the number of modes used; in our case 2.

Table 17.Average precision, recall, and F1 score for multimodal classification using a SVM trained on posterior probabilities.

	TF-IDF + PMP	**NMF + PMP**
Precision	**82.8%**	81.9%
Recall	77.4%	**80.3%**
F1	79.7%	**80.8%**

The second late fusion method analyzed was the product rule, similarly to the semantic SVM method, we are still required to train unimodal SVMs, but not a semantic SVM. Table 18 presents the results for this technique. This time, the combination using NMF as the textual features outperformed the TF-IDF combination in every metric. F1 score obtained with this fusion was also the highest among all results, unimodal or fused.

Table 18. Average precision, recall, and F1 score for multimodal classification using the product rule.

	TF-IDF + PMP	**NMF + PMP**
Precision	85.8%	**86.4%**
Recall	80.6%	**81.2%**
F1	82.9%	**83.5%**

Summarizing, Figure 17 presents the reported results for each approach side-by-side (visual-only features, textual-only features, early fusion, and late fusion). Considering unimodal classification, NMF performed the best, with a 77% F1 score. The experimented fusion techniques have all improved on unimodal features, even if just slightly. Among the fusion techniques experimented, late fusion was superior to early fusion in both approaches tested, with the product rule combining NMF and PMP performing the best. The obtained 83.5% F1 score represents a 6% improvement from the best unimodal results (NMF) and a 9% improvement over the original visual-only PMP approach. These results indicate that multimodal classification can be really helpful in the SL domain.

Considering these results, in our current instantiation of SLaDL, we are using NMF to extract textual features from metadata, and late fusion with product rule to combine these features with visual features extracted from PMP. This combination strategy is also the one used for the experiments of the next section. Therefore, every time we mention a combination of visual and textual features from this point and forward, we are referring to this strategy, unless explicitly explained otherwise.

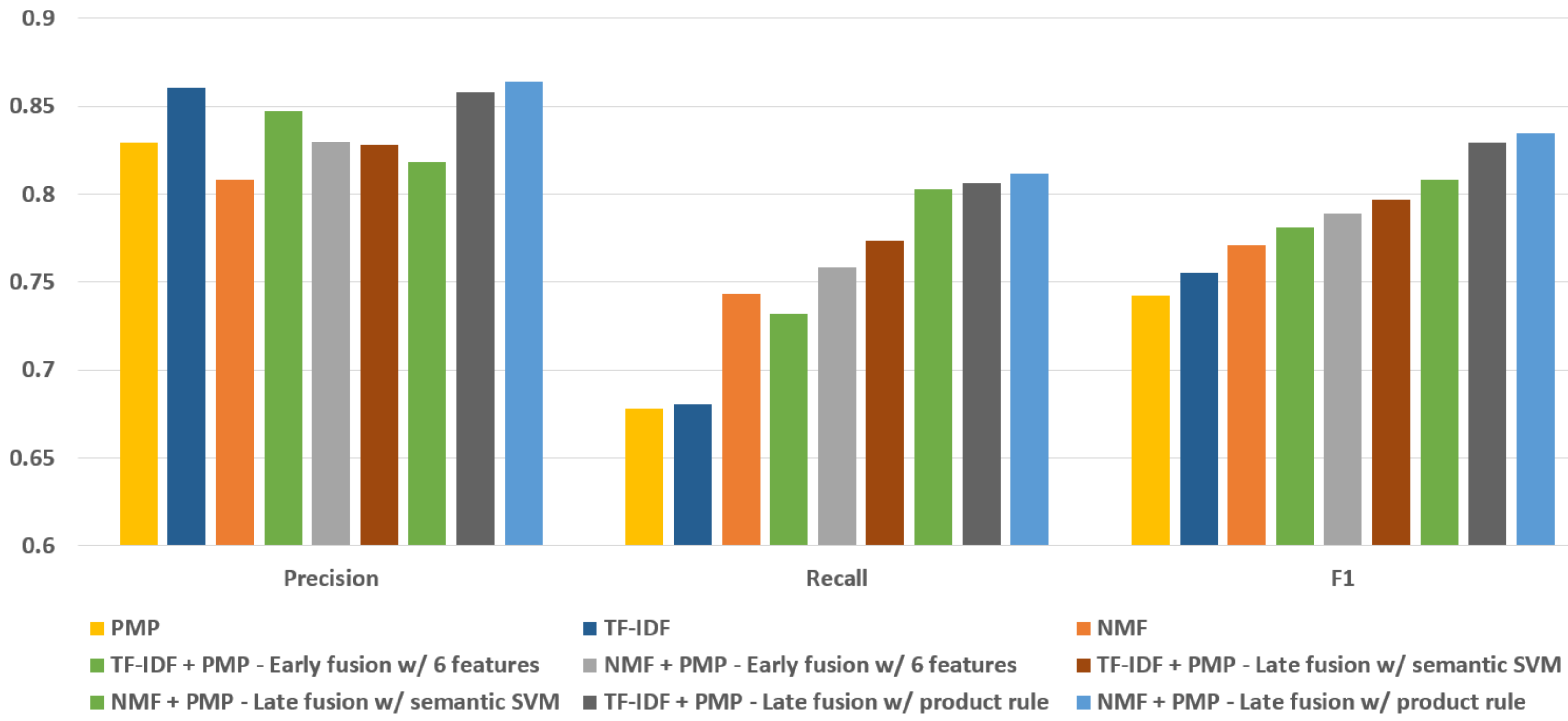


Figure 17. Precision, Recall, and F1 score for all the combinations approaches experimented and their corresponding unimodal results.

7.3 Cascading classifiers strategies

Once we figured out how to best combine textual and visual features, we analyzed different configurations of cascading classifiers to identify the best compromise between classification accuracy and processing time.

To start our analysis, we expanded the processing time assessment performed by Duggina [43] to consider not only face detection costs but all the costs involved for extracting visual features used in classification. For all three visual features representation used (Keyframe, Frame sampling, and Full segment processing), the time spent for processing a video can be divided in two sequential steps: first, face detection and foreground extraction are performed to get some basic representation of the video; then, the PMP algorithm is applied on this video transformation, resulting in the final feature vector used to represent the video in the classification system. Once textual and visual features are extracted the video can then be classified by a trained SVM. These other two steps (extracting textual features and classifying a new sample using the SVM) are considerably faster than visual features computation, thus, we can estimate the time to classify a new video as the sum of time required to perform face detection and background-foreground separation, plus the time spent generating PMPs. Table 19 presents the average time for these steps when processing videos in our testing dataset.

Table 19. Average processing time for different visual features representations.

	Keyframe	**Frame sampling**	**Full segment**
Face detec. + BG subtraction	7.22s	18s	183.4s
PMP	1.56s	211s	211s
Total	8.78s	229s	394.4s

As is observable, classification at the first two stages is much faster, with the frame sampling taking a little more than half of the full segment processing time, while the Keyframes approach provides a processing time roughly 44 times lower. It is interesting to note that while frame sampling provides a 10-fold gain when performing face detection and background subtraction, this advantage is reduced by the fact that computing PMP for this technique demands the same effort as full video segments. The reason for that is that sampling is limited to the face detection portion of the processing, thus, the background subtraction algorithm is still applied to all the frames of the video segment.

When a video is classified at one of these three stages, we can assume that the time spent for processing this video is the time spent in the current stage, plus all the time spent in previous stages. Considering this, videos classified in the first two stages are actually being classified faster than if just the full video segment approach was used with no cascading. On the other hand, if a video needs the last stage for classification, it would take longer to classify it with the cascading than without it.

Figure 18 compares the average processing time of a video if it is classified at different stages, along with a baseline comparison with the non-cascading version. Because the keyframes approach executes so fast, even for videos classified at the frame sampling stage the total time spent for processing the video is much lower than a non-cascading approach using full video segments, 237.78s compared to 394.4s. This means that for every two videos classified at the second stage, we are already able to classify a video at the last stage without increasing the average time due to the cascading. In fact, with this rate we would still obtain an average processing time of 366.32s per video, about 28s less than the non-cascading average. Once we factor in videos being processed at the first stage, the computational efforts could be much more prominent.

Therefore, for the cascading approach to be valuable, a sufficient amount of videos have to be classified at the first two stages. Videos can only receive a final classification at a given stage if both metadata and visual features agree on a class for the video, and if the computed confidence in the prediction is above a set threshold. The computed product from the selected multimodal strategy fits nicely to our definition of prediction confidence, thus, at any given stage where the modes agree on a class and the product rule value is greater than the threshold, we will obtain the final classification for the video and no further processing is required.

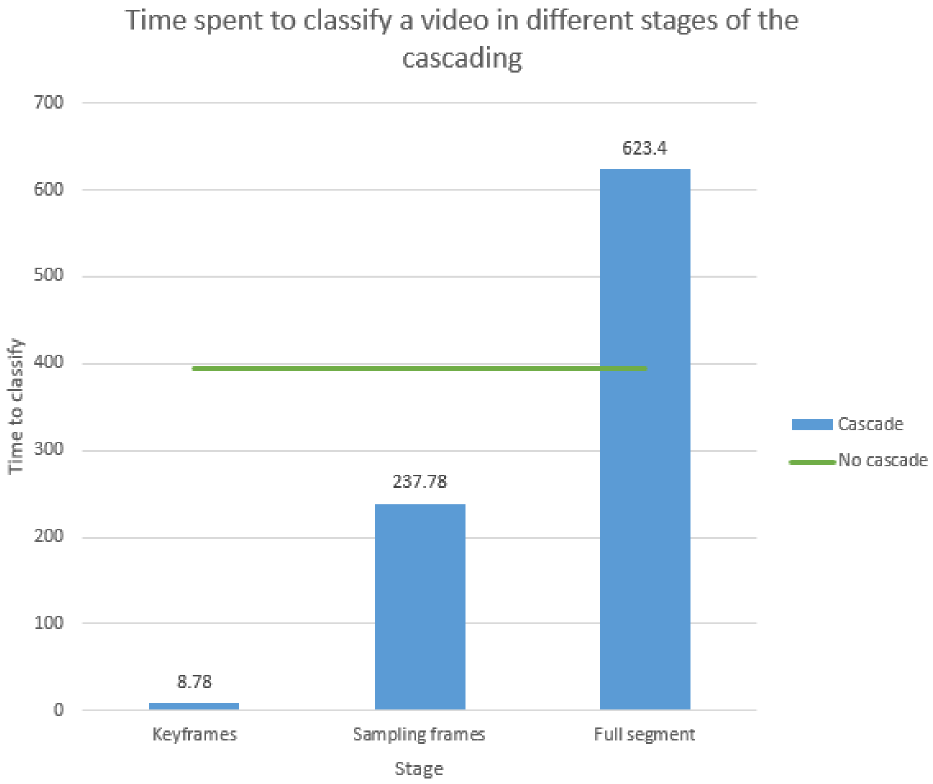


Figure 18. Time required to classify a video at different stages of the cascading of classifiers in comparison to a non-cascading approach.

We explored how the different cascading configurations perform when the decision threshold for the keyframe and sampling stages vary. First, we analyzed Configuration A (single cascading with 3 stages) on the ASL vs non-ASL dataset. Figure 19 shows the average time spent to process a video as the threshold required for classification increases. It is interesting to note that for thresholds up to 30%, the average processing time is less than half of the expected time of the non-cascading approach using full video segment, and even on a 40% threshold, the average time spent is just 230s, basically the same as the processing time for the sampling stage alone. We can observe that thresholds higher than 50% result in an average processing time longer than the non-cascading approach. This is caused by the fact that with the higher threshold, fewer videos are able to be classified at the first stages, leaving more videos to be classified by the most time-consuming stage, and consequently, increasing the average time per video.

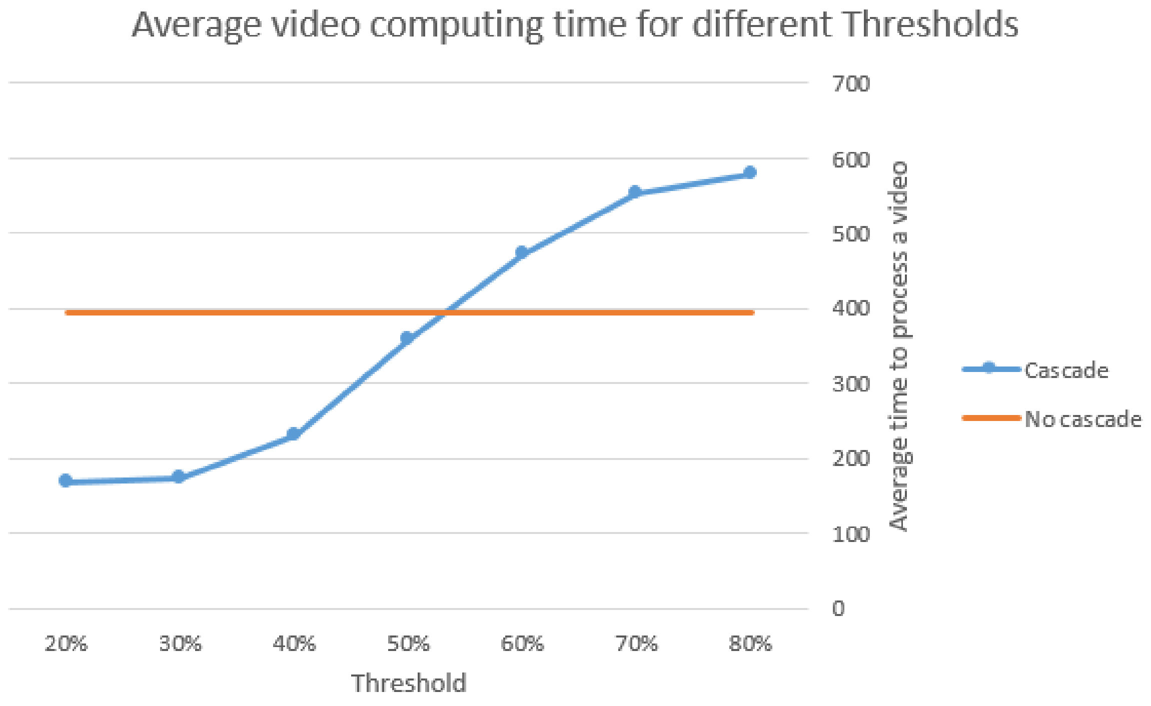


Figure 19. Average time spent to process a video for different confidence thresholds in Configuration A.

A closer look at the average number of videos classified at each stage confirms this, as shown in Figure 20. For thresholds of up to 40%, more of the videos are being classified at the keyframes stage than the other stages. This is the reason for the computational savings when using this threshold. Then, starting at the 50% mark, more of the videos reach the last stage for classification, but even then, the amount of videos being classified at the earlier two stages offset the overhead caused by the last stage classification. Once we set the threshold for values greater than 50%, then the number of videos classified at the first two stages rapidly decreases, which explains the cascading average time being greater than the non-cascading approach.

With respect to the cascading classifier's performance, we have analyzed the impact of varying the threshold in the measures of precision, recall, and F1 score. As a baseline, we first present the results for the non-cascading multimodal approach and the corresponding unimodal classifiers. This ASL vs non-ASL dataset is the same used for the experiments of sub-section 7.2

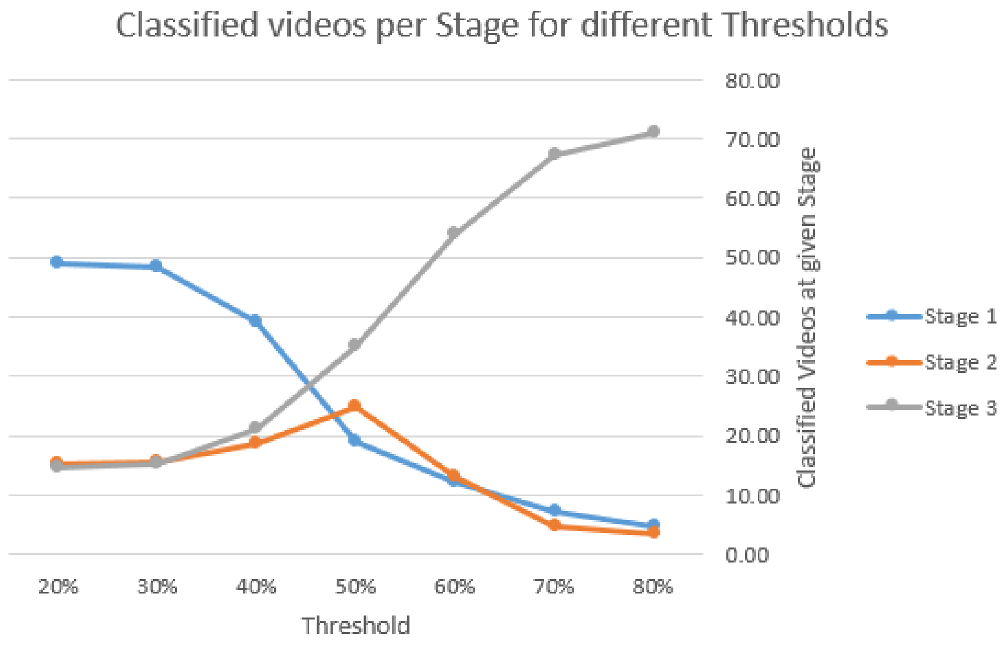


Figure 20. Average number of videos classified at each stage for different thresholds in Configuration A.

but at the time of this new set of experiments, a few videos were removed from YouTube, making it impossible to collect their metadata for the multimodal classifier. Therefore, to ensure a proper comparison we ran the non-cascading approach in the same subset of videos used for the cascading experiments. Figure 21 displays the results.

The effects of varying the decision threshold on precision, recall, and F1 score are shown in Figure 22. Overall, the F1 score barely changed as the decision threshold increased from 50% to 80%, indicating that the additional time spent with higher thresholds does not translate to a better classifier performance. For the thresholds up to the 50% mark, we can observe that the F1 score slowly increases, but even with the lowest threshold of 20%, the F1 score was just about 2% below the non-cascading multimodal approach, although this is the result of a combination of a ~3% increase in precision and ~7% decrease in recall. Overall, the results indicate that with just a small hit on the classifier performance, we can classify a new video in less than half of the time required by the non-cascading approach, on average. Not only that, but using thresholds of up to 30% would even be faster than a non-cascading classifier using the frame sampling approach.

Given how similar was the performance of the frame sampling approach and the original PMP technique as described in section 4.3, we have created the Configuration B for the cascading of classifiers. In this configuration we disable the full video segment processing and all decisions must be made at latest in the frame sampling step. We conducted the same analysis as for Configuration A. Figure 23 displays the average processing time for a video under the different thresholds. In this chart we have not only the baseline for non-cascading using full segment processing but also a baseline for a non-cascading version using the frame sampling method.

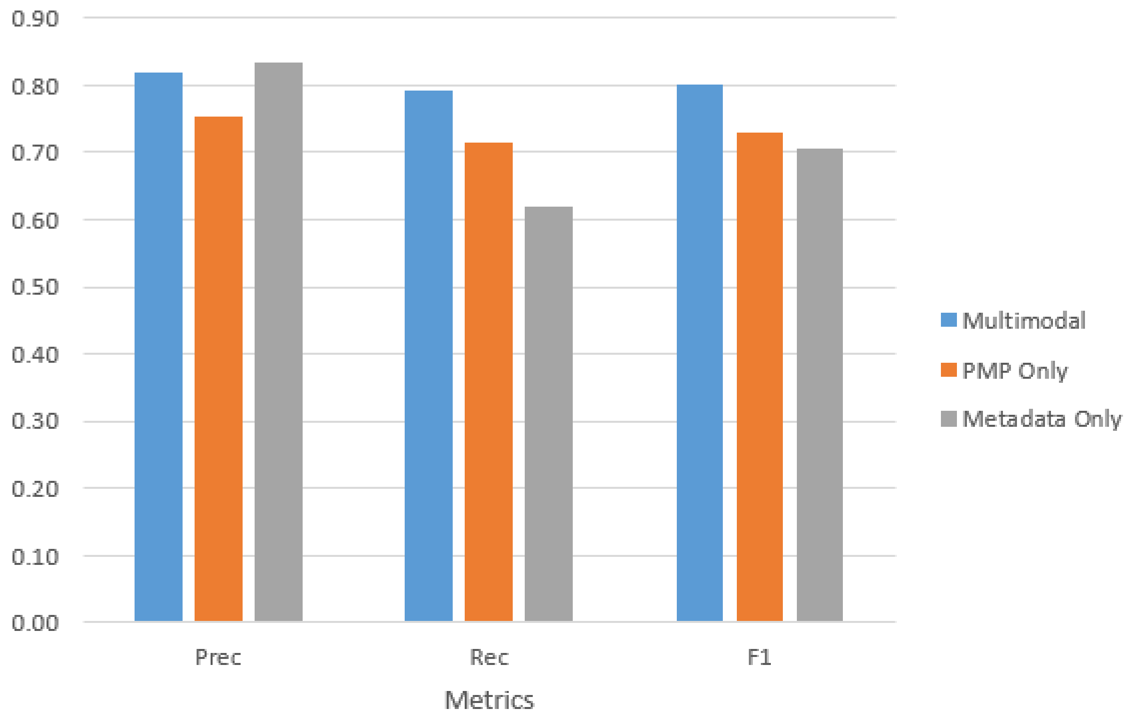


Figure 21. Precision, recall, and F1 score for non-cascading approaches.

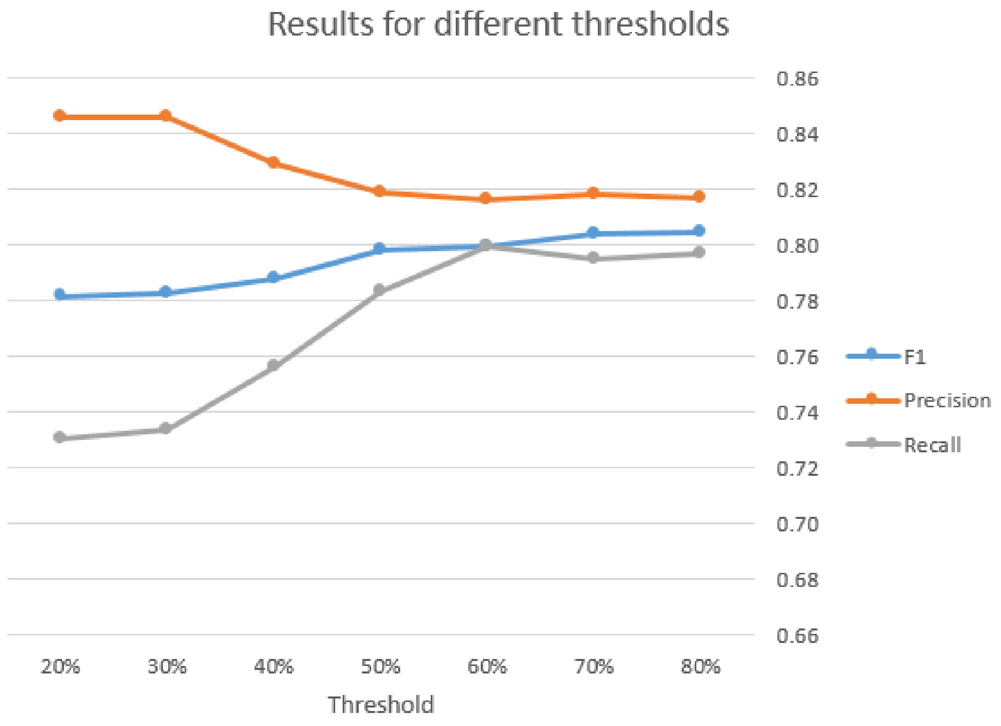


Figure 22. Precision, recall, and F1 score for Configuration A of the cascading classifiers using different thresholds.

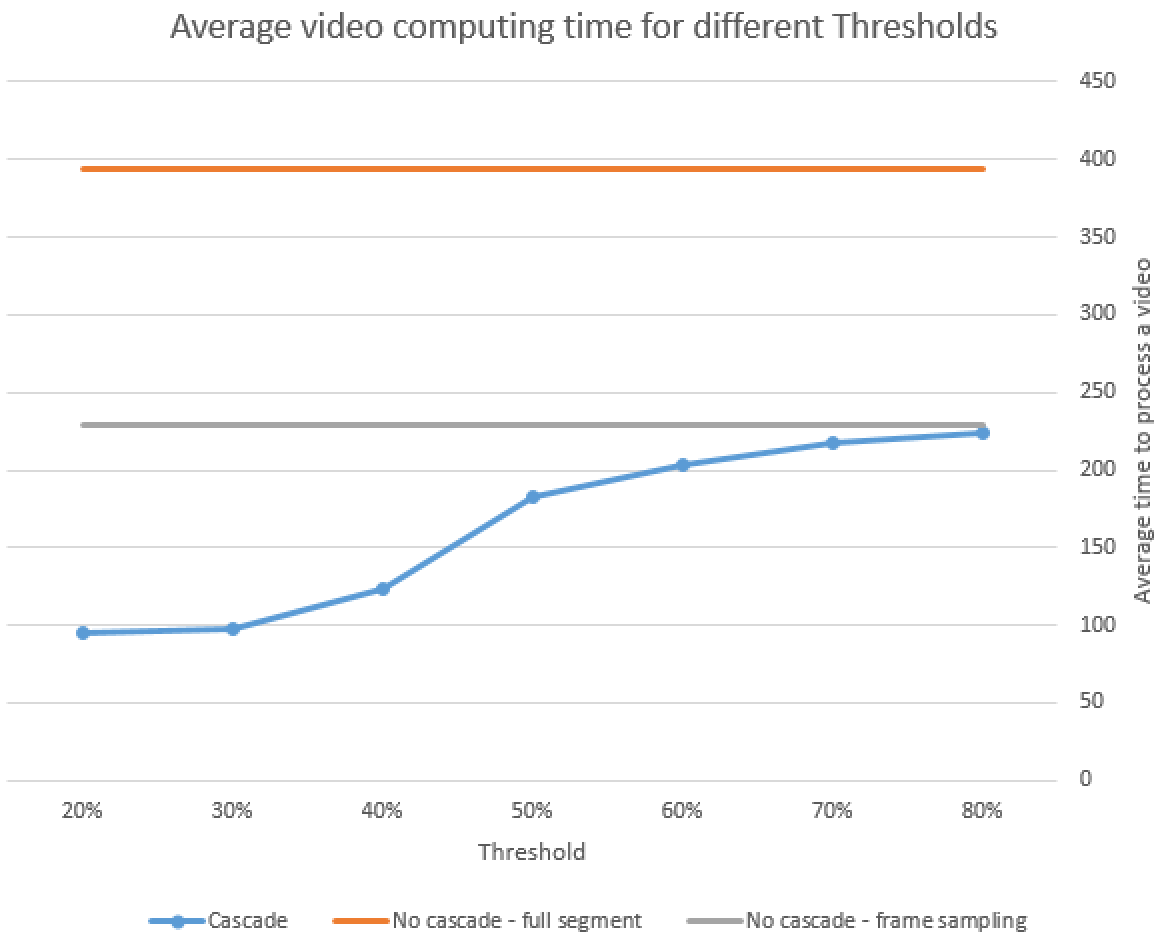


Figure 23. Average time spent to process a video for different confidence thresholds in Configuration B.

Since we are never doing full segment processing, no matter how high is the threshold, we are always going to have an average time much lower than the non-cascading version with full segment processing. Even for the extreme case were all the videos are just classified at the last stage, our average runtime would be 238s, almost half of the processing time required by the full segment processing. Furthermore, because the processing time for the keyframe stage is so low, even for the highest threshold experimented, the cascading average time is still slightly lower than frame sampling alone. As for the amount of videos classified at each stage, the only difference when compared to Configuration A is that all the videos that were being classified at the full segment stage before, are now classified at the frame sampling stage, as displayed by Figure 24.

As for the performance of the classifier, it remained basically the same, as is observable in Table 20. This indicate that, at least for detecting sign language, a 2 stages cascade performs as good as the more expensive 3 stages configuration while spending much less computational time. It is also worth to mention that in both configurations there is a trend of increasing recall and decreasing precision as the threshold gets higher. For a system like SLaDL, a higher precision means that fewer irrelevant videos are going to be present in the corpus, while a lower recall means that we are discarding good videos off the collection corpus. Thus, we need a balance between these two.

Table 20. Average precision, recall, and F1 score for Configurations A and B of the cascading classifiers.

	3 stages cascade			**2 stages cascade**		
Threshold	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**
20%	84.6%	73%	78.1%	84.7%	73.7%	78.6%
30%	84.5%	73.3%	78.2%	84.7%	73.9%	78.7%
40%	82.9%	75.6%	78.8%	83%	76.3%	79.2%
50%	81.9%	78.3%	79.8%	82.3%	78.8%	80.3%
60%	81.6%	78.9%	80%	82.6%	79.2%	80.6%
70%	81.8%	79.5%	80.4%	82.5%	79.7%	80.8%
80%	81.7%	79.7%	80.4%	82.4%	79.9%	80.9%

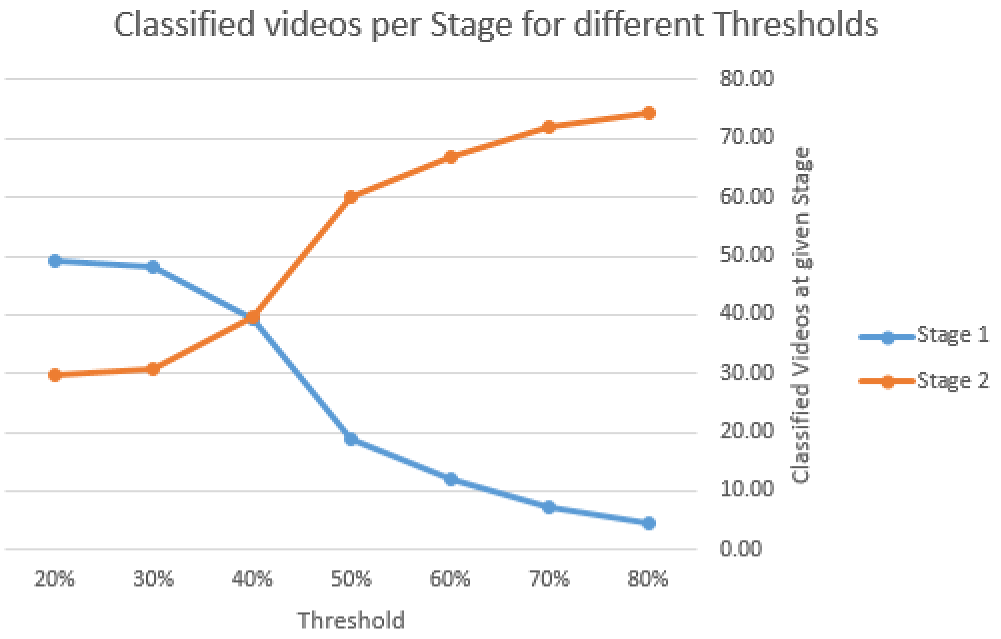


Figure 24. Average number of videos classified at each stage for different thresholds in Configuration B.

Comparing the two stages configuration with the best non-cascading approach we had (multimodal classifier using full segment processing), if we set the cascading threshold to 50%, we are obtaining once again comparable results, 80.3% F1 score for both cases, while reducing the average computing time by more than a half, from 394s in the non-cascading approach to 182s in Configuration B of the cascading.

The SLaDL system will contain videos from multiple sign languages, not only ASL. Therefore, it is important that the automatic classifier is capable of not only detecting SL but also identifying the presence of a particular sign language in a video. To explore that, we combined the ASL vs Non-ASL dataset used in the previous experiments, with the BSL videos collected from YouTube and used in the work described at section 4.4. In this multiclass problem, beyond the two cascading configurations previously mentioned (A and B), we have analyzed another two configurations, C and D, that separates detection and identification in two different steps. Table 21 presents the results for Configuration A in this three classes problem.

Table 21. Confusion matrices for multiclass classification using Configuration A and different prediction confidence thresholds.

	ASL	**Non-ASL**	**BSL**
20% Threshold **(Overall accuracy = 59.1%)**			
ASL	43.9%	20.4%	35.7%
Non-ASL	10.5%	65.2%	24.3%
BSL	23.9%	7.8%	68.3%
30% Threshold **(Overall accuracy = 58.6%)**			
ASL	44.8%	19.5%	35.7%
Non-ASL	11.6%	63.1%	25.3%
BSL	24.2%	7.8%	68%
40% Threshold **(Overall accuracy = 59%)**			
ASL	45.4%	20.4%	34.2%
Non-ASL	11.6%	63.5%	24.9%
BSL	24%	7.9%	68.1%
50% Threshold **(Overall accuracy = 58.7%)**			
ASL	45.2%	20.5%	34.3%
Non-ASL	11.9%	63%	25.1%
BSL	24.2%	7.9%	67.9%

As is observable, increasing the threshold does not increase the performance of the classifier, with the accuracy remaining basically stable around 59% for all the thresholds tested. Analyzing the number of videos classified at each stage on Figure 25, we can observe that even for the lowest threshold of 20%, a vast majority of the videos are classified just at the latest stage, this is justified by the fact that this classification task is considerably harder than just detecting sign language, and thus, prediction confidences are lower. In fact, the number of videos classified at the first two stages are so low that even for the 20% threshold, using the cascading approach with configuration A results in an average processing time higher than the non-cascading approach, 513 seconds. All this extra processing time, however, does not translate to a better performance, with this cascading configuration achieving a lower overall accuracy than the non-cascading approach (59% accuracy) for all but the 20% threshold.

Table 22 presents the results when using Configuration B. While overall accuracy is slightly lower than for Configuration A, its average processing time is much faster, given the absence of full segment processing.

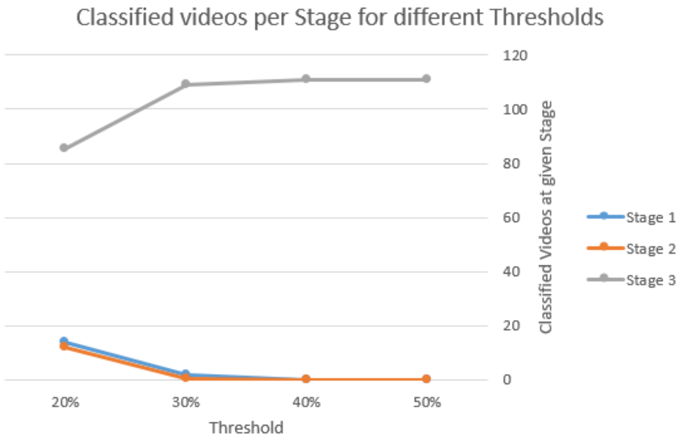


Figure 25. Average number of videos classified at each stage for the multiclass problem and different thresholds in Configuration A.

Table 22. Confusion matrices for multiclass classification using Configuration B and different prediction confidence thresholds.

	ASL	Non-ASL	BSL
20% Threshold **(Overall accuracy = 58.2%)**			
ASL	44.8%	19.9%	35.3%
Non-ASL	12.4%	62.6%	25%
BSL	24%	8.7%	67.3%
30% Threshold **(Overall accuracy = 58.1%)**			
ASL	46.9%	18.4%	34.7%
Non-ASL	13.6%	60.3%	26.1%
BSL	23.8%	9%	67.2%
40% Threshold **(Overall accuracy = 58.4%)**			
ASL	47.1%	18.9%	34%
Non-ASL	13.5%	60.5%	26%
BSL	23.9%	8.6%	67.5%
50% Threshold **(Overall accuracy = 58.1%)**			
ASL	46.8%	19.2%	34%
Non-ASL	14.1%	60.4%	25.5%
BSL	24.4%	8.6%	67%

For Configuration B we still have the majority of the videos being classified at the last stage. However, because the last stage is now the frame sampling, we guarantee that our processing time is much shorter than a non-cascading approach based on full video segment. Figure 26 compares the average processing times for this configuration and non-cascading approaches using video segment or frame sampling alone. Because very few, if any, videos are able to get a classification at the first stage, the average processing time for Configuration B is very similar to a non-cascading approach using frame sampling, with the cascading approach being about 10% faster with the lowest threshold and 4% slower for the highest one.

Considering that the main problem with the cascading approach in this problem was that very few videos were able to actually benefit from a faster processing, we developed two other configurations that would perform detection and identification of sign language in two sequential cascades. Comparing the number of videos classified per stage between the ASL vs Non-ASL problem and this one (ASL vs Non-ASL vs BSL), there is a strong evidence that detection of sign language (e.g. filter out videos that are not in sign language) can be performed much faster than identification of a particular language. Considering this, in these two last configurations, we focus on the detection problem first, filtering out non-sign language videos as quickly as possible, to then proceed to the identification task, where just a smaller set of videos are left to be classified. Another advantage of this method, is that all the processing done for the detection cascading, is reused in the identification cascade (e.g. if a video had PMP features extracted from the frame sampling approach in the detection cascading, the same features can be used in the identification cascade). Figure 27 displays the average number of videos classified per stage in this approach. If we compare this with the results from Figure 25, we can clearly observe that a reasonably larger portion of videos are being classified before the last stage. The confusion matrices for Configuration C are presented in Table 23.

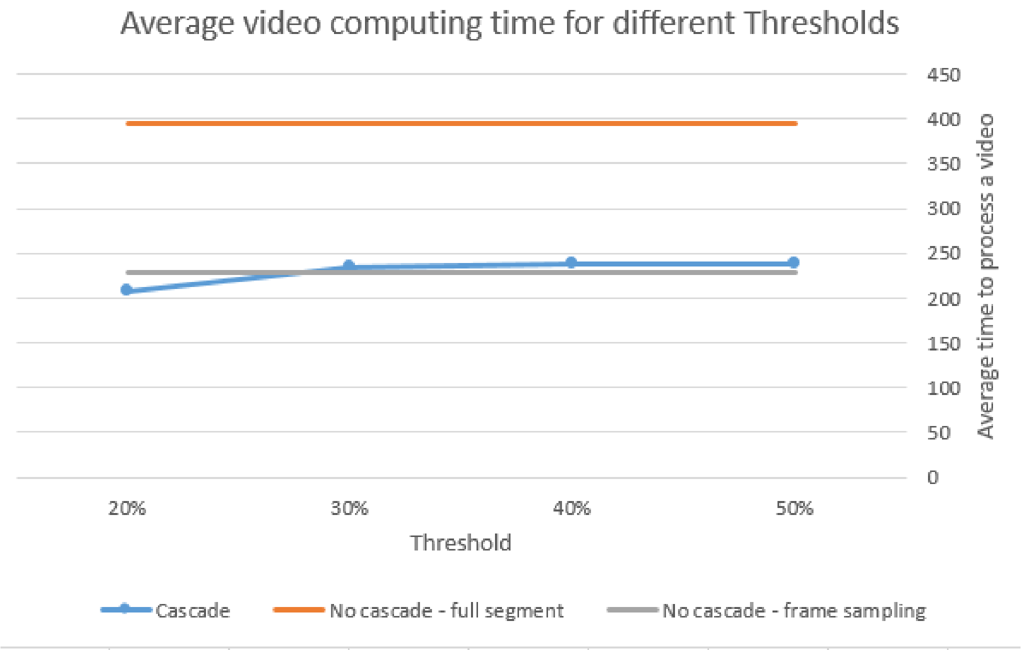


Figure 26. Average time spent to process a video with Configuration B in the multiclass problem for different confidence thresholds.

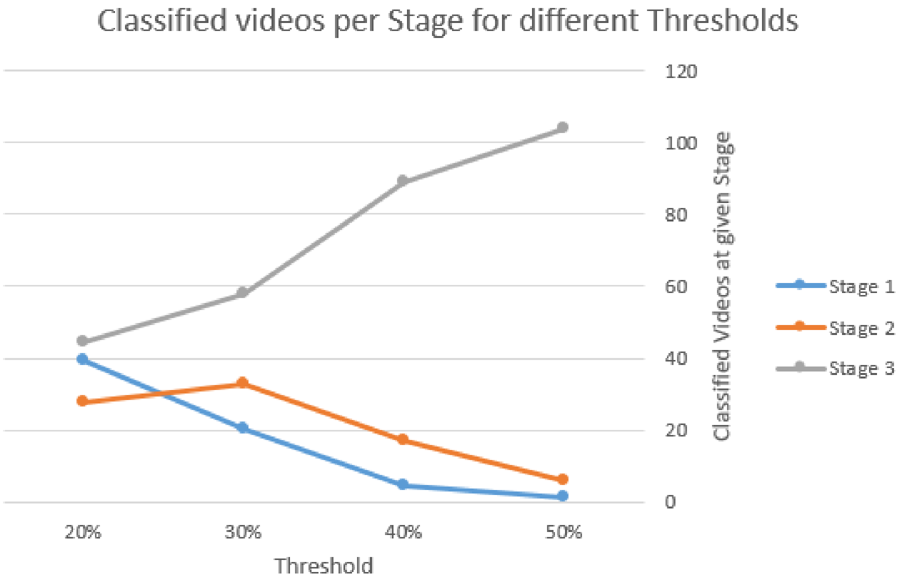


Figure 27. Average number of videos classified per stage with Configuration C in the multiclass problem for different confidence thresholds.

Table 23. Confusion matrices for multiclass classification using Configuration C and different prediction confidence thresholds.

	ASL	Non-ASL	BSL
20% Threshold (Overall accuracy = 55.2%)			
ASL	61.7%	4%	34.3%
Non-ASL	34.8%	43.7%	21.5%
BSL	37.1%	2.8%	60.1%
30% Threshold (Overall accuracy = 57.5%)			
ASL	64.7%	3.8%	31.5%
Non-ASL	31.5%	45.8%	22.7%
BSL	35%	3.1%	61.9%
40% Threshold (Overall accuracy = 59.9%)			
ASL	62.5%	5.2%	32.3%
Non-ASL	22.9%	52.8%	24.3%
BSL	31.3%	4.4%	64.3%
50% Threshold (Overall accuracy = 60.2%)			
ASL	62.4%	6%	31.6%
Non-ASL	22.4%	54.1%	23.5%
BSL	31.4%	4.6%	64%

As opposed to what we observed in Configurations A and B, for Configuration C the overall accuracy of the classifier increases alongside with the threshold. This increase is mainly due to the Non-ASL videos, indicating that with a low threshold, the chances of a false-positive are greater in the SL detection cascading. It is interesting to note that while the performance of this configuration can be better than the non-cascading approach, this only happened for the thresholds of 40% and 50%, where the processing time for the cascading is still considerably higher than the non-cascading version, as is observable in Figure 28.

Lastly, we experimented with Configuration D, which contained separate cascades for detection and identification but disabled the full video segment processing in both cascades. Table 24 present the results.

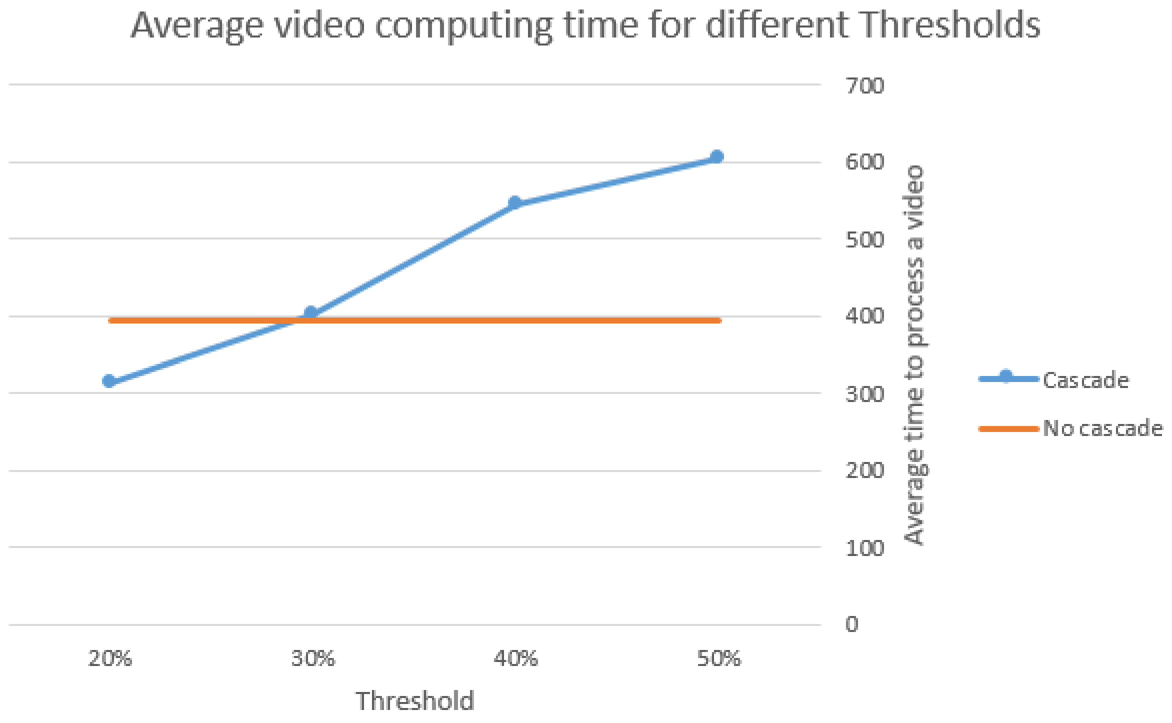


Figure 28. Average time spent to process a video with Configuration C in the multiclass problem for different confidence thresholds.

Table 24. Confusion matrices for multiclass classification using Configuration D and different prediction confidence thresholds.

	ASL	Non-ASL	BSL
20% Threshold (Overall accuracy = 54.8%)			
ASL	63.2%	4.2%	32.6%
Non-ASL	36.8%	41.8%	21.4%
BSL	37.7%	2.8%	59.5%
30% Threshold (Overall accuracy = 56.7%)			
ASL	65.6%	4%	30.3%
Non-ASL	34.3%	43.9%	21.7%
BSL	36.6%	3%	60.4%
40% Threshold (Overall accuracy = 60%)			
ASL	64.6%	5.2%	30.2%
Non-ASL	25.7%	52.2%	22.1%
BSL	33.1%	3.6%	63.3%
50% Threshold (Overall accuracy = 60.1%)			
ASL	64%	5.7%	30.3%
Non-ASL	24.5%	53.1%	22.4%
BSL	32.8%	3.9%	63.3%

Like with configuration B, the absence of the last stage makes this one much faster than the non-cascading approach with full video segment processing, as shown in Figure 29. Considering the trade-off between accuracy and processing time, the configurations with just two stages (B and D) are clearly a better compromise, since they have virtually the same accuracy as the three stages cascades but are able to process the videos in almost half of the time required by the non-cascade approach with full video segment. When choosing a configuration between B and D, while their accuracies are similar, there is an important distinction between them. Configuration B has a better accuracy for non-SL videos, but performs relatively poor for the ASL class, arguably the biggest SL language present in video sharing sites. Considering this, Configuration D seems like the better choice as to how structure the cascading of classifiers.

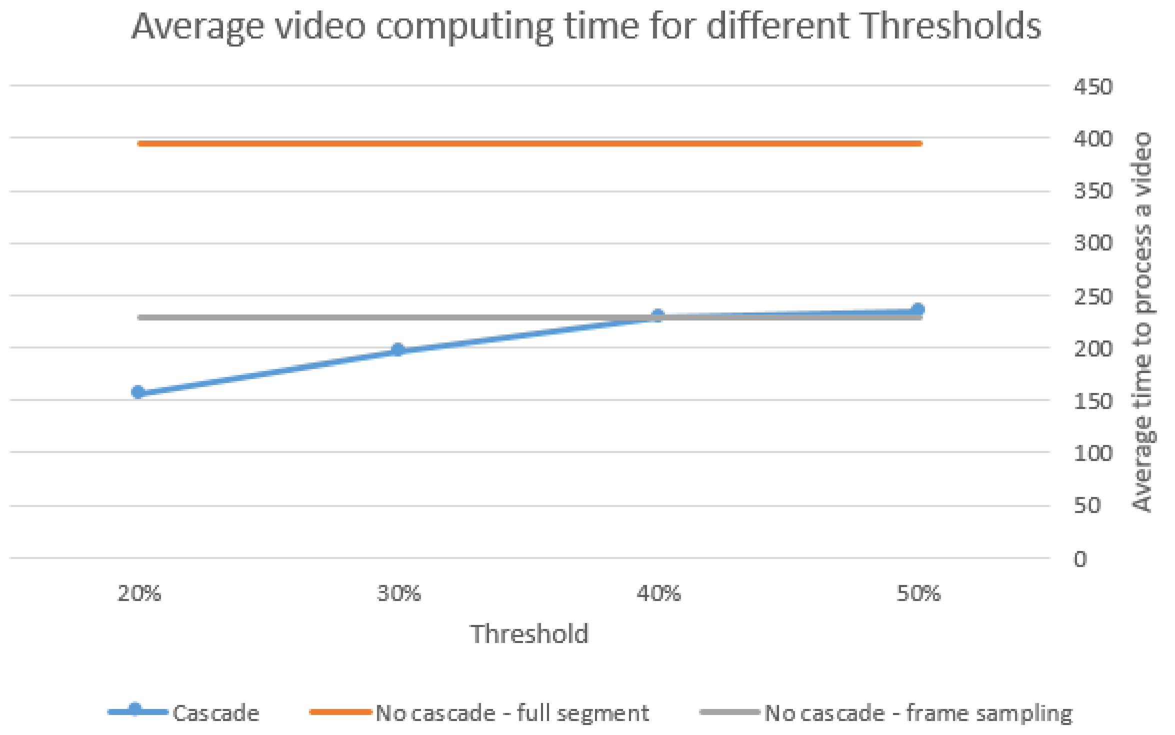


Figure 29. Average time spent to process a video with Configuration D in the multiclass problem for different confidence thresholds.

7.4 SLaDL instantiation

Considering the results obtained from exploring our three research questions, we have configured and implemented the first instantiation of SLaDL. In this setup, we are using a focused crawler to identify potential videos to be included in the collection. The crawler uses the same metadata features from the multimodal classifier and a SVM is trained for just detecting sign language, since our goal at the crawler stage is just to collect as many SL videos as possible, regardless of language, genre, or topic.

Videos with a high ranking score in the crawler frontier are going to be selected for classification. For the automatic classification we have explored two aspects of the system configuration: (1) how to combine visual and textual features; and (2) how to reduce the computational time to process the videos. For the former, we identified that late fusion using product rule between PMP visual features and NMF textual features as the best approach, and for the latter, we are using Configuration D of the cascading of classifiers. In this configuration, full video segment processing is disabled, and two sequential cascades are used, one for SL detection and another one for SL identification.

The results of all queries performed on the portal are included in the set of results as new seeds for the crawler. That way, the corpus of SL videos present in SLaDL will evolve and grow, with the queries performed acting as an indirect proxy for the interests of the community. All the models of the system can be retrained to include new videos that have been classified by the system, but until a proper user-feedback system is in place, manual validation is required to guarantee that no noise is present when training new models.

8. CONCLUSION AND FUTURE WORK

The work presented in this dissertation is the result of efforts investigating different problems related to sign language content accessibility. We started by tackling the problem of detecting sign language in pre-recorded videos, a problem simpler than SL transcribing but that has been mostly neglected by SL researchers. In our first approach, five hand-crafted features were computed and used to train a SVM classifier able to distinguish if a video had content in sign language or not. The initial SL detection approach was then generalized and improved by Karappa [5], this new version relaxed some of the constraints of our initial prototype and the classifier was now able to detect sign language in videos with more dynamic backgrounds and multiple signers. The increase in the classifier performance was coupled to a higher computation cost for processing new videos, and as soon as we began to idealize a digital library of SL content, we noticed that reducing computation costs were required. To that end, Duggina [43] proposed a series of optimizations for reducing the original PMP computational costs.

Like spoken languages, there are a multitude of sign languages, each one with their own structure, lexicon, and traits. For creating the sign language digital library (SLaDL), we had to be able to not only detect the presence of SL in videos, but also identify which sign language were being used in the video. A proof-of-concept study using a professionally curated dataset showed that the current set of PMP features were able to distinguish some pairs of sign languages, however, identifying sign language in videos posted on general purpose video sharing sites proved to be a more challenging task. Thus, we investigated how other feature modes, more specifically metadata, would be able to improve that.

Metadata features are much faster to compute than visual ones, therefore a combination of textual and visual features allowed us to not only improve SL detection and identification classifier accuracy, but also to explore ways to reduce computing time even further. For this, we have proposed the cascading of multimodal classifiers. These cascades employed cheaper visual feature techniques at earlier stages and tried to classify a video with as little computational effort as possible.

To materialize our initial version of the SLaDL, we also needed a way to establish and steadily grow a collection of potential sign language videos. Thus, we have explored different crawling strategies to find one suited to identify potential SL videos that have not been added to the SLaDL corpus yet. With the results of all of our experiments, we proposed the configuration of an initial version of SLaDL. In this version a focused crawler would collect and rank videos posted online for later processing, while a sequence of two cascades would classify new videos obtained from the crawler. Where the first cascade would focus on the detection of SL task, and the second would perform SL identification. Figure 30 depicts the problems and solutions explored by this project and their publication throughout its progress.

There are a variety of future work opportunities in this project. When considering the automatic classifier, we can investigate improvements on both the performance and average computational time of it. For the performance, currently PMPs are computed for every face located in a frame and then an average is computed. Such strategy might reduce the indication of a person signing if there are non-signers present in the video at the same time. This is common on picture-in-picture videos or recordings of events/presentations that include a SL interpreter. These videos are also less likely to include SL related metadata, making visual classification even more important.

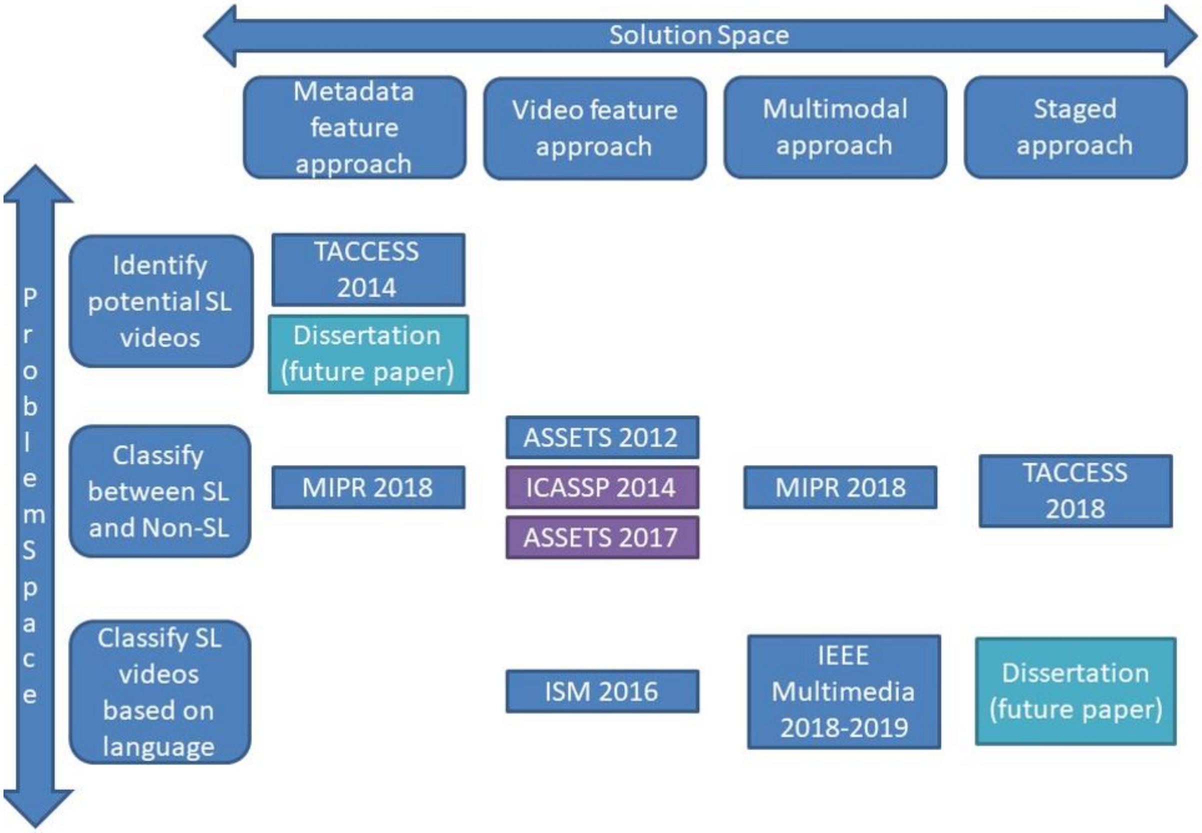

Solution Space
Metadata feature approach
Video feature approach
Multimodal approach
Staged approach
Problem Space
Identify potential SL videos
TACCESS 2014
Dissertation (future paper)
Classify between SL and Non-SL
MIPR 2018
ASSETS 2012
ICASSP 2014
ASSETS 2017
MIPR 2018
TACCESS 2018
Classify SL videos based on language
ISM 2016
IEEE Multimedia 2018-2019
Dissertation (future paper)

Figure 30. Overview of problems and solutions explored in this project, including external contributions (in purple).

Due to the difficulty of locating SL videos on video sharing sites and consequently building a testing corpus, our experiments have been limited to two specific sign languages (ASL and BSL), now that a crawler is available, a natural next step would be to crawl as much videos as possible in other languages to evaluate the staged classifier with other sign languages included.

For reducing the average computational time, there is still a lot of room for improvement. Our best performing cascade can, on average, classify new videos in about 4 minutes. There is reason to believe we can do it considerably faster. One approach is to modify the current frame sampling approach so that it samples frames not only for the face detection algorithm, but also for the background-foreground separation. This strategy could reduce the number of frames that need to have a PMP computed, thus significantly reducing the overall time to process a video.

Considering the bigger picture, a long-term goal is for the SLaDL portal to be a central place for easily locating sign language content in various languages and topics. While this first instantiation has the crawler and cascade of classifiers that allow it to be an ever-growing collection of SL resources, user feedback on the appropriateness of a video in terms of language is not available in the system. Such a mechanism would enable a higher quality of the digital library corpus, with the community being able to filter out non-SL videos misclassified by the classification system and annotate videos with additional metadata. A user study with members of the SL community (i.e. the targeted audience of our system) would be critical to understand the best way to implement the feedback mechanisms, as well as how to encourage users to provide such feedback. Given that for many users of our system a written language can be considered their second language, there are special HCI implications to be considered, so another user study should be made to validate and improve user interfaces and user experience aspects of the SLaDL portal, ensuring accessibility for a wide range of users.

REFERENCES

[1] J. Holt, S. Hotto, and K. Cole, "Demographic Aspects of Hearing Impairment: Questions and Answers," *research.gallaudet.edu*, 1994. [Online]. Available: https://research.gallaudet.edu/Demographics/factsheet.php. [Accessed: 07-Mar-2017].

[2] J. A. Holt, T. E. Allen, and C. B. Traxler, *Stanford 9: Interpreting the Scores: a User's Guide to the 9th Edition Stanford Achievement Test for Educators of Deaf and Hard-of-hearing Students*. Gallaudet University, Gallaudet Research Institute, 1997.

[3] C. C. Marshall, "No Bull , No Spin : A comparison of tags with other forms of user metadata," in *Proceedings of the 9th ACM/IEEE-CS joint conference on Digital libraries (JCDL '09)*, 2009, pp. 241–250.

[4] F. M. Shipman, R. Gutierrez-Osuna, and C. D. D. Monteiro, "Identifying Sign Language Videos in Video Sharing Sites," *ACM Trans. Access. Comput.*, vol. 5, no. 4, pp. 1–14, Mar. 2014.

[5] V. Karappa, "Detection of Sign-Language Content in Video through Polar Motion Profiles," *MS Thesis,* Texas A&M University, 2014.

[6] C. Valli, C. L. Washington, and D. C. Gallaudet, *Linguistics of American sign language: an introduction* 3rd edition, pp. 157–164, 2003.

[7] F. Shipman, R. Gutierrez-Osuna, T. Shipman, C. Monteiro, and V. Karappa, "Towards a Distributed Digital Library for Sign Language Content," in *Proceedings of the 15th ACM/IEEE-CE on Joint Conference on Digital Libraries*, 2015, pp. 187–190.

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

[9] C. D. D. Monteiro, C. M. Mathew, R. Gutierrez-Osuna, and F. Shipman, “Detecting and Identifying Sign Languages through Visual Features,” in *2016 IEEE International Symposium on Multimedia (ISM)*, 2016, pp. 287–290.

[10] F. M. Shipman, S. Duggina, C. D. D. Monteiro, and R. Gutierrez-Osuna, “Speed-Accuracy Tradeoffs for Detecting Sign Language Content in Video Sharing Sites,” in *Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility - ASSETS ’17*, 2017, pp. 185–189.

[11] T. Starner and A. Pentland, “Real-time American Sign Language recognition from video using hidden Markov models,” in *, International Symposium on Computer Vision, 1995. Proceedings*, Springer, 1995, pp. 265–270.

[12] G. Somers and R. N. Whyte, “Hand posture matching for irish sign language interpretation,” in *Proceedings of the 1st International Symposium on Information and Communication Technologies (ISICT ’03)*, 2003, pp. 439–444.

[13] D. Dimov, A. Marinov, and N. Zlateva, “CBIR approach to the recognition of a sign language alphabet,” in *Proceedings of the 2007 international conference on Computer systems and technologies*, 2007, p. 96.

[14] M. Potamias and V. Athitsos, “Nearest neighbor search methods for handshape recognition,” in *Proceedings of the 1st international conference on PErvasive Technologies Related to Assistive Environments*, 2008, p. 30.

[15] G. Caridakis, O. Diamanti, K. Karpouzis, and P. Maragos, “Automatic sign language recognition: vision based feature extraction and probabilistic recognition scheme from multiple cues,” in *Proceedings of the 1st ACM international conference on PErvasive Technologies Related to Assistive Environments - PETRA ’08*, 2008, p. 8 pages.

[16] C. Vogler and D. Metaxas, “Toward scalability in ASL recognition: Breaking down signs into phonemes,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1999, vol. 1739, pp. 211–224.

[17] J. L. Hernandez-Rebollar, “Gesture-driven American sign language phraselator,” in *Proceedings of the 7th international conference on Multimodal interfaces - ICMI ’05*, 2005, vol. 1, no. 202, p. 288.

[18] N. Cherniavsky, R. E. Ladner, E. A. Riskin, and others, “Activity detection in conversational sign language video for mobile telecommunication.,” in *FG*, 2008, pp. 1–6.

[19] B. G. Gebre, P. Wittenburg, and T. Heskes, “Automatic sign language identification,” in *2013 IEEE International Conference on Image Processing*, 2013, pp. 2626–2630.

[20] G. B. Gebre, O. Crasborn, P. Wittenburg, S. Drude, and T. Heskes, “Unsupervised Feature Learning for Visual Sign Language Identification,” in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2014, pp. 370–376.

[21] Ming-Kuei Hu, “Visual pattern recognition by moment invariants,” *IEEE Trans. Inf. Theory*, vol. 8, no. 2, pp. 179–187, Feb. 1962.

[22] E. Efthimiou, S. Fotinea, C. Vogler, T. Hanke, J. Glauert, R. Bowden, A. Braffort, C. Collet, P. Maragos, J. Segouat, “Sign language recognition, generation, and modelling: A research effort with applications in deaf communication,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2009, vol. 5614 LNCS, no. PART 1, pp. 21–30.

[23] M. Huenerfauth, E. Gale, B. Penly, M. Willard, and D. Hariharan, “Comparing Methods

of Displaying Language Feedback for Student Videos of American Sign Language," in *Proceedings of the 17th International ACM SIGACCESS Conference on Computers &#38; Accessibility*, 2015, pp. 139–146.

[24] K. A. Weaver, T. Starner, and H. Hamilton, "An evaluation of video intelligibility for novice american sign language learners on a mobile device," in *Proceedings of the 12th international ACM SIGACCESS conference on Computers and accessibility - ASSETS '10*, 2010, pp. 107–114.

[25] J. J. Tran, B. Flowers, E. A. Risken, R. J. Ladner, and J. O. Wobbrock, "Analyzing the intelligibility of real-time mobile sign language video transmitted below recommended standards," in *Proceedings of the 16th international ACM SIGACCESS conference on Computers & accessibility - ASSETS '14*, 2014, pp. 177–184.

[26] J. Gugenheimer, K. Plaumann, F. Schaub, P. Di Campli San Vito, S. Duck, M. Rabus, E. Rukzio, "The Impact of Assistive Technology on Communication Quality Between Deaf and Hearing Individuals," in *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing - CSCW '17*, 2017, pp. 669–682.

[27] X. Zhu, C. C. Loy, and S. Gong, "Learning from Multiple Sources for Video Summarisation," *Int. J. Comput. Vis.*, vol. 117, no. 3, pp. 247–268, May 2016.

[28] D. Cai, X. He, Z. Li, and J.-R. Wen, "Hierarchical Clustering of WWW Image Search Results Using Visual , Textual and Link Information," *Proc. 12th Annu. ACM Int. Conf. Multimed.*, pp. 952–959, 2004.

[29] W.-H. Lin and A. Hauptmann, "News video classification using SVM-based multimodal classifiers and combination strategies," *Proc. tenth ACM Int. Conf. Multimed. - Multimed. '02*, pp. 323–326, 2002.

[30] J. Yew, D. a. Shamma, and E. F. Churchill, “Knowing funny: genre perception and categorization in social video sharing,” *Proc. 2011 Annu. Conf. Hum. factors Comput. Syst.*, pp. 297–306, 2011.

[31] O. Madani, M. Georg, and D. Ross, “On using nearly-independent feature families for high precision and confidence,” in *Machine Learning*, 2013, vol. 92, no. 2–3, pp. 457–477.

[32] C. Snoek, M. Worring, and A. Smeulders, “Early versus late fusion in semantic video analysis,” *Proc. 13th Annu. ACM Int. Conf. Multimed. - Multimed. ’05*, pp. 399–402, 2005.

[33] Peng Wang, Rui Cai, and Shi-Qiang Yang, “A hybrid approach to news video classification with multi-modal features,” in *Fourth International Conference on Information, Communications and Signal Processing, 2003 and the Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint*, vol. 2, pp. 787–791.

[34] K. Filippova and K. B. Hall, “Improved video categorization from text metadata and user comments,” in *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information - SIGIR ’11*, 2011, pp. 835–842.

[35] S. Agarwal and A. Sureka, “A focused crawler for mining hate and extremism promoting videos on YouTube.,” in *Proceedings of the 25th ACM conference on Hypertext and social media - HT ’14*, 2014, pp. 294–296.

[36] A. Arasu, J. Cho, H. Garcia-Molina, A. Paepcke, and S. Raghavan, “Searching the Web,” *ACM Trans. Internet Technol.*, vol. 1, no. 1, pp. 2–43, Aug. 2001.

[37] J. Cho, H. Garcia-Molina, and L. Page, “Efficient crawling through URL ordering,” *Comput. Networks ISDN Syst.*, vol. 30, no. 1–7, pp. 161–172, Apr. 1998.

[38] M. Hersovici, M. Jacovi, Y. S. Maarek, D. Pelleg, M. Shtalhaim, and S. Ur, “The shark-search algorithm. An application: tailored Web site mapping,” *Comput. Networks ISDN Syst.*, vol. 30, no. 1–7, pp. 317–326, Apr. 1998.

[39] S. Chakrabarti, M. van den Berg, and B. Dom, “Focused crawling: a new approach to topic-specific Web resource discovery,” *Comput. Networks*, vol. 31, no. 11–16, pp. 1623–1640, May 1999.

[40] M. Diligenti, F. Coetzee, S. Lawrence, L. C. Giles, and M. Gori, “Focused Crawling using Context Graphs,” in *26th International Conference on Very Large Databases, VLDB 2000*, 2000, pp. 527–534.

[41] M. Khoo, “Community design of DLESE’s collections review policy: a technological frames analysis,” *Proc. 1st ACM/IEEE-CS Jt. Conf. Digit. Libr.*, pp. 157–164, 2001.

[42] E. Osmeloski, “2011 Yahoo! In Review: Top US Searches In 30 Categories.” [Online]. Available: http://searchengineland.com/2011-yahoo-in-review-top-us-searches-in-30-categories-103215. [Accessed: 15-Mar-2017].

[43] S. Duggina, “Evaluation of Alternative Face Detection Techniques and Video Segment Lengths on Sign Language Detection,” Texas A&M University, 2015.

[44] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, 2001, vol. 1, p. I--511.

[45] N. Cristianini and J. Shawe-Taylor, *An introduction to Support Vector Machines*. 2000.

[46] Z. Zivkovic, “Improved adaptive Gaussian mixture model for background subtraction,” in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, 2004, vol. 2, pp. 28–31.

[47] D. McKee and G. D. Kennedy, “A lexical comparison of signs from American, Australian and New Zealand sign languages.,” in *The Signs of Language Revisited: An Anthology To Honor Ursula Bellugi and Edward Klima*, 2000, pp. 49–76.

[48] J. Adcock, M. Cooper, L. Denoue, and L. A. Rowe, “TalkMiner : A Lecture Webcast Search Engine,” in *MM ’10 Proceedings of the 18th ACM international conference on Multimedia*, 2010, no. 21, pp. 241–250.

[49] E. A. Fox *et al.*, “Ensemble PDP-8,” in *Proceedings of the 10th annual joint conference on Digital libraries - JCDL ’10*, 2010, pp. 341–344.

[50] G. Salton and C. Buckley, “Term-weighting approaches in automatic text retrieval,” *Inf. Process. Manag.*, vol. 24, no. 5, pp. 513–523, Jan. 1988.

[51] M. Hoffman, F. R. Bach, and D. M. Blei, “Online learning for latent dirichlet allocation,” in *Advances in Neural Information Processing Systems*, 2010, pp. 856–864.

[52] C.-J. Lin, “Projected Gradient Methods for Nonnegative Matrix Factorization,” *Neural Comput.*, vol. 19, no. 10, pp. 2756–2779, Oct. 2007.

[53] H.-T. Lin, C.-J. Lin, and R. C. Weng, “A note on Platt’s probabilistic outputs for support vector machines,” *Mach. Learn.*, vol. 68, no. 3, pp. 267–276, 2007.

[54] J. R. Quinlan, “Improved use of continuous attributes in C4. 5,” *J. Artif. Intell. Res.*, vol. 4, pp. 77–90, 1996.

[55] Chih-Wei Hsu and Chih-Jen Lin, “A comparison of methods for multiclass support vector machines,” *IEEE Trans. Neural Networks*, vol. 13, no. 2, pp. 415–425, Mar. 2002.