SignNet: End-to-End ASL to Text Translation

Adith Balamurugan
University of California, Berkeley
abala@berkeley.edu

Abstract

Sign language is an important means for convenient communication between deaf community and the majority population. This project describes a method for automated American Sign Language (ASL) to text conversion. There are a few existing methods which currently combat this problem using deep learning, such as the work done in State University of New York, Buffalo (Bheda and Radpour, 2017). However, nearly all of these methods focus on the recognition and translation of single letters and digits from the ASL to text. In this project, we choose to use deep learning architecture to take variable length raw video input and output a text translation (Vijayalakshmi and Aarthi, 2016). This translation will not be restricted to solely letters and digits, but will also generalize to complex words and phrases which can be found in our training corpus.

1 Introduction

The objective of this project is to put together an end to end pipeline to aid in ASL translation.

We take as input raw video footage of a speaker signing in ASL and output the text translation of the actions in the footage. The problem of ASL translation is very nuanced. Currently, the state of the art for fingerspelling recognition, where the speaker signs single letters and digits at a time to spell out words and sentences, can achieve up to 99.99% percent accuracy (Kang et al., 2015), which is incredibly accurate. However, many ASL speakers do not choose to communicate this way since the communication is very inefficient and slow. Instead, they typically choose to utilize the wide variety of signs and actions which each represent more complex ideas and concepts than just a letter in the alphabet. The problem we wish to solve is to generalize the fingersignalling recognition to the broader scope of ASL, including common signs for words and phrases as per the *Gallaudet Dictionary of American Sign Language*.

In American Sign Language, every word/phrase is defined by 5 key aspects of its sign. These are, in no particular order, **handshape** of dominant and non-dominant hand (if applicable), **movement** (direction of motion), **location of hand(s)**, **palm orientation**, and **non-manual markers** (e.g. **facial expression**, shoulder tilt). So in order to effectively translate ASL into English text, we must first accurately capture each of these components of the language. We address in detail how we deal with each of these components in section 4.

We actually will build off the work done by (Assael et al., 2016), which is a state of the art work done in the scope of machine lipreading. While this is not the same problem, they are very similar.

There are many aspects of the lipreading application which translate over to the ASL translation problem. For instance, we are attempting to handle raw video in both problems and must isolate a particular region

of interest per frame to interpret in both scenarios. The lips and their movement must be captured in the former and the hands and their position must be isolated in the latter. In order to develop good performance, we will harness the American Sign Language Lexicon Video Dataset (ASLLVD)

http://secrets.rutgers.edu/dai/queryPages/search/search.php

created and maintained by (Neidle and Vogler, 2012; Neidle et al., 2012). This dataset consists of videos of greater than 3300 ASL signs each performed by 1-6 ASL signers taken from different angles. For simplicity's sake, we make some reductions. In the interest of time and resources, we reduce the problem to single (dominant) hand signals, videos in the *front* viewpoint, and we will restrict the experiment to focus on correctly translating select phrases and sentences which all contain specific signals (nouns and verbs). We ultimately analyze the model's ability to disambiguate the meaning between videos with similar content in different orders and contexts.

Due to limitations in time and compute, we limit our objective to be able to accurately translate a smaller subset of phrases and sentences, each which are comprised of combinations of gestures. We limit the gestures to pertain to a finite selection of noun and verb choices which form coherent sentences or phrases. This allows for streamlined training and evaluation given our constraints. The method we introduce, however, can be generalized to the original objective by using training data containing a more vast ASL video vocabulary.

2 Related Works

Wide Residual Network (Kania and Markowska-Kaczmar, 2018) focuses on the fingerspelling problem, but many relevant suggestions can be taken from it.

Firstly, the preprocessing techniques they perform will be vital in the success of our own work, even more so than in theirs, because the videos we wish to translate will consist of full hand gestures rather than fingerspellings, so we will need to gather two specific pieces of information from each frame we analyze. First, we wish to localize to the hand. We will use a CNN on each frame in question to detect the hand object and choose the one with the greatest confidence as the hand. The coordinates of this hand in the frame will be recorded. Secondly, in order to gain localized pose information, we get the bounding box around the hand and pass the smaller image from cropping just the hand into a hand pose feature extraction model. This model will enable us to determine what position the hand is in. These two pieces of information together will allow us to make a prediction on the gesture the frame belongs to.

In this work the authors explore a Wide Residual Network approach in their CNN. Their results indicate that increasing the number of filters used reduced the number of operations performed and was far faster than the standard ResNet with comparable number of weights. Once the signals are translated, the text translation from the letters comes from being passed through a BiLSTM, which is the model we indent to use in our own work. One major thing which this work and all the following works share is that they harness transfer learning in order to detect the hands and extract meaningful features from the hands in each frame of the video. Training this accurately from scratch requires am immense amount of labeled data and compute time. In order to mitigate this problem, we borrow a frozen network from existing works which specialize in feature extraction from hands in the image.

Lipreading We draw inspiration from a state of the art machine lipreading work (Assael et al., 2016) and how they approach the problem having to handle variable length raw video input using Spatiotemporal convolutional neural networks (STCNNs), which can process video data by convolving across

time, as well as the spatial dimensions. In this work, the authors view the video frame by frame and attach a sound syllable to each frame based on the position of the mouth or ”-” if it is a space and the RNN is used to stitch the syllables together to generate the sequence of characters to form words. This is a similar approach that we will take, only we will have to classify entire gestures, which could come from variable number of frames and each gesture will correspond to a word or phrase, so stitching them together will be a slightly different problem since we will have to deal with more semantics of the entire sentence or phrase.

The LipNet’s performance was measured using a WER metric, which is defined as the minimum number of word (or character) insertions, substitutions, and deletions required to transform the prediction into the ground truth, divided by the number of words (or characters) in the ground truth. This will be a very useful metric for us to gather our results, since we will not be evaluating on phrases or sentences of a fixed length. Also, the phrases and sentences will have the property of containing words from a finite pool, so the WER is an appropriate metric to use in our evaluation (on a word level). More information on our evaluation process and specific can be found below in section 4.

Additionally, we draw from the work (Au and Heins), which also focuses on lipreading translation with one key addition. Mainly, the authors not only use the frames of the video themselves to determine the syllables corresponding to the lip position, but also provide additional information by looking at the delta difference between subsequent frames in the video. We believe this will be a valuable piece to the implementation for the ASL to speech translation as well.

In this work, the authors elect to use a pre-trained CNN known as ”VGG-Face” in order to extract facial features from the raw video frames (Parkhi et al., 2015). We opt into a similar design choice as we utilize a pretrained model which focuses on collecting hand pose features based on the hand object we detect in the frame of the raw video. The feature extraction process for each video is done separately, prior to training to reduce overhead, also because computing features takes a nontrivial time per video frame.

In (Petridis et al., 2017), the authors also utilize the delta image concept with their visual speech recognition method. The pipeline begins with a ROI cropping on the mouth in every frame. At every time step, these cropped sub-images and the delta image with the previous frame are passed through encoding layers before the final computed vector is passed in as input to a bidirectional LSTM. The improvement they observed when using the raw image and the difference image was very significant, so we choose to take advantage of this method as well.

Hand Tracking We observe in (Wirayuda et al., 2013) that they explore a separate method applicable to Sign Language translation by leveraging hand tracking systems. This utilizes geometric features from the palm of the hand by using modified Competitive Hand Valley Detection (CHVD) algorithm to get the reference point. Then the image of the palm of the hand is given a marker on the reference point from mapping result for identifying the shape descriptor based active fingers. Their work focused more on the recognition of the hand angle and conditions themselves since the focus of the work was more than just the sign language translation problem. They presented results on being able to correctly recognize the shape of the palm 74% of the time in optimal conditions, but that number dropped to 8-16% in extreme conditions. Extreme conditions include things such as motion blur and occlusion (object such as another hand in front of hand). These conditions can pose a problem to us in our translation problem. For this reason we choose to focus on single (dominant) hand signals.

In our work, we have a component that utilizes a similar aspect of hand tracking.

We use an object detection model to track the hand position in each frame of the image and track the position of the hand within the frame as we progress through the video. The relative positions of the hand along with the pose based features we get about the position of the hand help us determine what motion the hand is making and what gesture or word that frame belongs to. This information will assist in generating an appropriate prediction of the phrase that is being signed in the video.

FingerSpelling Classification In (Vijayalakshmi and Aarthi, 2016), the authors explore classic deep learning approaches using the CNN to classify the hand position from frame to frame within the video input. The recombination of the single frame classifications into the output text follow along the same suite as all the approaches above, using a BiLSTM model or a BiGRU as in (Assael et al., 2016). These are the two models for the text conversion we use following the frame-level translations.

3 Data

As mentioned in section 1, this paper takes advantage of the ASLLVD collected by (Neidle and Vogler, 2012; Neidle et al., 2012). The ASLLVD consists of videos of 3300+ ASL signs, each performed by 1-6 ASL signers, and each of those videos are taken from up to 4 different angles.

The breakdown of the statistics for the videos available in the dataset can be seen in Figure 1

To make it clear how this chart should be read, a total of 2,284 monomorphemic lexical signs were collected. For some signs, there is more than one variant, resulting in a total number of distinct sign variants that is greater: 2,793. For 621 of those sign variants, there are examples from a single signer; for 989 of them, there are examples from 2 signers, etc., and for 141 of those sign variants, there are examples from 6 native signers.

Class of signs	Number of signs	Number of sign variants	# sign variants with { 1, 2, 3, 4... } consultants		# tokens (examples) per sign { 1,2,...,6, >6 }		Number of sign tokens	
Monomorphemic lexical signs	2,284	2,793	x1	621	587	x1	8,585	Two-handed
			x2	989	858	x2		5,713
			x3	394	386	x3		67%
			x4	563	491	x4		
			x5	85	142	x5		One -handed
			x6	141	154	x6		2,873
					175	>6		33%
Compound signs	289	329	x1	129	117	x1	749	
			x2	106	107	x2		
			x3	48	46	x3		
			x4	33	33	x4		
			x5	4	11	x5		
			x6	9	13	x6		
					2	>6		
Number signs	76	88					260	
Loan signs	46	52					136	
Classifier constructions	27	31					38	
Fingerspelled signs	21	21					25	
ALL	2,742	3,314	--	--	--	--	9,794	

Figure 1: Overview of the statistics from the dataset

Many of the videos in the dataset are single gestures repeated by the same signer two to three times. These single-sign videos were annotated with gloss tables. An example can be found in Figure 2

B	C	D	E	F	G	H	I	J
ACCIDENT	Liz	ACCIDENT	ACCIDENT	S	S	S	S	N
	Naomi	ACCIDENT	ACCIDENT	S	S	S	S	N
	Liz	ACCIDENT	(5)ACCIDENT	S	S	A	A	N
	Naomi	ACCIDENT	(5)ACCIDENT	S	S	10	10	N
	Lana	ACCIDENT	(5)ACCIDENT	S	S	S	S	N
	Dana	ACCIDENT	(5)ACCIDENT	S	S	S	S	N
	Tyler	ACCIDENT	(3)ACCIDENT	3	3	S	A	N

Figure 2: Example annotation for ”accident” gesture videos

To make sense of the annotation in Figure 2, observe that the glosses have been assigned only so as to assure a unique gloss for each sign variant. Lexical variants of a sign have been grouped together, i.e., with the same gloss in Column D but a distinct glosses for each variant in Column E.

The use of the + sign indicates one repetition/reduplication beyond what would be the base form of a sign. Glosses containing different numbers of +s are considered equivalent for purposes of grouping (i.e., are not considered as distinct sign variants).

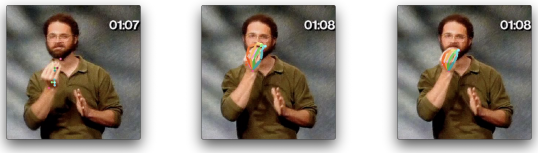
The dominant start handshape, non-dominant start handshape (if any), dominant end handshape, and non-dominant end handshape (if any) are listed in columns H and I.

These single-sign videos from our chosen vocabulary terms were used as training data for the translation model while the longer videos containing sequences of gestures strung together to tell a story were used for evaluation, since the goal of the project is to translate ASL to English, not to simply classify single gestures into their English gloss.

Vocabulary Our chosen vocabulary included finger spellings for single English letters and digits as we found that they were commonly used in the gesture videos we planned to use for evaluation. Additionally, we chose common pronouns such as *I/me, you, my, your*. Verbs such as *eat, work, go, come, study, live, work, want, take*. Other common gestures were *deaf, hearing* and nouns such as *children, home*. Our choice in vocabulary was strongly influenced by the data we had available to us, but also due to the problem we foresaw in occlusion where the hand or face could go behind another object in the frame and the information would be lost. Consequently, we chose our vocabulary such that the gestures were predominantly **one handed** or **two handed and symmetric** where obstruction of face or hand would be unlikely when viewing the video from the front view.

One additional important component to the model was developing a CNN which could vectorize the hands found in each frame of the input video sequence. Just as we capture the important features of the face in each frame of the video, the hand(s) encode valuable information regarding the meaning of the gesture. Specifically, we need to capture the handshape and the palm orientation of both the dominant and non-dominant hand (if applicable) in each frame of the video.

In order to effectively do this, we need training data which is in the form of an image of a hand and a label which identifies the handshape and the palm orientation of the hand in the image. To create this dataset, we went through the training videos before hand and used an existing model to detect the ROI for the hand(s) in each frame and crop the hands to get our training images and use the labels which were provided as part of the annotation for each gesture.



(a) Frame 0 (b) Frame 14 (c) Frame 22

Figure 3: Three frames from the gesture meaning "eat" with pose feature points highlighted

These images and labels are first used to train a model much like "VGG-Face" from (Parkhi et al., 2015) which can classify hands by shape and pose. Note, we did not use this model to be able to handle all ∼45 handshapes as our vocabulary was limited to words which only covered ∼10 of those handshapes. Additionally, our intention is not to use the classifier directly, but to remove the dense layers at the end of the architecture and use the CNN as a "featurizer" of the hands in each frame of the video. For ease of reference, we will refer this model as **HandNet** in the Method section where we detai the pipeline used in this work to solve this translation problem.

4 Method

Here we outline the pipeline we are using for this problem. The raw video input undergoes some normalization and centering prior to being passed into the model as this is the required input image for both the frozen models, VGG-Face and HandNet, that are being used on each frame. Both of these models were trained on preprocessed images, so we preprocess each frame of the video in the same way to get the best performance out of these two models. A sequence of T frames will be used as input. Each frame will first be passed into the VGG-Face network and the HandNet.

The main purpose of using the VGG-Face network is to capture the non-manual cue, facial expression, which is one of the key aspects of the ASL. The VGG-Face is being used as a featurizer of the image (frame), so

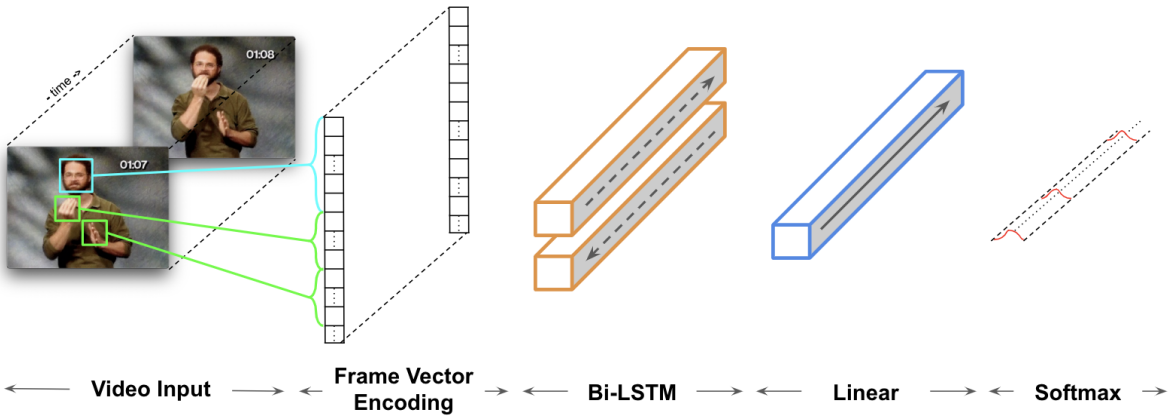


Figure 4: Model flow outline: 1) Raw Video Input fed into model, 2) Each frame is featurized into vector encoding 5 key elements of ASL. This includes featurization of facial expression, dominant and non-dominant hand (if applicable) along with x, y-position of hand(s) in frame. 3) Feature vectors passed as Input to BiLSTM. 4) Linear pass through outputs of LSTM to resolve erroneous/duplicate frames. 5) Softmax activation to classify best gesture for each frame (timestep). 6) Combine classifications to output single translation (video level)

the dense layers at the end of the network, which classified for facial recognition have been removed, so the output of this model will be a 4096-dimensional vector that encodes information about the face object which was detected in the image. If no face is detected in the image, the a zero vector is outputted.

The HandNet model is similarly used as a featurizer of each input frame. We are interested in the hands, because they contain the information corresponding to **handshape**, **palm orientation**, and **location**, which are 3 more important aspects of the language and we ensure that the information is not lost. The Handnet model attempts to find two hands in the frame which is passed in, and orders them in descending level of confidence. Should a (or both) hands not be found, the model will output the zero vector as the featurization for that hand. The hand with higher confidence is tied to the dominant hand and the one with lesser confidence is used as the non-dominant hand. Each hand gets output a 1024-dimensional vector which encodes information on its handshape and palm orientation. Additionally, the x, y-position of each of the detected hands in the frame is also outputed and appended to the end of the feature vector.

The outputs of the VGG-Face model (4096-dim vector) and the HandNet model (two 1024-dim vector *might be zero vectors, 2 pairs of coordinates *negative if hand not found) are all concatenated together to form a 6148-dimensional vector which corresponds to the provided frame.

Now we have T feature vectors, one for each timestep, which is computed for the frame. These features extracted up to this point will be processed by a BiLSTM model. The decision to use a BiLSTM model was due to the face that in the end we wish the translation to be a valid, syntactically correct sentence or sequence of words in the English language and that requires the ASL to be translated using context from both before and after the current timestep. However, the problem is not as simple as having to select the correct word a the given timestep based on the features. Each timestep contributes a frame which is nothing more than a moment in a larger gesture or sign which is the actual token we are interested in classifying or selecting. For this reason, we need the BiLSTM's power to look at frames from before and after the current one to determine which sign the current frame belongs to.

One more key reason as to why we chose

to use the BiLSTM model at this juncture was due the last aspect of the sign language that we had yet to effectively capture. The **movement of the hand(s)**. At each frame we are saving the x, y-position of each of the detected hands in the frame, however this alone does not capture the concept of movement. This alone does not describe that the hand is in a downward diagonal motion. Since the position is passed into the BiLSTM at each timestep as part of the feature vector, we hope that the model itself will learn that certain patterns in the hand positions from one frame to the next (and previous) indicate certain types of motion. We address one point of concern with this assumption in the Analysis section.

The output of the BiLSTM at each timestep is then passed through a Linear layer which is designed to classify each frame as belonging to a certain gesture from our selected vocabulary using a softmax activation function and an argmax sign selection. Then the sequence of classifications are resolved by removing duplicates and taking majority class over runs of gesture classifications to determine a single sequence of gestures.

The final step passes the English equivalent of these translations through a Language model to output an English sentence which is syntactically correct and best communicates the meaning of the gesture sequence we predicted.

The loss for the model is a categorical cross entropy over our chosen vocabulary on the gesture level translation. The loss is back propagated through the Linear resolver and the BiLSTM network, but is not used to update the VGG-Face model or the HandNet as these are both trained and frozen prior to the training done on video inputs.

5 Analysis

We choose to use the WER metric as a measure of performance. In the final text output, we will string together a sequence of words or phrases each tied to one of the gestures seen in the video. The language model will clean up the conjugations of words output by the model to make the sentence grammatically correct if possible.

The WER is defined as the **minimum** number of word/phrase insertions, substitutions, and deletions required to transform the prediction into the ground truth, divided by the number of words/phrases in the ground truth. The words/phrases in question are the ones in the vocabulary we are limiting our datasets to (i.e the words/phrases that can be expressed in a single gesture).

We can compute this WER for speakers that have appeared in the training data, but with samples not seen in the dataset. We can also compute the WER for unseen speakers performing hand signals from our limited vocabulary, but never seen in the training datset. This allows us to measure how well our model generalizes for the gesture recognition objective itself as well as how well the model generalizes for variations in signing introduced by different speakers previously unseen.

	WER	
	Unseen Speaker	**Seen Speaker**
SignNet	63.3%	55.6%

Table 1: SignNet performance on unseen speakers and speakers seen in training

Unfortunately, we do not have a very good baseline metric to compare our performance to since there has not been any work done in this area attempting to reconstruct English text from full ASL gestures. Our vocabulary did include finger spelling for single letters and digits, and since we used our HandNet to capture handshape and palm orientation information, we have metrics for performance on fingerspelling recognition. These can be compared to recent works and are even comparable considering this was not our main objective and was introduced to the vocabulary for convenience. These results can be found in Table 2.

6 Conclusion

This work is very promising, as it presents a very novel problem which has predom-

Method	# of classes	Input	Accur(%)
(Nagi et al., 2011)	6	Color	96
(Kuznetsova et al., 2013)	24	Depth	87
(Dong et al., 2015)	24	Depth	90
(Kang et al., 2015)	31	Depth	99.99
Ours	24	Color	83.4

Table 2: Fingerspelling performance of model compared to prior works

inantly remained unexplored. The performance leaves much room for improvement, and since there is no baseline value to compare to we can not be sure how well the proposed method actually performs compared to human error which would be the ideal target. We would need to perform a further study to gather that information by having ASL proficient participants translate the ASL gestures they observe from the videos seen in the data set and observe their rate of error. The error metric would also have to be WER in order to compare the performance to the results we gathered using the method outlined above and depicted in Figure 4. This would require that the participants would be evaluated on their ability to translate each gesture in the sequence independently as well as their ability to construct a coherent English translation of the sequence.

The performance we observe for unseen speakers versus seen speakers does provide reassurance that the model has learned to recognize certain video segments and the BiLSTM has the ability to provide reasonable classifications for each frame for previously seen videos as compared to brand new speakers. We definitely have many parameters which have been left un-tuned and could prove to be very crucial in improving the performance of our approach.

Additionally, we are pleased with the performance of the model on fingerspelling classification where the specific letters were simply left as additional classes in the vocabulary. While the performance is not state of the art by any means, we proved that even using raw video input, we were able to accurately classify the correct gesture for letters using the handshapes from only the RGB data from each frame in the video.

Undoubtedly, there is much room for improvement with this proposed method to resolve a text translation for an ASL input in the form of a video. However, we provide the groundwork in this paper to go beyond classifying fingerspelling in ASL for letter and digits, and actually categorize and translate more complex words and phrases in the ASL.

References

Yannis M. Assael, Brendan Shillingford, Shimon Whiteson, and Nando de Freitas. 2016. Lipnet: Sentence-level lipreading. *CoRR*, abs/1611.01599.

Andy Au and Adam Heins. Automated lip reading using delta feature preprocessing and lstms.

Vivek Bheda and Dianna Radpour. 2017. Using deep convolutional networks for gesture recognition in american sign language. *CoRR*, abs/1710.06836.

Cao Dong, M. C. Leu, and Z. Yin. 2015. American sign language alphabet recognition using microsoft kinect. In *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 44–52.

Byeongkeun Kang, Subarna Tripathi, and Truong Q. Nguyen. 2015. Real-time sign language fingerspelling recognition using convolutional neural networks from depth map. *CoRR*, abs/1509.03001.

Kacper Kania and Urszula Markowska-Kaczmar. 2018. American sign language fingerspelling recognition using wide residual networks. In *Artificial Intelligence and Soft Computing*, pages 97–107. Springer International Publishing.

A. Kuznetsova, L. Leal-Taix, and B. Rosenhahn. 2013. Real-time sign language recognition using a consumer depth camera. In *2013 IEEE International Conference on Computer Vision Workshops*, pages 83–90.

Jawad Nagi, Frederick Ducatelle, Gianni A Di Caro, Dan Cireşan, Ueli Meier, Alessandro Giusti, Farrukh Nagi, Jürgen Schmidhuber, and Luca Maria Gambardella. 2011. Max-pooling convolutional neural networks for vision-based hand gesture recognition. In *Signal and Image Processing Applications (ICSIPA), 2011 IEEE International Conference on*, pages 342–347. IEEE.

Carol Neidle, Ashwin Thangali, and Stan Sclaroff. 2012. Challenges in development of the american sign language lexicon video dataset (asllvd) corpus.

Carol Neidle and Christian Vogler. 2012. A new web interface to facilitate access to corpora: development of the asllrp data access interface. In *In Proceedings of the International Conference on Language Resources and Evaluation*.

Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, et al. 2015. Deep face recognition. In *BMVC*, volume 1, page 6.

Stavros Petridis, Zuwei Li, and Maja Pantic. 2017. End-to-end visual speech recognition with lstms. *CoRR*, abs/1701.05847.

P. Vijayalakshmi and M. Aarthi. 2016. Sign language to speech conversion. In *2016 International Conference on Recent Trends in Information Technology (ICRTIT)*, pages 1–6.

T. A. Budi Wirayuda, H. A. Adhi, D. H. Kuswanto, and R. N. Dayawati. 2013. Real-time hand-tracking on video image based on palm geometry. In *2013 International Conference of Information and Communication Technology (ICoICT)*, pages 241–246.