University Of Exeter

College of Engineering, Mathematics and Physical Sciences

Video-based Sign Language Recognition

Literature Review and Project Specification

Jelena Kolomijec

November 19, 2019

Abstract

This paper is written in relation to a final-year BSc Computer Science project. The report is separated into two major parts of a literature review and a discussion of project specification with its requirements. Automatic Sign Language recognition system could benefit millions of hearing-impaired people to communicate with the world more efficiently. This project aims to review past research in the field and apply a new technique for solving this problem.

I certify that all material in this dissertation which is not my own work has been identified.

Signed:
Jelena Kolomijec

1 Introduction

Today, there is a big demand in automatic Sign Language (SL) recognition. This kind of interpreter could help hearing-impaired people communicate more efficiently and be understood better by those who do not know SL. At the moment, the only communication alternatives are written conversation or in-person human translator. This drove the motivation for the "Video-based Sign Language Recognition" project.

The aim of this project is to create a real-time British Sign Language (BSL) translator using a regular computer camera. The user should be able to sign in front of their device and the system should provide audio or text translation as the person is signing. However, there is a number of obstacles in Sign Language Translator (SLT) creation. First of all, there are computer vision limitations, which result in a number of false positives. Secondly, the difference in grammar between spoken and sign language is adding to the challenge. Finally, the computational complexity of the problem makes the translation too slow, defeating the purpose of the system. With the improvement of Computer Vision and Machine Learning (ML) technologies, the creation of SLT becomes more realistic. Nonetheless, there is a great deal of research and improvements to be done before the creation of such software is achievable.

This paper is split into two sections, providing a literature review and a project specification. § 2 will discuss published literature and technologies relevant to the project. § 3 provides the project specification, describing project requirements and evaluation processes, as well as technologies to be used in our SLT implementation. Lastly, these sections will be followed by a conclusion in § 4, highlighting the most significant elements of this paper.

2 Literature Review

Sign Language Recognition requires use of Computer Vision (CV) and Machine Learning (ML) algorithms, as well as Natural Language Processing (NLP) techniques. All these fields are rapidly evolving areas of computer science, with vast research and improvement done in recent years. With such a huge variety of literature, we will review the most relevant research done in the field and look into SL linguistics.

This section introduces main SL concepts and reviews different computer vision and machine learning techniques. We split this literature review into the following subsections. § 2.1 will expand on SL linguistics. § 2.2 lists translation components and stages. § 2.3 presents hand localization and tracking techniques. § 2.4 discusses relevant Pose Estimation frameworks. § 2.5 provides facial emotion recognition techniques. § 2.6 will expand on previously used sign classification methods.

2.1 Sign Language

Understanding the structure and nature of SL is crucial in order to build SLT. This is important in every stage of the translation, including feature extraction and sign classification. Therefore, here we will discuss the linguistics of SL.

Similar to spoken language, SL is not a universal language. Some countries such as Belgium, USA and India may even use multiple SLs in their country. Hundreds of SLs are in use today, making communication for hearing impaired people even more difficult. Every sign language consist of these three major components:

1. Finger-spelling, which is used for spelling words letter by letter. Finger-spelling signs can be found in Fig. 1.
2. Manual signs, which make up the sign vocabulary used for the majority of communication.

3. Non-manual features, which include body position and facial expressions. Example of non-manual features changing meaning of signs is demonstrated in Fig. 2.

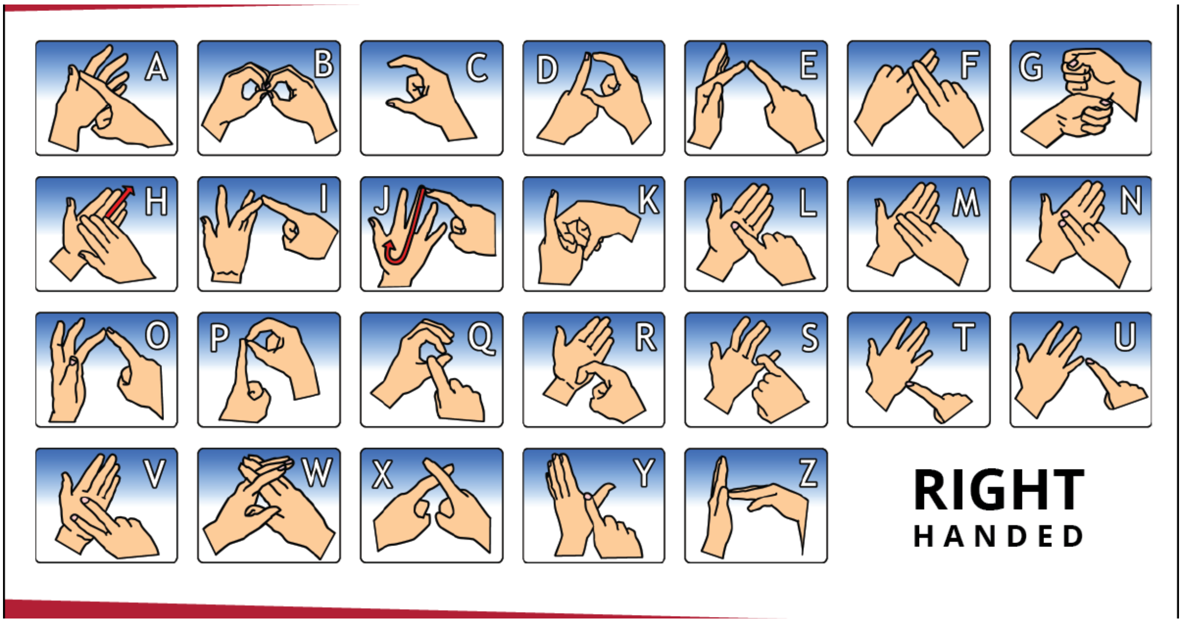

Figure 1: British Sign Language right-handed finger-spelling. Image retrieved from [5]

Most data is conveyable from hand motion and shape. Further information is taken from the body posture and facial features. As demonstrated in Fig. 2, facial expression and body language can change the meaning of a sign, just like intonation can change meanings in spoken language. Therefore, without observing non-manual features we cannot decipher the full meaning of the sentence. This means that SLT requires analysis on multiple layers. However, most research done in SLT is focused on classifying the lexical meaning of each sign.

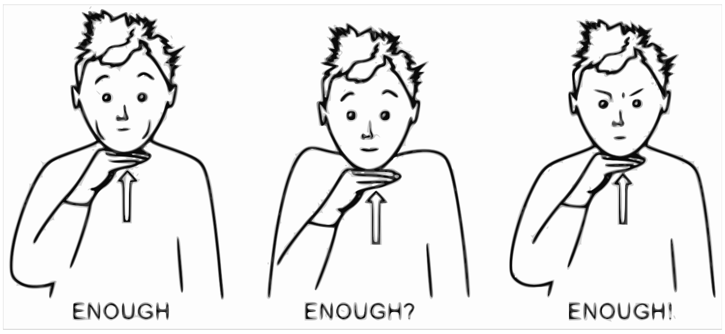

Figure 2: Example of non-manual features changing meaning of the sign. Image retrieved from [11]

From Fig. 2 we can decipher the lexical meaning of the sign ”enough”. However, without considering non-manual features in the second and third example we would not be able to tell if it was a question, or a command to stop an action. Therefore it is essential to consider both manual and non-manual features when translating SL.

2.1.1 Manual signs

According to sign linguists [31, 29, 22], the basic components of a sign gesture consist of handshape (finger configuration), hand orientation (direction in which the palm and fingers are pointing), location (where the hand is placed relative to the body), and movement (traces out a trajectory in space). The first phonological model, proposed by Stokoe [29], concentrated on the simultaneous organization of these components. Liddell and Johnson's Movement-Hold model [22], however, emphasized sequential movement of the components. The definition of movement segment is period during which change in state occurs - some part of the sign is conveyed in the execution. Hold segment is period when no such change occurs. More recent model [35, 31] aims to represent both the simultaneous and sequential organization of signs.

It is important to note, that in order to form a sentence, the hands need to move from the ending location of the current sign to the starting location of the next sign. Simultaneously, the hand-shape and hand orientation change in a similar manner. These changeover periods occur most frequently during continuous signing, therefore this is very important to keep in mind when processing SL.

2.1.2 Non-manual Features

In Fig. 2, three facial expressions are drawn, using the same manual sign. The 2.b figure shows raised eyebrows and widened eyes, which is an example of using upper line facial expressions, often accompanied with head and body movements [31]. This usually indicates emphasis on a sign or a sentence type such as question and negation. Indicators may involve eye gaze, direction, eyebrows and/or eye blinks. The eyebrows can be raised to indicate surprise or to ask a question. Eyebrow contraction can be used to express anger (see Fig. 2.c).

Facial expressions are used not only to show emotion or lexical distention, they are also important for grammar [31]. According to Sutton-Spence and Woll, "Head nods and head shakes have a range of forms and functions, and are important to the grammar and in conversation." Eye gaze also has lexical and grammatical functions.

2.2 Translation Components

Some of SL research has focused solely on the classification of finger-spelling alphabets and numbers [25]. The range for hand motion is very small in finger-spelling and consists of mainly finger configuration and orientation information (Fig. 1). This is very different from full signing, where whole hand movement and location as well as hand-shape and orientation are important. Therefore, interpreting full SL involves solving problems that are common to other research areas. This includes the tracking of the hands, face, body parts, the classification of these features, etc.

Most Machine Learning (ML) SLT approaches consist of these stages:

1. Feature selection: This stage refers to the attribute selection for ML model training. The process includes the selection of sign features that construct full needed information about the sign. Based on § 2.1, the required features are: hands, fingers, arm joints movement and location as well as facial expressions.

2. Pre-processing and feature extraction: ML algorithms require numerical data for learning and training. Feature extraction is done to transform arbitrary data, such as sign videos, to gather the relevant numerical data. Techniques relevant to our research will be covered in § 2.3, 2.4 and 2.5.

3. Signs Classification: Based on the interpretation of the features, the system performs data classification. It makes use of various algorithms such as Neural Networks and Random Forest Search. This is covered in § 2.6.

2.3 Hand Localization and Gesture Tracking

Most of the hand tracking techniques split into two categories: glove-based methods [30], and vision-based methods. Glove-based recognition systems generally require the user to wear a device connected to the computer using multiple cables. Since the goal of SLT is for the system to be portable and usable on personal devices such as smart phones or laptops, our solution must be able to recognise symbols by observing the data through a regular camera in different environments. Therefore glove-based methods would not be suitable for our project. However, vision-based approaches pose a number of challenges, as it needs to be background invariant, lighting insensitive, person and camera independent to achieve plausible performance. Nonetheless, in this project we will take the vision-based approach.

In order to extract needed information, the hand(s) must be located in the video. This is often implemented by using color, motion, and edge information. In skin-color based detection, it is often required to wear long-sleeved clothing, with restrictions on other skin-colored objects in the background [1, 15, 37]. Skin-color detection was combined with motion cues in Akyol and Alvarado [1], and combined with edge detection in Terrillon et al. [32]. In [32], a multi-layer perceptron neural network-based frontal face detector was used with the aim of face separation. Another approach was using colored gloves in combination with color difference [2, 3].

In all these and other vision-based approaches, it is particularly challenging to avoid occlusion. Some works avoid occlusion by excluding left hand and/or face from the image [32, 30]. Some use colored gloves, which makes face/hand overlap easier to solve. However, methods for dealing with occlusions with bare/uncovered hands have been found generally unsatisfactory. In [7], predictions of unmarked hand location were based on the assumption of small, continuous hand movement, achieving test recognition rate over 90% for 20 different gestures. However, it is dependant on stationary background and is only capable of detecting one hand at a time. In [15], where skin color-based approach with Kalman filter was taken, the systems ability of hand tracking would severely degrade in the instances of hands crossing/overlapping.

In the recent years, a few monocular RGB approaches have been proposed. In [38], plausible results have been achieved, however, their system only obtains relative 3D positions and does not work well with occlusions. A better monocular RGB solution has been proposed in [24], where the method obtains the absolute 3D position by kinematic model fitting. The second method has been proven to be more robust to occlusions and outperforms other state of the art hand tracking systems from monocular RGB.

This year, Google AI Blog released a new hand tracking framework with MediaPipe [4]. This approach achieves hand and finger tracking by employing machine learning (ML) to infer 21 3D key-points of a hand from a single frame. Currently, this is the only method capable of real-time performance on a mobile phone, and even scales to multiple hands, whereas all the other state-of-the-art approaches rely primarily on powerful desktop environments.

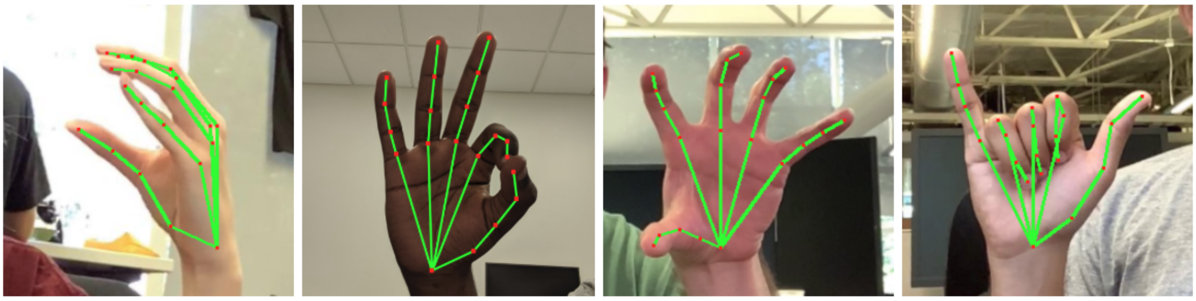

Figure 3: Example results of MediaPipe hand tracking. Image taken from [4].

Example results from MediaPipe hand tracking can be seen in Fig. 3. The pictures demonstrate ability of the framework to work well in different backgrounds, illumination levels and detecting different hand gestures.

2.4 Pose Estimation and Joint Localization

As identified in section 2.1, hand and finger gesture tracking is not enough to fully understand the meaning behind signs. Pose estimation and joint localization can also be very helpful in SLT. Relevant work on pose estimation was done in [6], where estimation of joint locations were found over frames of sign language videos. This was done by first performing background subtraction and then predicting joints location as a regression problem solved with random forest (this is demonstrated in Fig. 4). Another pose estimation technique usable for SLT is [26], there the authors use deep Convolutional Neural Networks (CNN) in order to regress over a heatmap of body joints. Here, temporal information from consecutive frames were used to improve performance.

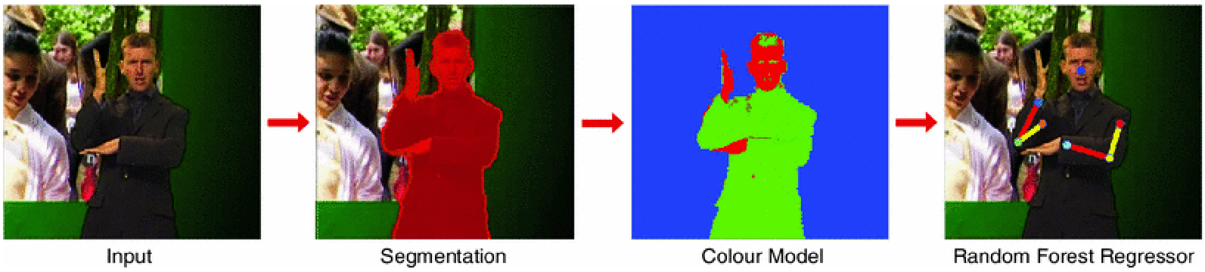

Figure 4: Example of [6]. Arm and hand joint positions are predicted by first segmenting the signer using a layered foreground/background model, and then feeding the segmentation together with a colour model into a random forest regressor.

2.5 Facial Expression Recognition

As previously demonstrated in Fig. 2, emotion recognition can make SLT more accurate. Currently, Facial Expression Recognition (FER) techniques are rapidly developing, however, research into integrating FER into SLT is insufficient [28]. Some state-of-the-art FER techniques include Support Vector Machine [13] and Deep Learning [36].

2.6 Signs Classification

There are two main approaches in sign gesture classification. The first consists of a single classification stage, where the sign is classified as a hole. The second consists of three stages, where the sign is split into simultaneous components, then each individual component is classified. Lastly, the components are integrated together for final sign-level classification. In the next subsection we introduce classification methods that can be used both for full-sign and individual component classification.

2.6.1 Classification Methods

Recently, Deep Learning has gained great popularity and achieved state-of-the-art performance in various fields such as Computer Vision and Speech Recognition. Previously, SLT researchers have mainly used intermediate representations [9, 18] for manual sign recognition, and the temporal changes in these features have been modelled using classical graph based approaches, such as Hidden Markov Models (HMMs) [2, 21]. With the growth of Deep Learning methods, SLT researchers have switched to Convolutional Neural Networks (CNNs) for manual [20, 20, 14] and non-manual [19] SL feature classifications, and Recurrent Neural Networks (RNNs) for temporal modelling [10, 8, 17].

Neural Networks

Neural networks is a ML technique, which consists of a network of learning units called neurons. These neurons learn how to convert input signals (e.g. picture of a sign) into corresponding output signals (e.g. the label “hello”). An illustration of CNN can be found in Fig. 5. CNN is separated into four main steps: convolution, subsampling, activation and full connectedness.

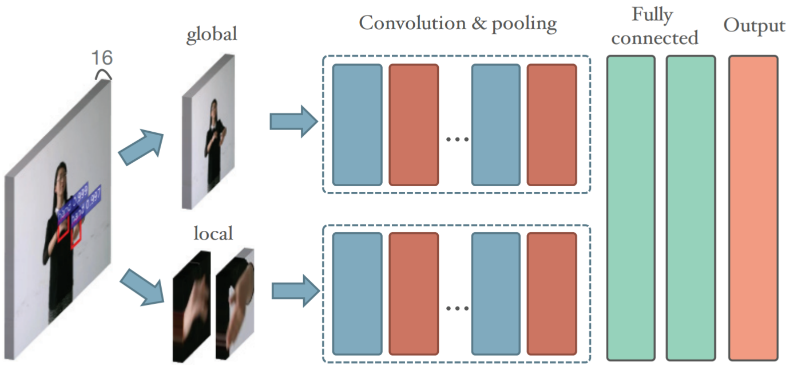

Figure 5: Two-stream 3-D CNN [14]. Blue and red blocks represent convolutional and pooling layers, respectively.

The convolution layer is the main building block of CNN. The network may consist of several Convolutional layers each of which can have a number of independent filters. Inputs from the convolution layer can then be “smoothened” in the subsampling process to reduce the sensitivity of the filters to noise and variations. Next, a polling layer reduces the amount of parameters and computation in the network. The activation layer is responsible for the flows of the signals from one layer to the other. Here, output signals with strongly associated past references would activate more neurons, enabling signals to be propagated more efficiently. The final layers are then connected to perform classification.

Before CNN became a popular classification technique, other Neural Networks and their variants have been used for this task. For example, Vamplew and Adams [34] used Multilayer perceptrons (MLP) to classify the hand-shape, hand location, orientation, and movement.

Using Neural Networks for video classification has not been as vastly explored as image classification. This is due to the problem being more complex as it has temporal, time-series data. In our case it’s movement trajectories and sign gestures, both of which consist of many data points and have variable temporal lengths. One approach to tackle this problem, is to treat space and time as equivalent dimensions of the input and perform convolutions in both time and space [16]. Another approach is fusing the features of two CNNs, one for the spatial and one for the temporal stream [14].

Hidden Markov Models

Several works have used Hidden Markov Models (HMMs) for sign classification. HMMs are able to process temporal data by discounting time variations through the use of skipped-states and same-state transitions [2, 21]. Because of this, HMMs are widely used in speech recognition systems. Continuous speech can be segmented into individual words which are chained together into a tree-structured network, which allows predicting most probable sequence of words. This idea has been transformed for recognition of continuous signs, using various techniques to increase computational efficiency. In [3], authors used language modeling, beam search and network pruning in order to increase efficiency.

Some researchers define sequential subunits, which are similar to phonetic acoustic models in speech, making every sign a concatentation of HMMs which model subunits. This reduces the training data and enables scaling to larger vocabularies. In [3], accuracy of 92.5% was achieved for 100 signs, signs using 150 HMMs. Encouragingly, recognition accuracy of of 81.0% was achieved for 50 new signs without retraining the subunit HMMs.

2.6.2 Integrating Component-Level Classifications

In component based approach, it is common to create a lexicon of the signs with hand-shapes, orientation, location, and movement components [25]. Identifying the full sign label from component-level results is then achieved by comparing the sign categories with the corresponding components. In [34], the lookup of signs from the lexicon was performed by the nearest-neighbor algorithm with a heuristic distance measure for matching sign word candidates.

An advantage of this approach is that new signs can be learned without retraining the component-level classifiers, as it would only require to add the sign description into the lexicon. This is due to a limited amount of components that can form a sign, which requires some categories to be reused to form more signs. In [21], Liang et. al have shown that using four gesture parameters (hand-shape, position, orientation, and motion) lexicon of 250 Taiwanese SL words can be identified using just 51 fundamental hand-shapes, 6 orientations, and 8 motion primitives.

2.6.3 Classification Challenges

According to [25] survey, most of the previous research, at least until 2005, dealt with isolated sign recognition where the user either performed the signs one at a time, starting and ending at a neutral position, or used an external switch between each word. To extend isolated recognition to continuous signing, automatic word boundaries detection algorithms need to be created. However, not enough research has been done in this field. According to the same survey, different signing styles of different individuals also poses a challenge for individual-invariant SLT creation. Similarly as in spoken language, signers might not always use 'proper' SL, creating shortcuts to the signs. This poses a challenge of configuring the system to different peoples needs.

3 Project Statement

This project aims to create a real-time British Sign Language (BSL) translator using a regular computer camera. The project will make use of free-source and community-made sign language videos to build a data set for a Machine Learning (ML) model. Open source frameworks and libraries will be used for computer vision and ML algorithms.

This section specifies the technical requirements and evaluation criteria of the project based on the literature review done in the previous section. The discussion of functional and non-functional requirements can be found in the first subsection. Here, motivation for technical choices will also be presented. The second subsection contains discussion of the final evaluation of the project, providing guidance for judgement of the final product.

3.1 Requirements

First of all, it is important to specify which sign language will be used in our project. Even though significant progress in the field has been done with American Sign Language, due to our current location, we decided to use British Sign Language for our research. Another motivation for our choice is potential support from the local deaf community.

3.1.1 Input/ Output

Next, we have to decide on the type of input. Here, we have a choice between glove-based or visual-based approach. Since glove-based method would require additional equipment, we decided to use the visual-based input. Our aim is real-time SLT creation, therefore, the final product will use real-time video stream from a personal computer camera as an input.

The output of the program should be a stream of translated signs, which can be displayed both as text and as audio. This should be displayed as soon as the sign is recognised, ideally in real-time. If time permits, a speech-to-text framework should be used, which will allow for reading spoken text as the speaker talks to the program.

Fig. 6 demonstrates the example input and output of the application. The design of the page will be similar, with the words written beside the signer.

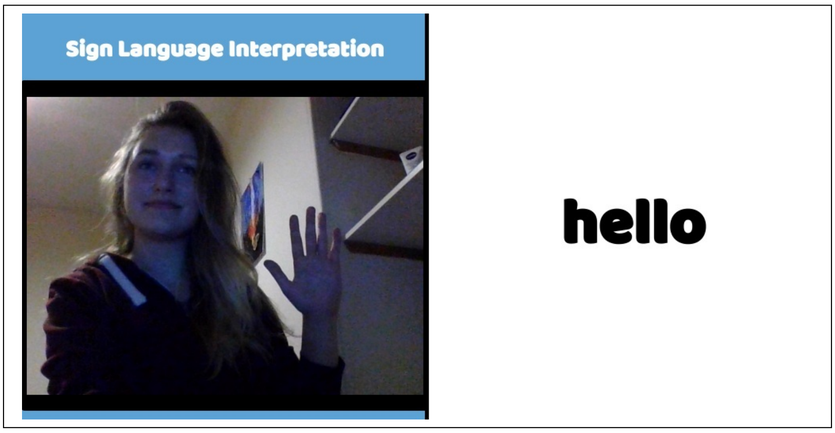

Figure 6: Example input/output of the program with the first design of the User Interface

3.1.2 Frameworks and Libraries

After establishing the expected input and output of the program, we will now present the technical requirements of the project. The program will implement some of the ideas discussed in the literature review in this paper. Presently, we were able to test MediaPipe hand tracking framework [4] on a personal device. The results of this framework were proven to perform very well, capable of tracking hands and fingers in real-time, avoiding occlusions and able to detect complex finger configurations. Therefore, our application should make use of the newly released On-Device, Real-Time Hand Tracking framework [4]. This technique has not yet been used in SLT, therefore this will be a novel approach to the problem.

As established in section 2, hand tracking is not enough for full sign recognition. Therefore a suitable pose estimation technique should be chosen. The method from [6] was able to achieve plausible results. In combination with MediaPipe [4], this should form a strong foundation for feature extraction. If time permits, a facial expression recognition technique should also be used in order to get the full meaning of the signs.

Next, a suitable classification method needs to be selected. The state-of-the-art CNN and RNN techniques have been proven to be successful in this field [17, 10]. Another motivation for using CNN is we were not able to find labeled BSL datasets, therefore, our datasets for ML model will have to be manually built and labeled. This means that the data will be very limited, and our classification technique should be able to work with such limited resources, which makes CNN the perfect candidate for the job. Some of the training videos will be taken from online BSL dictionaries [33, 27]. Manually taken videos might also be required in order to expand the dataset and make the system more accurate.

Due to insufficient video resources, the size of the lexicon for our SLT would have to adjust accordingly. Creating datasets manually is quite time consuming, therefore for now the vocabulary should be 30 different signs. If time permits, this should be increased.

3.1.3 Programming Languages

The language of choice for this project will be Python. This is due to numerous libraries existing for this language. Some of the frameworks of interest include TensorFlow, Torch and OpenCV. Another advantage of Python is it's simplicity of syntax, making it easier to create programs faster. Since Python is relatively slow language, any heavy calculation would be done in C++. If time permits, this program will be integrated into a webpage based approach to create a User Interface. In that case HTML, CSS and JavaScript will likely be used.

3.1.4 Other Requirements

The first performance requirement is about speed of translation. Since the aim of the project is making the communication more efficient and accessible, ideally, the program should run in real-time. It is difficult to judge whether or not this is achievable, as previous research either focused on translating very restricted vocabulary or would not aim to translate in real-time. Therefore, a trade-off would have to be made between the speed of the program, its accuracy and the size of known vocabulary.

The next requirement of the project is accuracy. It is apparent that the accuracy of each sign translation should be better than by chance. Based on previous research results, accuracy of at least 80% should be achievable.

The program should also not rely on any high quality cameras. Therefore, a regular computer camera should suffice. Since ML training will be very computationally expensive, a physical requirement for ML model training will be using high-performance computing. Since personal equipment does not allow to do so, free cloud computing with GPU services will be used. Some of the potential providers include Google Colab [12] or Microsoft Azure [23].

3.2 Evaluation criteria

The final product will be evaluated by comparing previous research results with the success rate of this application. The trained BSL signs should be interpreted with a better success rate than by chance (above 50%), ideally reaching success rate of above 80%. The size of BSL vocabulary should be no less than 30 signs, with the chance of expansion.

The accuracy of individual sign translation will be judged by performing trained BSL signs in front of the system and calculating its success rate to translate a known sign. In case some signs have very low accuracy rate, the reasons should be identified and well reasoned.

The speed of the program should be one of the main priorities. Translation of a sign should not be much slower than writing information down manually.

The program should also be able to run on a personal device without the use of external high-performance devices. The ML model training will make the use of free cloud GPU, as it will be very computationally expensive. The program itself, however, should run on a regular CPU machine.

4 Conclusion

Automatic Sign Language translation is a challenging, yet worthwhile task. With improvement of computer vision technologies, this task is becoming more achievable. The proposed approach will make use of Pose Estimation, Hand Tracking and Machine Learning tools in order to translate limited vocabulary British Sign Language. However, since such software is not yet fully built, it is vital to be aware of the risks and time constraints. Overall, this project has potential to be of high value and could help hearing-impaired people in their day to day lives.

The previous research done in the field was sufficiently discussed in section 2, creating a solid foundation for this project to proceed. The project was then specified in detail in section 3, explaining motivation behind technical and non-technical choices of the project. The criteria for project evaluation was also clearly stated, with focus on speed of the SL translation. Overall, this project has potential to use a novel approach to Sign Language recognition problem, helping future research in the field.

References

[1] S. Akyol and P. Alvarado. Finding relevant image content for mobile sign language recognition. In *IASTED International Conference-Signal Processing, Pattern Recognition and Applications (SPPRA), Rhodes*, pages 48–52, 2001.

[2] M. Assan and K. Grobel. Video-based sign language recognition using hidden Markov models. In *International Gesture Workshop*, pages 97–109. Springer, 1997.

[3] B. Bauer and K.-F. Kraiss. Video-based sign recognition using self-organizing subunits. In *Object recognition supported by user interaction for service robots*, volume 2, pages 434–437. IEEE, 2002.

[4] V. Bazarevsky and F. Zhang. On-Device, Real-Time Hand Tracking with MediaPipe. *Google AI Blog. Last Accessed: 18/11/2019. URL: https://ai.googleblog.com/2019/08/on-device-real-time-hand-tracking-with.html*. 2019.

[5] British Sign. Fingerspelling alphabet. *British-sign. 2019. https://www.british-sign.co.uk/fingerspelling-alphabet-charts/*. Last Accessed: 15/11/2019.

[6] J. Charles, T. Pfister, M. Everingham, and A. Zisserman. Automatic and efficient human pose estimation for sign language videos. *International Journal of Computer Vision*, 110(1):70–90, 2014.

[7] F.-S. Chen, C.-M. Fu, and C.-L. Huang. Hand gesture recognition using a real-time tracking method and hidden markov models. *Image and vision computing*, 21(8):745–758, 2003.

[8] N. Cihan Camgoz, S. Hadfield, O. Koller, H. Ney, and R. Bowden. Neural sign language translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7784–7793, 2018.

[9] H. Cooper, E.-J. Ong, N. Pugeault, and R. Bowden. Sign language recognition using sub-units. *Journal of Machine Learning Research*, 13(Jul):2205–2231, 2012.

[10] R. Cui, H. Liu, and C. Zhang. Recurrent convolutional neural networks for continuous sign language recognition by staged optimization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7361–7369, 2017.

[11] DeafBooks. Let'ssign for signs with facial expressions so crucial to language communication can add or change meaning [twitter post]. *2014. Retrieved from https://twitter.com/deafbooks/status/542646088269049857*. Last Accessed: 15/11/2019.

[12] Google. Google Colab main page. *2019. Google. colab.research.google.com*. Last Accessed: 18/11/2019.

[13] C.-C. Hsieh, M.-H. Hsih, M.-K. Jiang, Y.-M. Cheng, and E.-H. Liang. Effective semantic features for facial expressions recognition using svm. *Multimedia Tools and Applications*, 75(11):6663–6682, 2016.

[14] J. Huang, W. Zhou, Q. Zhang, H. Li, and W. Li. Video-based sign language recognition without temporal segmentation. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[15] K. Imagawa, S. Lu, and S. Igi. Color-based hands tracking system for sign language recognition. In *Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition*, pages 462–467. IEEE, 1998.

[16] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):221–231, 2012.

[17] S.-K. Ko, C. J. Kim, H. Jung, and C. Cho. Neural sign language translation based on human keypoint estimation. *Applied Sciences*, 9(13):2683, 2019.

[18] O. Koller, J. Forster, and H. Ney. Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers. *Computer Vision and Image Understanding*, 141:108–125, 2015.

[19] O. Koller, H. Ney, and R. Bowden. Deep learning of mouth shapes for sign language. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 85–91, 2015.

[20] O. Koller, H. Ney, and R. Bowden. Deep hand: How to train a CNN on 1 million hand images when your data is continuous and weakly labelled. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3793–3802, 2016.

[21] R.-H. Liang and M. Ouhyoung. A real-time continuous gesture recognition system for sign language. In *Proceedings third IEEE international conference on automatic face and gesture recognition*, pages 558–567. IEEE, 1998.

[22] S. K. Liddell and R. E. Johnson. American sign language: The phonological base. *Sign language studies*, 64(1):195–277, 1989.

[23] Microsoft. Microsoft Azure main page. *2019. Microsoft.* `azure.microsoft.com`. Last Accessed: 18/11/2019.

[24] F. Mueller, F. Bernard, O. Sotnychenko, D. Mehta, S. Sridhar, D. Casas, and C. Theobalt. Generated hands for real-time 3D hand tracking from monocular rgb. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 49–59, 2018.

[25] S. C. Ong and S. Ranganath. Automatic sign language analysis: A survey and the future beyond lexical meaning. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (6):873–891, 2005.

[26] T. Pfister, J. Charles, and A. Zisserman. Flowing convnets for human pose estimation in videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1913–1921, 2015.

[27] SignBSL. British sign language dictionary. *2019. SignBSL.* `www.signbsl.com`. Last Accessed: 16/11/2019.

[28] N. Song, H. Yang, and P. Wu. A gesture-to-emotional speech conversion by combining gesture recognition and facial expression recognition. In *2018 First Asian Conference on Affective Computing and Intelligent Interaction (ACII Asia)*, pages 1–6. IEEE, 2018.

[29] W. C. Stokoe Jr. Sign language structure: An outline of the visual communication systems of the American deaf. *Studies in Linguistics: Occasional Papers 8,*, 1960.

[30] D. J. Sturman and D. Zeltzer. A survey of glove-based input. *IEEE Computer graphics and Applications*, 14(1):30–39, 1994.

[31] R. Sutton-Spence and B. Woll. *The linguistics of British Sign Language: an introduction.* Cambridge University Press, 1999.

[32] J.-C. Terrillon, A. Piplr, Y. Niwa, and K. Yamamoto. Robust face detection and Japanese sign language hand posture recognition for human-computer interaction in an intelligent room. In *Proc. Int'l Conf. Vision Interface*, pages 369–376. Citeseer, 2002.

[33] UCL. Sign search section of the bsl signbank. *2019. UCL.* `bslsignbank.ucl.ac.uk/dictionary`. Last Accessed: 17/11/2019.

[34] P. Vamplew and A. Adams. Recognition of sign language gestures using neural networks. In *European Conference on Disabilities, Virtual Reality and Associated Technologies*, 1996.

[35] R. B. Wilbur. Syllables and segments: Hold the movement and move the holds! In *Current issues in ASL phonology*, pages 135–168. Elsevier, 1993.

[36] X. Zhao, X. Shi, and S. Zhang. Facial expression recognition via deep learning. *IETE technical review*, 32(5):347–355, 2015.

[37] J. Zieren, N. Unger, and S. Akyol. Hands tracking from frontal view for vision-based gesture recognition. In *Joint Pattern Recognition Symposium*, pages 531–539. Springer, 2002.

[38] C. Zimmermann and T. Brox. Learning to estimate 3D hand pose from single rgb images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4903–4911, 2017.