Sign Language Recognition using Deep Learning

Mohamed Mahyoub
Tutor Reach
United Kingdom
m.mahyoub@tutorreach.com

Friska Natalia
Faculty of Engineering and Informatics
Universitas Multimedia Nusantara
Tangerang, Indonesia
friska.natalia@umn.ac.id

Sud Sudirman
School of Computer Science and Mathematics
Liverpool John Moores University
Liverpool, United Kingdom
s.sudirman@ljmu.ac.uk

Jamila Mustafina
Naberezhnye Chelny Institute
Kazan Federal University
Kazan, Russia
d.n.mustafina@kpfu.ru

***Abstract*— Sign Language Recognition is a form of action recognition problem. The purpose of such a system is to automatically translate sign words from one language to another. While much work has been done in the SLR domain, it is a broad area of study and numerous areas still need research attention. The work that we present in this paper aims to investigate the suitability of deep learning approaches in recognizing and classifying words from video frames in different sign languages. We consider three sign languages, namely Indian Sign Language, American Sign Language, and Turkish Sign Language. Our methodology employs five different deep learning models with increasing complexities. They are a shallow four-layer Convolutional Neural Network, a basic VGG16 model, a VGG16 model with Attention Mechanism, a VGG16 model with Transformer Encoder and Gated Recurrent Units-based Decoder, and an Inflated 3D model with the same. We trained and tested the models to recognize and classify words from videos in three different sign language datasets. From our experiment, we found that the performance of the models relates quite closely to the model's complexity with the Inflated 3D model performing the best. Furthermore, we also found that all models find it more difficult to recognize words in the American Sign Language dataset than the others.**

Keywords—Sign Language, Sign Language Recognition, Deep Learning, Convolutional Neural Networks,

I. Introduction

Deaf people all over the world use sign language to communicate visually. The most common ways to sign words and sentences are by waving fingers, arms, hands, and making motions with your face. Each sign language has unique characteristics and has its own vocabulary and grammar. Sign language recognition (SLR) is a form of action recognition problem which uses technologies, such as computer vision, that can be used to translate a sign language into another sign language or a verbal language. SLR is a complex task, it considers several factors when identifying a sign word, including hand orientations, movement of hands, the posture of the body, and facial expressions. Even with cutting-edge models, tackling the problem of a sign with a larger vocabulary using a computer in real-world circumstances remains a difficulty.

While much work has been done in the SLR domain, it is a broad area of study. Since sign language uses extensive body, face, and hand movements it is an excellent domain for gesture classification problems - a perfect application for machine learning, a technique that is used to make decisions based on past data and experience. Machine Learning is a popular technique for solving a wide variety of problems including digital watermarking [1], non-destructive testing [2], and tourism data analytics [3]. Machine learning techniques have been proposed to develop an SLR system, with the Hidden Markov Model (HMM) as one of the most popular models [4]. The model has many variants including Multi-Stream HMM [5] and Tied-Mixture Density HMM [6] that have been used to recognize Japanese and Chinese sign languages, respectively. Some other machine learning approaches are also popular, these include Neural Networks, Naive Bayes Classifiers, Multilayer Perceptron, Self-Organizing Maps, Self-Organizing Feature Maps, Simple Recurrent Networks, Support Vector Machines, and 3D Convolutional Residual Networks. Additionally, there are also other less common approaches including using Eigenvalue Euclidean Distance and the Wavelets.

Deep learning techniques have recently outpaced prior cutting-edge machine learning methods in a variety of applications from medical image classification [7], measuring organ or landmark sizes from medical images [8], to pest detection and classification in agriculture [9]. Deep learning is a neural network-based machine learning technique that consists of a large number of processing layers to extract progressively higher-level features from data. Due to its large number of layers, deep learning models require a large amount of data to train. The development of extensive, top-notch, and publicly accessible labeled datasets such as ImageNet [10] together with the capabilities of parallel GPU processing are two significant elements that have greatly contributed to the development of deep learning.

Our research aims to assess the suitability of the deep learning approach in recognizing and classifying words from video frames in different sign languages. Our objectives are first to identify suitable deep learning models of varying complexities, to identify suitable datasets covering different sign languages, and lastly to provide analysis of the word classification accuracy that is measured from an experiment. The outcome of our research survey related to solving the problem of SLR is summarized in the next section.

II. Literature Review

The development of a sign language recognition system for sentence translation, or words into voice and text, is one of the fundamental problems in allowing communication between the deaf majority and hearing people. A system needs

to be developed which can enable real conversation between hearing people and the deaf. These systems should also account for the problem of splitting videos, including sentences or sign words into separate words. The problems that are prevalent in the domain of SLR are mainly two - isolated SLR and continuous SLR. Word-by-word recognition is an example of isolated SLR, whereas translating entire sentences is an example of continuous SLR [11].

The current methods for continuous SLR while using isolated SLR as building blocks also add layers of pre-processing and post-processing which indicate temporal segmentation and sentence synthesis, respectively. Most existing SLRs fall into the category of isolated SLR which deals with the recognition of words or expressions. Being more challenging, continuous SLR involves reconstructing sentence structures which divide the problem of recognizing sentences into three stages which are the segmentation of videos with time, recognizing isolated word/expression, and sentence synthesis with a language model.

The development of an end-to-end and sequence-to-sequence model that can generate video captions is a relevant research area. The state-of-the-art performance in generating image captions is demonstrated by Recurrent Neural Networks (RNN), specifically Long Short-Term Memory (LSTM) networks [12] [13]. After being trained using pairs of video-sentence, the model associates a sequence of video frames to a sequence of words to describe an event in the video clip. Attention mechanisms can also be incorporated into LSTM for the automatic selection of the most likely video frames [14].

Industry labs and research groups have created solutions that are open-source for SLR called Neural Machine Translation (NMT). It is a Neural Networks-based approach that translates phrases from a source language to a target language. The solutions are based on platforms related to deep learning. However, these tools are targeted only toward research groups that have a good understanding of deep learning architecture and know how to handle large code bases. One such solution, called Joey NMT [15] was designed as a model which provides a minimum code platform with quality that can be compared to more standard benchmarks with complex code bases. It includes standard network architectures such as RNN, transformers, different attention mechanisms, input feeding, configurable encoder/decoder bridge, standard learning techniques (dropout, learning rate schedule, weight tying, early stopping criteria), and visualization/monitoring tools.

The use of deep Convolutional Neural Networks (CNN) stacked with temporal fusion layers for extracting features, and bidirectional RNN for sequence learning modules was proposed with an iterative optimization process to exploit the representational capability of deep neural networks with limited data [16]. The end-to-end recognition model for the alignment proposal was trained first, and then the proposal was used to tune the feature extraction module. This process can run iteratively to achieve improvements in recognition performance.

For detecting hands from various sources of input, such as skeletons, images, flow features, videos, and so on, most deep learning-based models employed a CNN or a combination of a CNN with another approach [17]. Even though extensive research has been made in hand detection recently and several methods are suggested, there are still many problems to overcome. Even accurate key point annotations are difficult to make manually due to high occlusions in hand key points.

III. Material and Method

In this section, we will be discussing the datasets used and the approach we employ to achieve our aim and objectives. Three datasets of sign language will be used, they are INCLUDE: A Large Scale Dataset for Indian Sign Language Recognition [18], WLASL: Word-Level American Sign Language [19], and AUTSL: Ankara University Turkish Sign Language [20]. The summary of the datasets and some samples of video frames from each dataset are provided in Table I and Figure 1, respectively below.

TABLE I. Sign Language Datasets used

	INCLUDE	**WLASL**	**AUTSL**
Sign Language	Indian Sign Language (ISL)	American Sign Language (ASL)	Turkish Sign Language (TSL)
# Words	263	2,000	226
# Word Categories	15	-	-
# Videos	4,292	21,083	38,336
# Frames	270,000	-	-
Video Resolution	1920x1080	-	512x512
# Signers	7	119	43

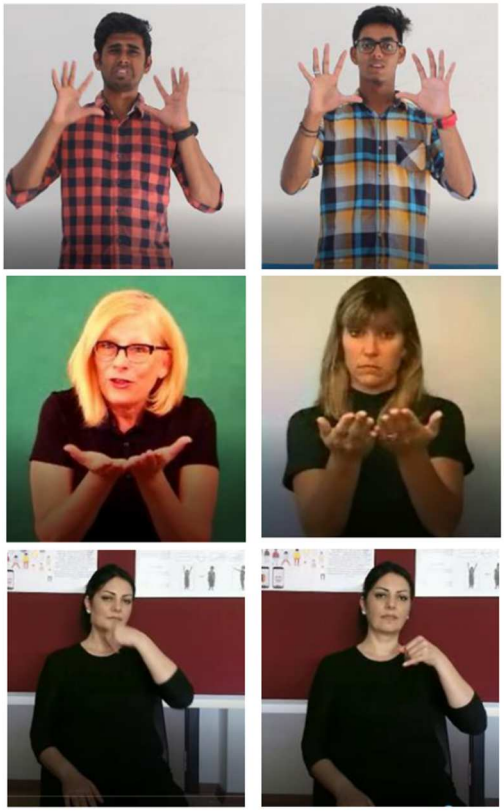

Fig. 1. Examples of the video frames in INCLUDE (top), WLASL (middle), and AUTSL (bottom) datasets.

We are comparing five different deep learning models and architectures on the above three datasets. The description of each deep learning model and architecture is provided below.

A. Method 1. A 4-Layer CNN

Convolutional Neural Networks are used in a variety of applications and without a doubt, the most widely used deep

learning architecture. The enormous popularity and effectiveness of CNN have sparked a recent rise in interest in deep learning. AlexNet [21] sparked interest in CNN in 2012, and it has grown rapidly since then. Researchers went from an 8-layer AlexNet to a 152-layer ResNet [22] in just four years. CNN has become the go-to model for any image-related classification problems because they outperform the competitors in terms of accuracy. The fundamental advantage of CNN over its predecessors is that it discovers essential traits without the need for human intervention. In addition, CNN is computationally efficient. It performs parameter sharing and uses special convolution and pooling algorithms. CNN models can also be made compact to be able to be used on smaller devices, making them universally appealing [23].

In this first model, the following architecture will be used: Four (Convolutions + Pooling) layers followed by two fully connected layers with a SoftMax layer at the end. The input is an image, and the output is the predicted class. The diagram shown in Figure 2 illustrates the architecture of the CNN used.

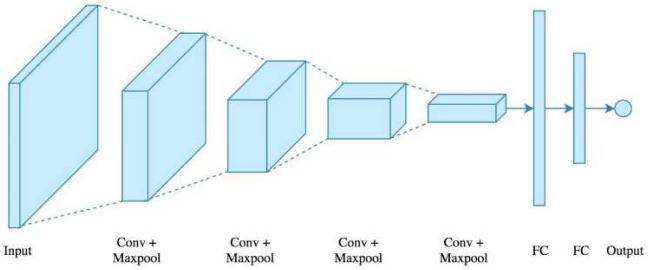

Fig. 2. The architecture of the 4-layer CNN.

A RELU activation function is used when creating the convolution while the padding option is enabled, and the stride value is set to one. When creating the max pooling layer, a 2x2 window is used. During training, a neuron is momentarily "dropped out" or inhibited with probability p at each repetition. This technique is known as dropout and is used to prevent overfitting. This signifies that at this iteration, all this neuron's inputs and outputs will be disabled. At each training step, the dropped-out neurons are resampled with probability p, so a dropped-out neuron at one step can become active at the next. The dropout rate p is set to 0.5.

B. Method 2. VGG16

VGG16 is one of the most popular CNN models [24]. The creators of this model analyzed the networks and enhanced the depth using an architecture with very small (3x3) convolution filters, which outperformed previous state-of-the-art setups significantly. The depth was increased to 16 weight layers, resulting in 138 trainable parameters. VGG16 achieves a 92.7 percent accurate object identification and classification on ImageNet [10]. A pre-trained VGG16 model on this dataset is available and can be used to classify more general images by utilizing transfer learning.

Fig. 3. VGG-16 architecture

Figure 3 shows the VGG16 architecture. The key pointers to note here are as follows: The 16 in VGG16 stands for 16 weighted layers. VGG16 comprises thirteen convolutional layers, five Max Pooling layers, and three dense layers, for a total of twenty-one layers, but only sixteen of those are weight or learnable parameters layers. VGG16 uses a 224x224 input tensor size with three RGB channels.

The most distinctive feature of VGG16 is that, rather than having a huge number of hyper-parameters; it uses 3x3 filter convolution layers with stride 1 and always uses the same padding and Max Pooling layer of 2x2 filter with stride 2. The convolution and Max Pooling layers are placed in a regular pattern throughout the architecture. The Conv-1 layer has 64 filters, the Conv-2 layer has 128 filters, the Conv-3 layer has 256 filters, and Conv-4 and Conv-5 layers have 512 filters. Following a stack of convolutional layers, three Fully Connected layers are added: the first two have 4096 channels each, while the third performs 1000-way image classification and so has 1000 channels (one for each class). The SoftMax layer is the final layer.

Training a VGG16 model from random initial weights takes a long time. The model's trained weights are 528 MB in size. As a result, it consumes a significant amount of storage space and bandwidth, making it inefficient.

C. Method 3. VGG16 with Attention Mechanism

Bahdanau et al. established the famous "Attention Mechanism" approach [25] – which is an NMT technique. Even though the concept of attention has evolved, the mechanism described in this study is still recognized as "Bahdanau Attention". Until this study, such NMT models have relied on numerous networks, each of which had to be trained separately. The research proposes that a single, massive neural network be built and trained to comprehend a sentence and correctly translate it, which is the foundation for all current Sequence to Sequence models based on Encoder-Decoder architecture.

Machine Translation is analogous to finding a target sentence y that maximizes the conditional probability of $p(y|x)$, where x is the source sentence, from a probabilistic standpoint. The goal of an NMT task is to use a parallel training corpus to maximize the conditional probability of sentence pairs. To simulate such a relationship, a parameterized model would be employed with a backpropagation technique utilized to learn the parameter weights. A source sentence is fed into an encoder, which converts it into a fixed-length vector. The translation (target sentence) from the Encoded Vector is output by a decoder. For a given source-target sentence pair, the encoder-decoder system is jointly trained to maximize the conditional probability of an accurate translation. There are some limitations with encoder-decoder architecture. For information about the source sentence, the decoder only uses the last encoded fixed-length vector. It's very difficult for the encoder to compress all the information into a single vector when the source sentence is quite long. The performance of a basic encoder-decoder degrades significantly as the length of a source sentence increases, according to actual evidence.

That study proposes an Encoder-Decoder model extension that learns to 'align' and 'translate' together. When the NMT model generates a translated term, it does a soft search for a set of positions in the source sentence and looks for the positions with the highest concentration of relevant information. It is like selecting the words that make the most sense in the final translation. This is incompatible with the idea of storing the full source sentence into a single fixed-

length context vector. The NMT model then predicts a target translation using context vectors associated with these source positions as well as previously generated translation outputs. The source text is encoded as a sequence of vectors, and the decoder selects a subset of these vectors to produce the translation. It allows the NMT model to interpret long words and do a selective search based on context importance rather than squashing all the information into a single vector.

D. Method 4. VGG16 with Encoder and Decoder

In this method, we use the VGG16 model with Transformer Encoder and Gated Recurrent Units (GRUs) - based Decoder. Due to advances in Sequence Modelling, such as LSTM, and the development of GRUs, generating captions in videos and summarizing them have been recently popular [26]. Existing architectures use CNNs to extract spatiotemporal information and soft attention layers to model dependencies using GRUs or LSTMs. The layers which are attention-based, help in paying attention to the important aspects where recurrent units are also improved; nonetheless, there are problems from recurrent units' intrinsic flaws, with some are addressed by using a different network design.

E. Method 5. I3D with Encoder and Decoder

In this final method, we use a two-stream Inflated 3D (I3D) model with Transformer Encoder and GRUs-based Decoder. Recent advancements in the field of activity recognition have resulted in a variety of network designs that can be used to extract spatiotemporal features. Instead of depending on a recurrent network to encode information from each time step, architectures that can directly offer temporal information are looked at. For example, in [27], these features are extracted for the Transformer model using I3D Convolutional Neural Networks for Activity Recognition. Rather than employing frame-level feature extractors, 3D convolution networks can be used to extract spatiotemporal information from videos. 3D convolutions are used in these structures to encode both spatial and temporal information in videos. Whereas using 2D convolutions on an image or a video (series of frames) results in a single feature map, using 3D convolutions on a set of frames, on the other hand, produces a set of feature maps. The size of the temporal kernel and the strides employed determine the number of feature mappings. Techniques that can reduce dimensions are used to control the total size of a model.

The summary of the models and modifiers used in each method is shown in Table II below.

TABLE II. SUMMARY OF THE METHODS

	Model	Modifier
Method 1	4-Layer CNN	None
Method 2	VGG16	None
Method 3	VGG16	Attention Mechanism
Method 4	VGG16	Transformer Encoder + GRU-based Decoder
Method 5	I3D	Transformer Encoder + GRU-based Decoder

IV. EXPERIMENT AND RESULT ANALYSIS

A. Implementation Setup

The experiment is implemented using the following software and hardware setup.

- Operating System: Windows
- Programming Language: Python 3.9.1, Shell Script
- Package Manager: PIP
- Python Libraries: OpenCV, NLTK, Matplotlib, Numpy, CSV
- A laptop with SSD: 512GB, RAM: 40GB, GPU: NVIDIA 2080 RTI, 12GB

Each dataset is split into three sets namely the training, testing, and validation sets with the following ratio of 70:10:20, respectively.

B. Evaluation metric

Since we have a balanced dataset, in this study we decided to use Accuracy as the one evaluation metric. The accuracy metric, denoted here as A, is a measure of how well a method is getting the right result [28]. It is formally calculated as the percentage of correct predictions (sum of the True Positives and True Negatives) over the entire test population. Mathematically, it is calculated as:

$$A = \frac{TP+TN}{TP+TN+FN+FP} \quad (1)$$

where TP, TN, FN, and FP denote True Positive, True Negative, False Positive, and False Negative, respectively.

C. Results and Analysis

A summary of the results of the experiment is given in Table III below. The table shows that Method 1 is generally the worst of the five models tested. This could be because the four-layer CNN used in the experiment is quite simple as it contains only four feature extraction layers and it was trained from scratch. The results tend to get better as the complexity of the method increases and reach their highest when using Method 5. However, for some unknown reasons, the results on the WLSAL dataset are very poor for all methods. The best accuracy attained is only 0.35 using Method 5.

TABLE III. ACCURACY OF EACH METHOD ON DIFFERENT DATASETS

	INCLUDE	WLASL	AUTSL
Method 1	0.60	0.26	0.76
Method 2	0.63	0.28	0.78
Method 3	0.65	0.31	0.82
Method 4	0.89	0.32	0.86
Method 5	**0.98**	**0.35**	**0.96**

We also provided some example results that were produced by the methods in each dataset. These are shown in Tables IV, V, and VI for INCLUDE, WLASL, and AUTSL datasets, respectively.

TABLE IV. EXAMPLE RETURNED RESULTS USING INCLUDE DATASET

	1	2	3	4	5
Loud	Loud	Loud	Loud	Loud	Loud
Quiet	Quiet	Quiet	Quiet	Quiet	Quiet
Happy	Happy	Happy	Glad	Happy	Happy
Sad	Nice	Unhappy	Sat	Sad	Sad
Deaf	Flat	Dumb	Dean	Unheard	Dumb
Blind	Short	Blink	Blind	Unseen	Blink

TABLE V. EXAMPLE RETURNED RESULTS USING WLASL DATASET

	1	2	3	4	5
Book	Book	Book	Book	Book	Book
Drink	Drink	Drink	Drink	Drink	Drink
Computer	Computer	Computer	Laptop	Computer	Computer
Before	Being	Beside	Before	Beside	Beside
Chair	Charm	Table	Furniture	Chair	Chart
Go	Gone	Gone	Going	Gone	Been

TABLE VI. EXAMPLE RETURNED RESULTS USING AUTSL DATASET

	1	2	3	4	5
Acele	Acele	Acele	Acele	Acele	Acele
Acikmak	Acikmak	Acikmak	Acikmak	Acikmak	Acikmak
Agabey	Agabey	Bebek	Agabey	Agabey	Agabey
Agac	Bahce	Bahce	Bahce	Agac	Bahce
Aile	Baba	Aile	Arkadas	Erkek	Aile
Anne	Bekar	Bekar	Baba	Bebek	Anne

Overall, the quality of the translations is relatively good. In most cases where the exact wording differs, the translated words convey similar information (e.g., Glad for Happy, Unhappy for Sad, Laptop for Computer, Unseen for Blind, Table/Furniture for Chair, or Unheard for Deaf). There are also occasions where the translated words do not convey similar information such as (Dean, Dumb, and Flat for Deaf, Blink for Blind, or Sat for Sad). They could be due to the words' limited contexts in the training data.

V. CONCLUSION

We have presented in this paper the results of our experiment of using deep learning approaches to perform sign language recognition on three sign language datasets. We consider five deep learning models and architectures which are a four-layer CNN, VGG16, VGG16 with Attention Mechanism, VGG16 with Transformer Encoder + GRU-based Decoder, and lastly I3D with Transformer Encoder + GRU-based Decoder. The datasets used are INCLUDE, WLASL, and AUTSL. Our experiment shows that the I3D model with Transformer Encoder and GRU-based Decoder produces the best result out of the five tested methods. The method works well in both INCLUDE and AUTSL datasets achieving 0.98 and 0.96 accuracy, respectively. We predict that most of the difficulties in the field of SLR will be overcome with the aid of deep learning, because of faster hardware to process the input data, precise multi-modal methods, and fresh data illustrating the true variability and distribution of the problem at hand. Although most of the models that have been presented are focused on isolated sign language recognition, we anticipate that the community will soon begin to address the difficulties of continuous sign language recognition, including continuous annotated datasets, tokenization, and long-term multi-modal data modeling, particularly by combining vision and language models.

REFERENCES

[1] C. Song, S. Sudirman, M. Merabti, and D. Al-Jumeily, "Region-adaptive watermarking system and its application," in *Proceedings of 4th International Conference on Developments in eSystems Engineering*, 2011, pp. 215–220.

[2] S. Sudirman, F. Natalia, A. Sophian, and A. Ashraf, "Pulsed Eddy Current signal processing using wavelet scattering and Gaussian process regression for fast and accurate ferromagnetic material thickness measurement," *Alexandria Eng. J.*, vol. 61, no. 12, pp. 11239–11250, 2022.

[3] S. Monica, F. Natalia, and S. Sudirman, "Clustering Tourism Object in Bali Province Using K-Means and X-Means Clustering Algorithm," in *2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, 2018, pp. 1462–1467.

[4] T. Starner and A. Pentland, "Real-time american sign language recognition from video using hidden markov models," in *Motion-based recognition*, Springer, 1997, pp. 227–243.

[5] S. Sako and T. Kitamura, "Subunit modeling for japanese sign language recognition based on phonetically depend multi-stream hidden markov models," in *International Conference on Universal Access in Human-Computer Interaction*, 2013, pp. 548–555.

[6] L.-G. Zhang, Y. Chen, G. Fang, X. Chen, and W. Gao, "A vision-based sign language recognition system using tied-mixture density HMM," in *Proceedings of the 6th international conference on Multimodal interfaces*, 2004, pp. 198–204.

[7] F. Natalia, J. C. Young, N. Afriliana, H. Meidia, R. E. Yunus, and S. Sudirman, "Automated selection of mid-height intervertebral disc slice in traverse lumbar spine MRI using a combination of deep learning feature and machine learning classifier," *PLoS One*, vol. 17, no. 1, p. e0261659, 2022.

[8] F. Natalia, H. Meidia, N. Afriliana, J. C. Young, R. E. Yunus, M. Al-Jumaily, A. Al-Kafri, and S. Sudirman, "Automated measurement of anteroposterior diameter and foraminal widths in MRI images for lumbar spinal stenosis diagnosis," *PLoS One*, vol. 15, no. 11, pp. 1–27, 2020.

[9] L. Liu, C. Xie, R. Wang, P. Yang, S. Sudirman, J. Zhang, R. Li, and F. Wang, "Deep Learning Based Automatic Multiclass Wild Pest Monitoring Approach Using Hybrid Global and Local Activated Features," *IEEE Trans. Ind. Informatics*, vol. 17, no. 11, pp. 7589–7598, 2021.

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, 2009, pp. 248–255.

[11] J. Huang, W. Zhou, Q. Zhang, H. Li, and W. Li, "Video-based sign language recognition without temporal segmentation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018, vol. 32, no. 1.

[12] C. K. M. Lee, K. K. H. Ng, C.-H. Chen, H. C. W. Lau, S. Y. Chung, and T. Tsoi, "American sign language recognition and training method with recurrent neural network," *Expert Syst. Appl.*, vol. 167, p. 114403, 2021.

[13] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko, "Sequence to sequence-video to text," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4534–4542.

[14] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville, "Describing videos by exploiting temporal structure," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4507–4515.

[15] J. Kreutzer, J. Bastings, and S. Riezler, "Joey NMT: A minimalist NMT toolkit for novices," *arXiv Prepr. arXiv1907.12484*, 2019.

[16] R. Cui, H. Liu, and C. Zhang, "A deep neural framework for continuous sign language recognition by iterative training," *IEEE Trans. Multimed.*, vol. 21, no. 7, pp. 1880–1891, 2019.

[17] L. Pigou, S. Dieleman, P.-J. Kindermans, and B. Schrauwen, "Sign language recognition using convolutional neural networks," in *European conference on computer vision*, 2014, pp. 572–578.

[18] A. Sridhar, R. G. Ganesan, P. Kumar, and M. Khapra, "Include: A large scale dataset for indian sign language recognition," in *Proceedings of the 28th ACM international conference on multimedia*, 2020, pp. 1366–1375.

[19] D. Li, C. R. Opazo, X. Yu, and H. Li, “Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison,” *Proc. - 2020 IEEE Winter Conf. Appl. Comput. Vision, WACV 2020*, pp. 1448–1458, 2020.

[20] O. M. Sincan and H. Y. Keles, “Autsl: A large scale multi-modal turkish sign language dataset and baseline methods,” *IEEE Access*, vol. 8, pp. 181340–181355, 2020.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.

[22] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceeding of IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[23] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, and others, “Recent advances in convolutional neural networks,” *Pattern Recognit.*, vol. 77, pp. 354–377, 2018.

[24] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” in *International Conference on Learning Representations*, 2015.

[25] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” *arXiv Prepr. arXiv1409.0473*, 2014.

[26] N. C. 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*, 2018, pp. 7784–7793.

[27] M. Bilkhu, S. Wang, and T. Dobhal, “Attention is all you need for videos: Self-attention based video summarization using universal transformers,” *arXiv Prepr. arXiv1906.02792*, 2019.

[28] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.