2018 17th IEEE International Conference on Machine Learning and Applications

Time Series Neural Networks For Real Time Sign Language Translation

Sujay S Kumar*, Tenzin Wangyal†, Varun Saboo‡ and Dr. Ramamoorthy Srinath§
Computer Science and Engineering Department, PES Institute of Technology
Bangalore, India
Email: {*sujay.skumar141295, †tenzin.wangyal1006, ‡v18saboo, §ramamoorthysrinath}@gmail.com

***Abstract*—Sign language is the primary mode of communication for the hearing and speech impaired and there is a need for systems to translate sign languages to spoken languages. Prior research has been focused on providing glove based solutions which are intrusive and expensive. We propose a sign language translation system based solely on visual cues and deep learning for accurate translation. Our system applies Computer Vision and Neural Machine Translation for American Sign Language (ASL) gloss recognition and translation respectively. In this paper, we show that an end to end neural network system is not only capable of recognition of individual ASL glosses but also translation of continuous sign language videos into complete English sentences, making it an effective and practical tool for sign language communication.**

***Index Terms*—Sign Language Translation, Computer Vision, Long Short Term Memory Networks, Neural Machine Translation, Attention Neural Networks, Deep Neural Networks, American Sign Language**

I. Introduction

Sign language is a visual language which involves the use of hands, body and face to communicate. Gestures may be made with a single hand or both hands, with the inclusion of facial expressions as well. Though there are some significant differences between spoken and sign languages, such as how they use space grammatically, sign languages exhibit the same linguistic properties and use the same language faculty as spoken languages.

Sign language recognition systems work on Gesture Parameter Set (GPS) which consists of five crucial parameters: hand shape, hand movement, hand and head orientation, hand and head location and facial expressions. In this paper, a sign refers to any gesture made using one or more parameters from the Gesture Parameter Set. A signer refers to the person making a sign in sign language and signing refers to the act of making a sign by a signer. A gloss refers to the written representation of a sign, which is the result of transcribing a sign in the text form.

There are two ways of signing in sign language -– Finger Spelling and Word Representation. Finger Spelling is a static technique of signing wherein the signer expresses himself with the use of alphabets of a spoken language. Each alphabet is represented in the form of a hand shape. In Word Representation method, the signer conveys meanings of words with the help of hand shape, hand movement and head movements. This technique involves continuous motion of hand and head and thus is more challenging to recognize.

As most of the communication revolves around word-level signing, there is a need for accurate and robust sign language translator systems which can recognize words by mapping to the parameters in the Gesture Parameter Set.

Considerable research has been done in the past twenty years on identifying hand gestures from the movement of human hands, the challenge being far greater in tracking techniques for non-rigid objects like bare human hands compared to tracking techniques for rigid objects. Hence, modern sign language translators require signers to wear gloves fitted with multiple sensors and electronic components in order to aid recognition of words. While the gloves are a state of the art technique for sign language translation [11], they are expensive and intrusive, with a degree of undesirable conspicuity.

To overcome the practical issues faced by gloves, we propose a system that uses a video capture module as input for translation. Our system is an ensemble of multiple neural networks, where each network is trained to optimize over separate stages of translation. This involves identification of ASL glosses as the first stage with the translation of ASL glosses to English sentences being the second stage. In the first stage, we implement a novel neural network architecture that is capable of identifying ASL glosses from a streaming video input which consists of signers enacting ASL equivalent of complete English sentences. In the second stage, we implement another neural network architecture to translate ASL glosses identified in the previous stage to English sentences, which is typically used in language translations.

II. Past Work

The research in the sign language domain, including translation into text/speech can be broadly split into two approaches - hardware based and visual cues based.

Gaikwad, P.B et al. [1] propose a hardware solution using flex sensors and accelerometers inside gloves to identify the signs. Each sign is uniquely identified by the combination of degree of bends and speed of movements of the hand as demonstrated by [15]. While statistically accurate, there are inherent constraints of such a device such as cost, maintenance and usability.

All authors contributed equally.

978-1-5386-6805-4/18/$31.00 ©2018 IEEE
DOI 10.1109/ICMLA.2018.00043

One of the earliest work was done by Starner, T.E. et al. [5], in which the authors propose a visual solution. Using a Hidden Markov Model, the system can recognize a subset of American Sign Language vocabulary by using a single camera module. The approach does not work on bare hands and requires the signee to wear specific colored gloves which allow the hands to be tracked by the camera. However, the drawback of requiring gloves hinders its regular use.

To tackle the challenging problem of tracking hands without gloves, Kishore, P.V.V. et al. [7] propose a solution employing active contour models [19]. This energy minimizing algorithm is used to track the hands and head of a signer in real time video. A simple feed forward neural network is employed for the classification of signs using error back propagation algorithm. However, the research only focuses on identifying single signs and does not attempt to identify continuous signs in a streaming video.

Koller, O. et al. [4] propose a continuous translation system on the German Sign Language database. They use a statistical approach to recognize the continuous sign language from different signers with a large vocabulary. HOG-3D features [20] are used to capture the edges of the hand spatially. While they achieved state-of-the-art results, the choice of using statistical features limits the potential of using similar features because of the vast differences between region specific sign languages.

Forster, J. et al. [3], [14] put forth the idea of the similarity between the automatic speech recognition system [13] and the sign language translation. They propose that the approaches employed in automatic speech recognition can be combined with computer vision in order to tackle automatic sign language recognition.

Nayak, S. et al. [6] present a framework which employs a probabilistic model to learn the recurring signs in multiple sign language videos which contain the vocabulary of interest, automatically. Each video can be split on commonly recurring patterns which are usually the stopwords, to obtain signs which can then be identified by a single sign detection as outlined in [7].

A different approach to identifying sign languages is through finger-spelling, a technique in which the signer signs a word by using individual signs for each letter. A popular technique is to store an existing copy of the static signs and compare the similarity between the images [2] and also its eigenvectors [9]. Slight variations in the hand positioning and viewing angles can be addressed by using a contour descriptor as described in [8]. Finger-spelling identification systems can be used for short conversations and simple gesture based inputs such as kiosks and personal assistant devices [2]. However, it is impractical for daily communication.

Sign language in practice involves heterogeneous signing, which consists of both word level signing and finger spelling, where the latter is used for complex words without corresponding signs.

III. Architecture And Methodology

The overall architecture of the system is illustrated in Fig. 1

A. Stage 1 - Isolated Gloss Recognition System

In this stage, the ASL glosses are identified from a stream of continuous sign language video. The isolated gloss recognition system consists of video pre-processing and a time series neural network module to accomplish its objective.

B. Stage 2 - Gloss to Speech Neural Translator

In this stage, the set of ASL glosses identified in the previous stage is translated into semantically appropriate English sentences and an attention network is used to accomplish this objective.

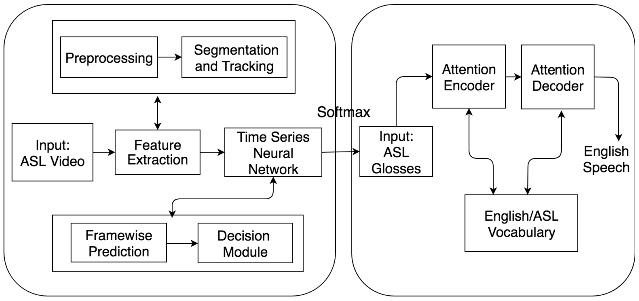


Figure 1. Sign Language Translation System Architecture

IV. Data Source

The videos used to train Isolated Gloss Recognition System and the dataset used to train Gloss to Speech Neural Translator were obtained from The National Center for Sign Language and Gesture Resources (NCSLGR) Corpus [16], [17] and [18]. Only raw videos with frontal view of the signer were used for training and testing.

V. Stage 1 : Isolated Gloss Recognition System

A. Video Preprocessing

The input to the Isolated Gloss Recognition system is a streaming video that consists of a signer performing gestures in front of the camera. Background noise cancellation is beyond the scope of this research and hence the video consists of a person wearing dark clothes with a dark background with minimal noise. Each video is processed frame wise, treating each image as input to the neural network at the current time step.

The first preprocessing step involves segmentation and extraction of hands and face from the frame. Hands provide the bulk of information required to identify the gloss represented by the signer, while facial expressions, combined with head movement, provide nuances to the expressed gloss. This step is accomplished by leveraging Gaussian Blur filter and Otsu's Binarization. Gaussian Blur achieves smoothening of the image (frame) by applying a Gaussian function transformation on each pixel in the image.

In the next step, Otsu's Binarization is used to extract the region of interest i.e hands and face of the signer. Since the video consists of primarily two tones of color, binarization technique can be reliably used to extract the region where the skin tone is present. Otsu's Binarization technique performs clustering based image thresholding and reduces the gray scale frame into a binary frame. It constructs a bi-modal histogram (foreground and background pixels) and calculates the optimum threshold separating them so that their intra-class variance is minimal (because the sum of pairwise squared distances is constant), as seen in (1). The result of this step produces a binary matrix where each position is marked by a boolean value indicating presence or absence of skin.

$$\sigma_w^2(t) = w_0(t)\sigma_0^2(t) + w_1(t)\sigma_1^2(t) \tag{1}$$

where weights w_0 and w_1 are the probabilities of the two classes separated by a threshold t, and σ_0^2 and σ_1^2 are variances of these two classes.

The next step involves identifying the points of interest from this binary image using Active Contours. Contour models define the boundaries of shapes in an image. Active contour model is an algorithm to delineate an object outline from a possibly noisy 2D image.

B. Feature Extraction

After pre-processing each frame, we end up with the coordinates C which identify the edges of the hands and face, effectively providing us with the shape, size and location as a set of (x, y) coordinates as seen in (2).

$$C = \{(x_i, y_i)\forall i\} \tag{2}$$

These values cannot be directly fed as the feature vector to the neural network due to the spatial uncertainty associated with the absolute coordinate space. These x and y coordinates are with reference to the bottom left corner of the frame. Hence, these coordinates are heavily dependant on the position of the signer in the video. In order to make our isolated gloss recognition system positionally agnostic, it was imperative that another approach had to be taken which preserves the spatial information even if the signer moves around in the frame. A novel angular-distance mapping technique is used to accomplish the same. This technique, known as Angular Hashing, is performed from the calculated $Point\ of\ Reference\ (PoR)$. The PoR considered is the centroid of the region of interest. For each coordinate (x_i, y_i) in the contour, relative angle $\theta_{i,PoR}$ and the distance $D_{i,PoR}$ from the PoR is computed, which provides an indication of the relative movement of the contour during the sign language gesture. Using the two calculated attributes, $X_t \in R^{360}$ is constructed as seen in (3).

$$X_t[\theta_{i,PoR} * 180/\pi] = D_{i,PoR} \quad \forall \quad (x_i, y_i) \in C \tag{3}$$

where $\theta_{i,PoR}$ is the angle between the point (x_i, y_i) and PoR, $D_{i,PoR}$ is the distance between the point (x_i, y_i) and PoR and X_t is the feature vector at time step t.

When there are multiple coordinates with angles between them less than 1°, collision occurs in the feature vector hashing algorithm as illustrated in the Fig. 2. In such cases, the distances are simply summed at their respective hashed keys. Thus, the feature vector for each frame, would be a 360 index array containing distances of pixels from the (PoR).

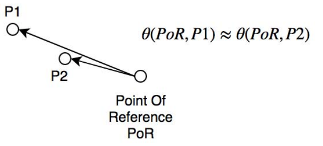


Figure 2. Angular Hashing - Collision

C. Univariate Time Series Classification

The problem of identifying the specific gloss after observing a video can be modeled as a classification problem which uses a time series Recurrent Neural Network (RNN) that processes a single frame at each time step.

A recurrent neural network (RNN) is a class of artificial neural network where connections between units form a directed cycle. This creates an internal state of the network which allows it to exhibit dynamic temporal behavior. RNNs leverage sequential information where the output not only depends on the current element of the sequence, but on the previous elements of the sequence as well.

LSTM (Long Short-Term Memory) is a type of recurrent neural network (RNN) architecture but unlike traditional RNNs, an LSTM network is well-suited to learn from experience to classify, process and predict time series when there are very long time lags of unknown duration between important events. This is one of the main reasons why LSTM outperforms alternative RNNs, Hidden Markov Models and other sequence learning methods.

Recognition of individual glosses from a streaming video consisting of multiple glosses cannot be modeled by a simple RNN based neural network as they need a fixed number of time steps in order to make a prediction. In this case, since each gloss can take variable number of time steps and since there was no indication of Beginning-of-Gloss or End-of-Gloss tags in the video, a variation of RNN is used consisting of another learnable parameter $\gamma \in R^T$

Our architecture consists of a many-to-many RNN composed of LSTM (Long Short-Term Memory) cells, where the feature vector from each frame is provided as the input at each time step t with a softmax classifier at the output layer over the entire ASL gloss vocabulary. At each time step γ parameter acts as a threshold that signifies whether the neural network can provide a classification for a gloss at that point or not. The layers used for the architecture are as follows:

1) Input Layer: Sigmoid neurons:

$$\sigma(\theta^T X) = \frac{exp(\theta^T X)}{1 + exp(\theta^T X)} \tag{4}$$

where, θ is the model parameter and X is the input vector

2) Hidden Layers: LSTM neurons:

$$f_t = \sigma_g(W_f X_t + U_f h_{t-1} + b_f) \tag{5}$$

$$i_t = \sigma_g(W_i X_t + U_i h_{t-1} + b_i) \tag{6}$$

$$o_t = \sigma_g(W_o X_t + U_o h_{t-1} + b_o) \tag{7}$$

$$c_t = (f_t \circ c_{t-1}) + (i_t \circ \sigma_c(W_c X_t + U_c h_{t-1} + b_c)) \tag{8}$$

$$h_t = o_t \circ \sigma_h(c_t) \tag{9}$$

where $c_0 = 0$ and $h_0 = 0$ and the $\circ$ operator denotes Hadamard Product.
$X_t \in R^D$: input vector to the LSTM unit
$f_t \in R^h$: forget gate's activation vector
$i_t \in R^h$: input gate's activation vector
$o_t \in R^h$: output gate's activation vector
$h_t \in R^h$: output vector of the LSTM unit
$c_t \in R^h$: cell state vector
$W \in R^{hXD}, U \in R^{hXh}$: Weight matrices
$B \in R^h$: Bias vector
σ_g: Sigmoid function
σ_c, σ_h: Hyperbolic tangent functions

3) Softmax Layer:

$$P(y_t^i | x_t^i) = \frac{exp(\theta^T x_t^i)}{\sum_{i=1}^{N} exp(\theta^T x_t^i)} \tag{10}$$

where $P(y_t^i | x_t^i)$ is the conditional probability of i^{th} gloss at time step t and x_t is the output vector of the last hidden layer at time step t.

4) Output Layer:

$$y_t = \begin{cases} argmax(P(y_t | X_t)), & \gamma_t \geq \eta_t \\ None \end{cases} \tag{11}$$

The LSTM architecture for this stage is illustrated in Fig. 3.

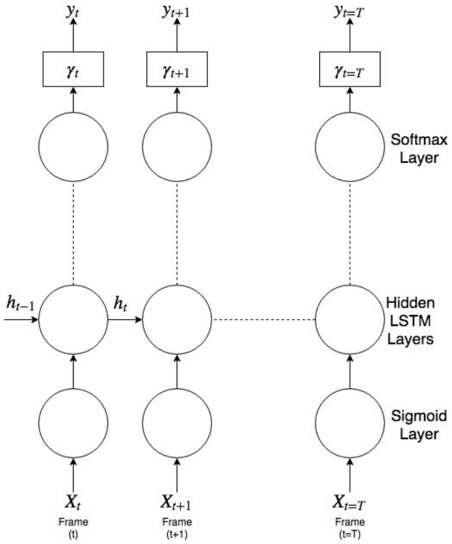


Figure 3. Time Series LSTM Network for Isolated Gloss Recognition

D. Isolated Gloss Recognition Results

The result of video pre-processing and feature extraction in real time is illustrated in Fig. 4, where the input frame is illustrated on the left hand side and the active contours extracted are overlayed on the right hand side.

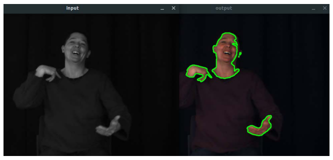


Figure 4. Hand and Face Segmentation with Active Contours

VI. Stage 2 : Gloss To Speech Neural Translator

A. Need for Translation

While ASL does follow a broad set of grammar rules during communication, these rules are not strictly followed, similar to other natural languages. Since ASL is just another natural language, a rule based translator cannot resolve ambiguities in natural language translation. Hence, we apply neural machine translation, which is the preferred approach for natural language translation.

B. Attention Based Sequence Model

Sequence to sequence models are built on top of language models by having an additional encoder step and a decoder step [10]. The encoder step involves a model that converts an input sequence into a single fixed vector and the decoder step is trained on both the output sequence as well as the single fixed vector from the encoder.

In order to prevent compressing all the encoder activations into a single fixed vector, in the attention mechanism, we provide a connection from the input sequences of the encoder to the decoder [12]. This mechanism stores all the hidden state activations from the encoder and gives the decoder a weighted average of the encoder hidden state activations at each time step of the decoder sequence as illustrated in Fig. 5. Implementing an attention based sequence model requires a slight modification to the existing encoder-decoder sequence model. We start by encoding the input sequence with an RNN

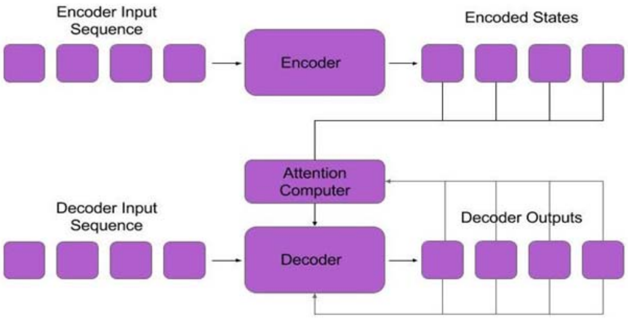


Figure 5. Attention Based Model

and storing the hidden activation at each encoding time step as seen in (12).

$$e = \{e_i \quad \forall \quad i \in T\} \tag{12}$$

The context (or attention) vector c_t is computed at each decoding time step. We compute a number, denoted by α_t, for each hidden state activation in the encoder sequence as seen in (14). Normalizing these numbers and applying a softmax provides us with the weights associated with each hidden state activation of the encoder sequence, denoted by α', as seen in (15). The context vector c_t is then calculated by computing the weighted sum over the hidden state activations $e_{t'}$ and the computed α' as seen in (16). The choice of the function f was selected to be a simple dot product of h_{t-1} and $e_{t'}$ as seen in (13).

$$f(h_{t-1}, e_{t'}) = h_{t-1}^T e_{t'} \in R \quad \forall \quad t' \tag{13}$$

$$\alpha_t = f(h_{t-1}, e_{t'}) \tag{14}$$

$$\alpha' = softmax(\alpha) \tag{15}$$

$$c_t = \sum_{i=0}^{n} \alpha'_{t'} e_{t'} \tag{16}$$

The context vector c_t is used to evaluate the hidden activation in the decoding phase at time step t as seen in (17). The function s_t is applied to this hidden state activation to convert it into a vector of dimension equal to the vocabulary size as seen in (18). The output of this step is used as an input to the softmax layer p_t that produces a probability distribution over the vocabulary as seen in (19). The input to the next decoder time step i_t is the predicted word in the current time step i.e $argmax(p_t)$ as seen in (20).

h_t, s_t, p_t, i_t are evaluated as follows:

$$h_t = LSTM(h_{t-1}, [w_{i_{t-1}}, c_t]) \tag{17}$$

$$s_t = g(h_t) \tag{18}$$

$$p_t = softmax(s_t) \tag{19}$$

$$i_t = argmax(p_t) \tag{20}$$

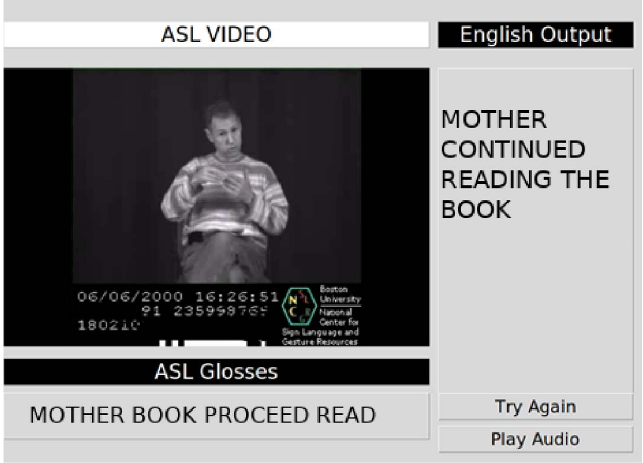
where, $h_t \in R^h$, $s_t := g(h_t) \in R^V$ and $p_t \in R^V$

Our decoder network is able to use different portions of the encoder sequence as context while it's processing the decoder sequence, instead of using a single fixed representation of the input sequence.

Hence, we model $P(Y_{t+1}|Y_0, ..., Y_n, x_0, ..., x_n)$ by evaluating $P(Y_{t+1}|Y_t, h_t, e)$

VII. Results And Discussion

The user interface of the complete system is illustrated in Fig. 6. The ASL glosses recognized can be seen in the bottom pane and the translated English sentence can be seen in the right pane in Fig. 6.



MOTHER CONTINUED READING THE BOOK

MOTHER BOOK PROCEED READ

Figure 6. User Interface

A. Isolated Gloss Recognition

In case of gloss recognition from sign language video stream, the performance is modeled on real time speech recognition systems and hence Gloss Error Rate (GER) is used. Along with GER, in order to provide a more meaningful metric due to the shortage of test data, Gloss Recognition Rate (GRR) was also measured.

$$GER = \frac{\text{No. of wrong glosses predicted}}{\text{Total no of glosses predicted}} \tag{21}$$

$$GRR = \frac{\text{No. of correct glosses predicted}}{\text{Total no of glosses in target set}} \tag{22}$$

The optimal trade-off obtained for recognition of individual glosses is 86% GRR with 23% GER. The results are presented in Tab. I. Here, γ is the threshold parameter at each time step as seen in (11).

Table I
Isolated Gloss Recognition Results

γ	***GER***	***GRR***
0.80	19%	51%
0.75	23%	62%
0.72	23%	86%
0.55	42%	91%

B. Gloss To Speech Neural Translation

BLEU [21], WER [22], PER [23] and perplexity were used as evaluation metrics. BLEU, WER and PER evaluate the overlap between the machine translation output and the reference output while perplexity evaluates the underlying language model. Perplexity (PP) measures the probability of a phrase or sentence being generated in a language. It is mathematically defined in (23) and can take values in the range $[1, \infty)$ with lower values indicating a better model. Perplexity is correlated to natural language tasks, but minimizing it does not guarantee improved results.

$$PP = 2^{-\frac{1}{N}\sum_{i=1}^{n}\log P(w_i)} \tag{23}$$

where N is the vocabulary size, n is the number of words in a sentence and w_i represents the i^{th} word. Our best system was able to achieve a perplexity measure of 1.51 for $N = 51$ ASL glosses. The results of machine translation metrics for different architectures are summarized in Tab. II.

Table II
Translation Results

Architecture	***BLEU***	***WER***	***PER***
1-layer-EncDec	41.7	57.5%	43.5%
2-layer-EncDec	44.3	50.7%	38.1%
1-layer-EncDec w/Attention	46.4	47.4%	36.4%
2-layer-EncDec w/Attention	49.7	43.7%	34.8%

VIII. Conclusion

In this paper, we have proposed a system to translate American Sign Language to English, based solely on visual cues and we have achieved reasonable success, as presented in Tab. I and Tab. II. To the best of our knowledge, this problem has not been tackled using deep learning methods in the literature as outlined in this paper. Deep learning methods are far better at handling such fuzzy decision making, like language translation than statistical models. This can be seen from the highly successful personal assistants like Siri and Cortana, which use neural networks to understand and translate human languages.

Hence, our system acts as a proof-of-concept that proves that neural networks can similarly be used for real time, non invasive and inconspicuous sign language translation.

IX. Future Work

We believe employing techniques like Convolutional Neural Networks instead of custom feature extraction algorithms will expand the capability of the Recurrent Neural Network by allowing the network to perform better feature extraction. The system can also be trained with other annotated sign languages, as and when the data corpus is available, without significant changes to the current system. Implementing sign language generation from English sentences will provide a comprehensive solution to sign language translation.

References

[1] Gaikwad, P.B. and Bairagi, D.V., 2014. Hand gesture recognition for dumb people using indian sign language. International Journal of Advanced Research in computer Science and Software Engineering, pp.193-194.

[2] Mandloi, D., 2006. Implementation of image processing approach to translation of ASL finger-spelling to digital text.

[3] Forster, J., Koller, O., Oberdörfer, C., Gweth, Y. and Ney, H., 2013. Improving continuous sign language recognition: Speech recognition techniques and system design. In Workshop on Speech and Language Processing for Assistive Technologies, Grenoble, France, August.

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

[5] Starner, T.E., 1995. Visual Recognition of American Sign Language Using Hidden Markov Models. Massachusetts Inst Of Tech Cambridge Dept Of Brain And Cognitive Sciences.

[6] Nayak, S., Duncan, K., Sarkar, S. and Loeding, B., 2012. Finding recurrent patterns from continuous sign language sentences for automated extraction of signs. Journal of Machine Learning Research, 13(Sep), pp.2589-2615.

[7] Kishore, P.V.V. and Kumar, P.R., 2012. Segment, track, extract, recognize and convert sign language videos to voice/text. (IJACSA) International Journal of Advanced Computer Science and Applications, 3.

[8] Sharma, R., Nemani, Y., Kumar, S., Kane, L. and Khanna, P., 2013, July. Recognition of single handed sign language gestures using contour tracing descriptor. In Proceedings of the World Congress on Engineering (Vol. 2, pp. 3-5).

[9] Dhobale, M.S.R. and Bhagat, V.B., A Survey On Gesture Recognition in Sign Language Recognition for Mute and Dumb People.

[10] Cho, K., Van Merriënboer, B., Bahdanau, D. and Bengio, Y., 2014. On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:1409.1259.

[11] Irving, M. (2018). Smart glove translates sign language gestures into text. [online] Newatlas.com. Available at: https://newatlas.com/sign-language-translate-glove/50474/.

[12] Luong, M.T., Pham, H. and Manning, C.D., 2015. Effective approaches to attention-based neural machine translation. arXiv preprint arXiv:1508.04025.

[13] Graves, A., Mohamed, A.R. and Hinton, G., 2013, May. Speech recognition with deep recurrent neural networks. In Acoustics, speech and signal processing (icassp), 2013 ieee international conference on (pp. 6645-6649). IEEE.

[14] Koller, O., Zargaran, O., Ney, H. and Bowden, R., 2016. Deep Sign: Hybrid CNN-HMM for Continuous Sign Language Recognition. In Proceedings of the British Machine Vision Conference 2016.

[15] Osborne, C. (2018). Student's smart glove translates sign language into speech — ZDNet. [online] ZDNet. Available at: https://www.zdnet.com/article/students-smart-glove-translates-sign-language-into-speech [Accessed 3 Mar. 2017].

[16] Neidle, C, Vogler, C, 2012, A New Web Interface to Facilitate Access to Corpora: Development of the ASLLRP Data Access Interface, Proceedings of the 5th Workshop on the Representation and Processing of Sign Languages: Interactions between Corpus and Lexicon, LREC 2012, Istanbul, Turkey.

[17] Bu.edu. (2018). ASL Linguistic Research Project at Boston University. [online] Available at: http://www.bu.edu/asllrp/ [Accessed 14 May 2017].

[18] Secrets.rutgers.edu. (2018). ASL Linguistic Research Project at Boston University - Database Query. [online] Available at: http://secrets.rutgers.edu/dai/queryPages/ [Accessed 14 May 2017].

[19] Kass, M., Witkin, A. and Terzopoulos, D., 1988. Snakes: Active contour models. International journal of computer vision, 1(4), pp.321-331.

[20] Dilsizian M., Yanovich P., Wang S., Neidle C., Metaxas D., 2014. A new framework for sign language recognition based on 3D handshape identification and linguistic modeling. In proceedings of 9th Language Resources and Evaluation Conference (LREC'14), pp. 1924-1929.

[21] Papineni K., Roukos S., Ward T., and Zhu W., 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 311-318.

[22] Su K., Wu M. and Chang J., 1992. A New Quantitative Quality Measure for Machine Translation Systems. In proceedings of the 14th conference on Computational linguistics, pp. 433-439.

[23] Tillmann C., Vogel S., Ney H., Zubiaga A., and Sawaf H., 1997. Accelerated DP Based Search For Statistical Translation. In proceedings of Eurospeech, pp. 2667-2670.