2019 International Conference on Sustainable Technologies for
Industry 4.0 (STI), 24-25 December, Dhaka

A New Benchmark on American Sign Language Recognition using Convolutional Neural Network

Md. Moklesur Rahman[*], Md. Shafiqul Islam[†], Md. Hafizur Rahman[‡], Roberto Sassi[§], Massimo W. Rivolta[¶] and Md Aktaruzzaman[||]

[*,†]*Dept. of Computer Science and Eng., The People's University of Bangladesh, Dhaka, Bangladesh.*
[‡]*Dept. of Electrical and Electronic Eng., Islamic University, Kushtia, Bangladesh.*
[§,¶]*Dipartimento di Informatica, Università degli Studi di Milano, Via Celoria 18, 20133, Milano, Italy.*
[||]*Dept. of Computer Science and Eng., Islamic University, Kushtia, Bangladesh.*

{[*]moklesur.ai, [†]msislam.iu, [‡]hafizur.iueee}@gmail.com, {[§]roberto.sassi, [¶]massimo.rivolta}@unimi.it and
[||]aktaruzzaman@iu.ac.bd

***Abstract*—The listening or hearing impaired (deaf/dumb) people use a set of signs, called sign language instead of speech for communication among them. However, it is very challenging for non-sign language speakers to communicate with this community using signs. It is very necessary to develop an application to recognize gestures or actions of sign languages to make easy communication between the normal and the deaf community. The American Sign Language (ASL) is one of the mostly used sign languages in the World, and considering its importance, there are already existing methods for recognition of ASL with limited accuracy. The objective of this study is to propose a novel model to enhance the accuracy of the existing methods for ASL recognition. The study has been performed on the alphabet and numerals of four publicly available ASL datasets. After preprocessing, the images of the alphabet and numerals were fed to a newly proposed convolutional neural network (CNN) model, and the performance of this model was evaluated to recognize the numerals and alphabet of these datasets. The proposed CNN model significantly (9%) improves the recognition accuracy of ASL reported by some existing prominent methods.**

***Index Terms*—Hand gesture, American Sign Language, Convolution neural network, Recognition, ASL.**

I. Introduction

According to the World Health Organization (WHO) [1], the number of people having hearing or listening disability increased from 278 million in 2005 to 466 million in early 2018. It is assumed that this number will be increased to 400 million by 2050 [1]. This deaf community uses a set of signs to express their language (called sign language), which is different for different nations. In other words, a sign language (SL) is a nonverbal communication language, which utilizes visual sign patterns made with the hands or any parts of the body, used primarily by the people who have the disability of hearing and/or listening. Sign languages (SLs) are full-fledged natural languages with their own lexicon and grammar. Different SLs such as American Sign Language (ASL), Australian Sign Language, British Sign Language (BSL), Danish Sign Language, French Sign Language, and many others have been developed for deaf communities. Although there are some striking similarities among the SLs, they are not mutually intelligible and universal. For example, the ASL and BSL are different, even though both of them use the same verbal language. The normal hearing and listening people find it extremely difficult to understand the sign language even of the nation itself. Hence trained SL interpreters are needed during medical and legal appointments, educational and training sessions, etc. The automatic recognition of an SL and its translation into a natural language can establish a proper communication interface between the hearing or listening impaired and normal people.

ASL also predominants as a second language to the deaf communities in the United States and Canada [2]. According to the National Association of Deaf (NAD) [3] in the United States of America, ASL is accepted by many high schools, colleges, and universities in the fulfillment of modern and foreign language academic degree requirements. Besides North America, ASL is also used in many countries across the World, including parts of Southeast Asia and much of West Africa.

There are some works [4]–[6] already reported in the literature for automatic recognition of ASL. Some of these methods have been studied on a sample dataset of few samples, and some using the traditional shallow neural network approach for classification. Shallow neural networks require manual identification of features and relevant features selection. The use of deep learning (DL) techniques for machine learning problems have significantly improved the performance of traditional shallow neural networks, especially for image recognition and computer vision problems. DL is a subfield of machine learning in artificial intelligence (AI). It is a set of algorithms, models with high-level abstractions through architectures which composed of multiple nonlinear transformations. DL algorithms utilize a huge amount of data to extract features automatically, aim in emulating the human brain's ability to learn, analyze, observe, and make an inference, especially for extremely difficult problems. DL architectures create relationships beyond immediate neighbors in the data and generate learning patterns, extract representations directly from data without human intervention. There are different deep

978-1-7281-6099-3/19/$31.00 © 2019 IEEE

learning architectures such as deep belief networks [7], stacked auto encoder [8], convolutional neural networks [9], and so on. Among them, the CNN have utilized multi-layered artificial neural network (ANN) to provide state-of-the-art accuracy in the field of computer vision, medical image analysis, speech recognition, bioinformatics and so on.

A convolutional neural network (CNN), one of the most popular deep learning algorithms, comprising of convolutional layers and then following by one or more fully connected layers, was proposed by Vallian *et al.* [10]. From a computer science perspective, a CNN is a set of digital filters whose weights are estimated during the learning phase. Naturally, there are more complex processes occurring in human brain. Following this analogy, each convolutional layer extracts features from training data. A CNN convolves learned features with input data, utilizes convolutional layers, and turning this architecture into well suited form to process data. CNNs learn to detect different features of data using multiple hidden layers. Every hidden layer increases the difficulty of the learned data features. The problems with the existing methods for ASL recognition are that they have reported their study on a specific dataset (most of the cases), rare comparison of the methods on a common dataset. So, we cannot compare in which degree a method is better/worse than another or which dataset does possess proper variation in samples for sufficient training of a classification model.

To address these problems, we have considered four publicly available ASL datasets on which a number of good works have been reported. We have studied the performance of the proposed model on each dataset, when trained using the training set and tested using the test set (if separate train and test sets are available, or using the 10 fold cross-validation). The performance of the proposed method has also been justified using cross dataset i.e., trained using one dataset and tested on a different dataset. In addition, the performance has been compared with previous methods on the same dataset.

The rest of this paper has been organized as follows: A brief description of related works reported in the literature has been given in section II. A brief description of the dataset considered in this study has been provided in section III. The proposed method has been described in section IV. The results of this study have been presented in section VI, and finally section VII summarizes the study and main findings.

II. Related Works

The researchers are paying more and more attention to recognition of sign language due to its numerous potential applications in many areas such as deaf people communication systems, human-machine interaction, machine control, etc. Research on SL recognition can be divided into two broad categories on the basis of the type of signs: i. static signs based recognition and ii. dynamic signs based recognition. Majority of the studies till conducted are recognition of static signs. The research on SL recognition has been started by the end of 1990.

Many researchers [4], [5], [11]–[13] have proposed techniques to recognize sign languages since then. Every research has its own limitations and is still unable to be used commercially. The brief description of some prominent works on ASL recognition is given here:

Vivek Bheda *et al.* [4] presented a method for the classification of alphanumeric characters of the ASL. Two datasets (one self-generated and MU HandImages ASL [6] were used in their study. They reported an average recognition rate of 67% and 82.5%, respectively for the alphabet and the digits of ASL. W. Huang *et al.* [11] propose 3D Hopfield neural network for hand tracking, feature extraction, and gesture recognition. Their model was tested on a set of 15 different hand gestures and reported an average recognition rate of 91%. T. Starner *et al.* [14] presented real-time systems for recognizing sentence-level continuous ASL. They described two extensible systems for recognition of words in which the first system achieve 92% accuracy when images are taken from a desk mounted camera and the second system obtains 98% recognition rate from cap mounted camera by the user. In both tests, they have used a 40-word lexicon.

Suk *et al.* [12] propose a dynamic Bayesian network (DBN) model for classifying hand sign from video stream. Skin extraction, modeling, and motion tracking are observe by the DBN model. The model is evaluated on the recognition of ASL by 10 isolated gestures and performed over 99% recognition rate. In this work, all gestures are noticeably different from each other. However, the motion tracking features are related to classifying the dynamic letters of ASL (J and Z).

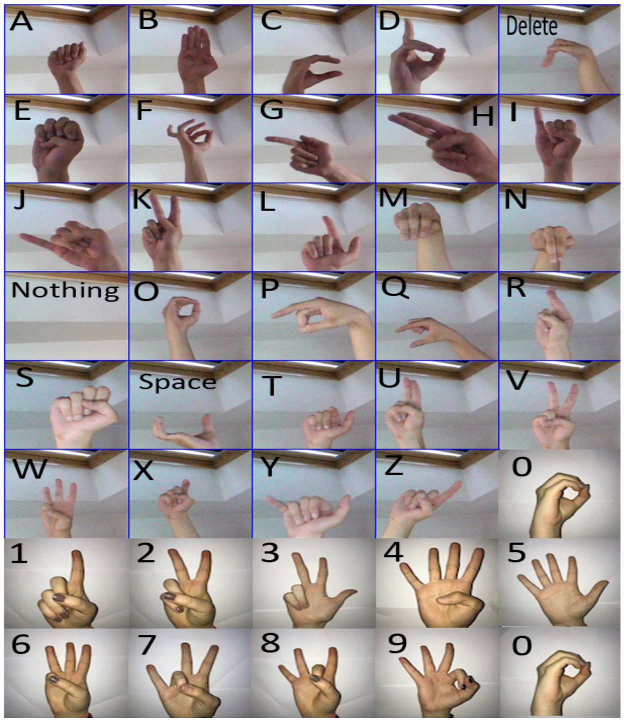

Fig. 1: Some samples of ASL Alphabet and Sign Language Digits dataset.

III. Dataset

In this paper, four separate datasets of ASL have been utilized to analyze the performance of the proposed method. The Massey University Gesture dataset [6] contains standard ASL hand gestures which consist of 2425 images in (PNG) format from 5 individuals and the dataset is called MU HandImages ASL. The sign language digit dataset [15] was collected from 218 Turkey Ankara Ayranci Anadolu high school students. There are 10 samples of each digit collected from each subject. The third dataset that we considered in this study is the ASL finger spelling dataset [5] collected by the Center for Vision, Speech and Signal Processing group at the University of Surrey, UK. The samples of this dataset was divided into color images and depth images. In our work, we consider only color images, consist of 24 static signs (excluding the letters J and Z) of the ASL alphabet, which were acquired from 5 individuals in different sessions with similar lighting and backgrounds. The dataset contains over 65,000 images of the ASL alphabet. The fourth and the last dataset that we considered is the ASL Alphabet dataset [16] consists of 87,000 samples of 29 classes (26 for the letters A-Z and three special characters: delete, nothing, space). A few samples from this dataset are shown in Fig. 1.

IV. Proposed Method

The methodological steps of the ASL recognition system have been described in Fig. 2.

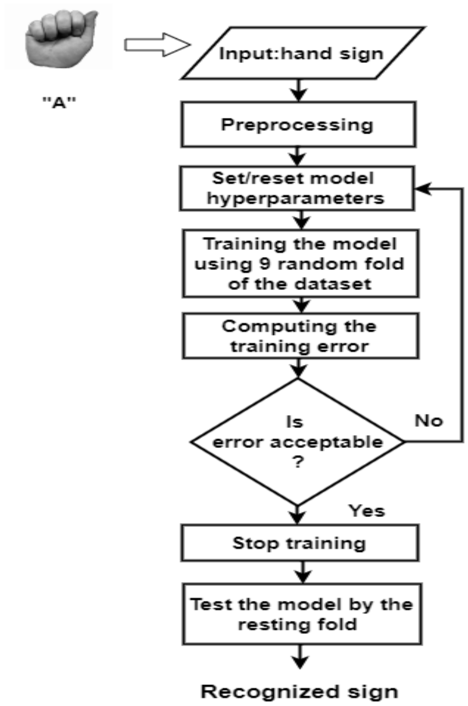
"A"
Input:hand sign
Preprocessing
Set/reset model hyperparameters
Training the model using 9 random fold of the dataset
Computing the training error
Is error acceptable ?
No
Yes
Stop training
Test the model by the resting fold
Recognized sign

Fig. 2: Methodological steps of the proposed ASL recognition system

A. Data Pre-processing

In this study, the raw images are transformed into grayscale images. The grays levels of input images are normalized by the maximum value of the gray level range. The use of low-resolution images provides faster training without too much impact on the recognition rate. The images are resized to 64×64 pixels.

B. CNN Model Description

The architectural design of a CNN contributes to the optimal performance by a proper selection of convolution layers and the number of neurons. There are no universally accepted standard guidelines to select the number of neurons and convolution layers. Here, we have proposed an architecture of a CNN (that we called SLRNet-8) which maximizes the recognition accuracy. Our proposed SLRNet-8 consists of six convolution layers, three pooling layers and a fully connected layer besides the input-output layers. The major steps of the proposed ASL recognition method have been described in the Fig. 3.

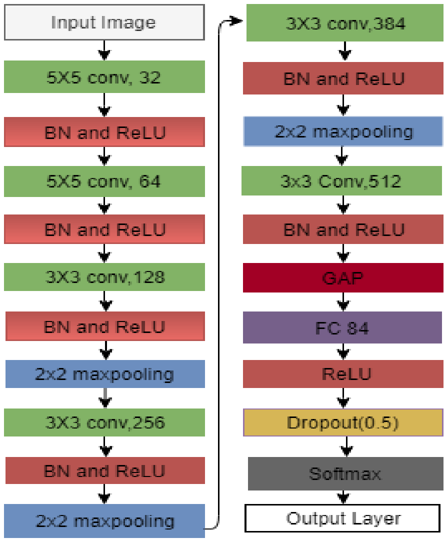

Fig. 3: Architecture of the proposed SLRNet-8 CNN. In this diagram, Conv and BN refer for convolutional layer and batch normalization, respectively.

Input Layer

The pre-processed images are directly applied to the network through its input layer. There are 4096 nodes in the input layer, each node corresponds to every pixel of the image at resolution 64×64.

Convolution Layer

Convolution is the first layer to extract features from an input data and serves as a basic building block of the CNN. In the convolution layer, the kernels extract the salient features from the input data through forward and backward propagation. In our study, this operation is performed by shifting the filters of dimension 3×3 and 5×5 over the input data matrix. At every shifting, it executes element-wise matrix multiplications, and then aggregates the results into a feature map.

The number of kernels used in the convolutional layers may affect the performance of a CNN model. There are no standard guidelines for opting the number of kernels in a convolution layer. In this work, we have performed experiment using different number of kernels from 32 to 512 at different step size, and finally, the combination which maximize the accuracy was selected. A batch normalization (BN) [17] layer which is responsible for accelerating the training process and reducing the internal covariate shift is proceeded by some convolution layer.

Activation Function

In a CNN architecture, activation functions decide which node should be fired at a time. We have applied a ReLU [18] activation function which substitutes all negative values to 0 and remains identical with the positive values. The selection of ReLU was inspired by the learning time of the model. In training, ReLUs are tended to be several times faster [19] than their equivalents (softplus, tanh, sigmoid), and it can diminish the problem of gradient vanishing. The ReLU function is expressed by:

$$\mathrm{ReLU(y)} = \mathrm{max(0, y)}, \tag{1}$$

where y refers to the input to a neuron.

Pooling Layer

Pooling is a significant concept used in deep learning process. It makes the training of a CNN faster and reduces the memory size of the network by reducing the linkages between the convolutional layers. Here, we have used max-pooling operation for this purposes. Max-pooling is the usage of a sliding window across an input space, where the largest value within that window is the output. We have opted 2×2 window size for the max-pooling operation. To remove the overlapping problem, the size of stride has been settled at 2. The resultant dimension of the max-pooling operation can be calculated by the following equation:

$$N_{out} = floor\left(\frac{N_{in} - F}{S}\right) + 1, \tag{2}$$

where N_{in}, F and S refer to the size of the input image, kernel, and stride, respectively.

Global Average Pooling Layer

The global average pooling (GAP) layer is very similar to the max-pooling layer. The only difference is that the entire area is replaced by the average value instead of maximum value. The GAP extremely reduces dimension, where a tensor of size height×width×depth is drastically decreased to 1×1×depth.

Fully Connected Layer

The features map generated by the GAP layer is fed into the fully connected layer (FC) layer. In a FC layer, the neurons in one layer is connected to the neurons in another layer. The FC layer also behaves like a convolution layer with filter of size 1×1.

Dropout Layer

Dropout is a regularization technique that sets input elements to zero with a given probability in a random manner. The over-fitting problem occurs when a model's training accuracy is too high in contrast to testing accuracy. In CNN models, a dropout layer followed by the FC layer allows to prevent the over-fitting problem and enhances the performance [20], [21] by setting activation to zero in a random manner during the training process. The probability of dropout used in this study was 0.5.

Output Layer

The output of the classification model i.e., the prediction of a class with a certain probability is obtained at this layer. Here it is also mentioned that the target class should have the highest probability. We set out the number of neurons in the output layer as there are categories. In the case of a multiclass classification problem, the Softmax function returns the probabilities of each class, where the target class will have the highest probability. The mathematical expression for the Softmax function is given by:

$$\sigma(X)_i = \frac{e^{x_j}}{\sum_{k=1}^{K} e^{x_k}}; \text{ for } i = 1, 2, 3, ...K. \tag{3}$$

where, x_i are the inputs from the previous FC layer used to each Softmax layer node and K is the number of classes.

V. Training Details

A. Data Augmentation

Data augmentation means increasing the number of samples as well as adding the variations in the samples for better training. The traditional data augmentation techniques [22], [23] include rotation, scaling, shifting and flipping. To keep smaller the computational burden, here, data augmentation are performed by randomly changing the angles between $-10°$ to $10°$, zooming by 10%, and shifting by 10% on height and width. These parameters were chosen by trial and error basis which provided optimum accuracy.

B. CNN Training

There was no distinct test or train sets for any of the datasets considered in this work. In this study, every dataset was randomly partitioned into K=10 folds, and $K-1$ of them were applied for training the model, and the remaining one was applied for testing the performance of it. This process of training was repeated 10 times, and finally, the average of all accuracies was reported as the accuracy of the model. Here, we choose cross-entropy cost function [24], and a gradient descent-based Adam optimizer [25] with a learning rate 0.001 was selected. Our SLRNet-8 model was trained for up to 200 epochs with 64 steps per epoch, and a batch size of 128. If the validation accuracy did not improve in six consecutive epochs, the learning rate of the model was updated to 75% of its previous value. We allowed early stopping, and training was halted if the validation loss did not improve for thirty consecutive epochs. The very small real numbers that come from normal distribution were initially assigned to the network weights with a weight decay rate of 1×10^{-6}. The model

TABLE I: **PERFORMANCE OF THE PROPOSED MODEL**

Dataset	Category	Accuracy(%)
MU HandImages ASL	Digit	100
	Alphabet	99.95
Sign Language Digits	Digit	99.90
ASL Alphabet	Alphabet	100
Finger Spelling	Alphabet	99.99

was trained on a desktop computer under 64 bit Windows 10 environment with NVIDIA Titan Xp PRO 12 bit GPU, 3.98 CPU, 8 GB RAM, and 1 TB HDD. The training of the model was completed within 150 epochs.

The performance of the proposed model was evaluated for recognition of digits and alphabet of each dataset separately using 10 fold cross-validation. Besides this, its performance was evaluated also for the case of mixing digits and alphabet of each dataset.

VI. Results

The training and validation accuracy of the model on MU HandImages ASL digit dataset has been depicted in Fig. 4. The average accuracy of the proposed model for recognition of ASL of every dataset has been presented in Table I. It is observed that the model recognized both digit and alphabet signs of every dataset with about 100% accuracy. The lowest accuracy (99.90%) has been reported for the digits of sign language digits dataset. On the other hand, the digits of MU HandImages ASL dataset has been recognized with 100% accuracy. This very good results may be due to the application of sufficient data augmentation that adds more variation in the training samples to make the model capturing all possible changes.

The performance of the proposed model has been evaluated also in the case when the digits and alphabet have been mixed together, and its performance has been represented in Table II. Since all datasets except MU HandImages ASL contain either alphabet or numeral signs of ASL and sign language digits dataset contain only the digit signs of the ASL, we have combined sign language digits with the ASL digits and finger spelling datasets separately to evaluate the performance on alphanumeric datasets. It is observed from Table I and Table II that the recognition accuracy of the model is slightly reduced for alphanumeric recognition than the individual recognition of digits or alphabet. The recognition rate on MU HandImages ASL is reduced from 100% (for digits) to 99.92%. However, it is still very high, approximately 100% accuracy. Similarly, the recognition rate of ASL (alphabet) of the finger spelling dataset reduced from 99.99% to the average recognition rate of 99.90%. Thus, we can conclude that the model performance is not affected significantly by mixing the digits with the alphabet datasets.

It is very difficult and not rational to strictly compare some methods when they are not evaluated on the same dataset, becuase the performance of a recognition method may vary due to the dataset used for training and also on the quality of test samples used to evaluate the model's performance. To compare the performance of the proposed model with some existing methods, we have chosen MU HandImages ASL and Finger Spelling datsets on which previous works have been done by [4] and [5] using CNN. The comparison between the proposed SLRNet-8 model and the models proposed by Bheda *et al.* [4] and Brandon *et al.* [5] has been mentioned in Table III. It is observed that out model has significantly ($\geq$ 9%) improved the recognition accuracy reported by the previous models [4], [5] of CNN on the same dataset.

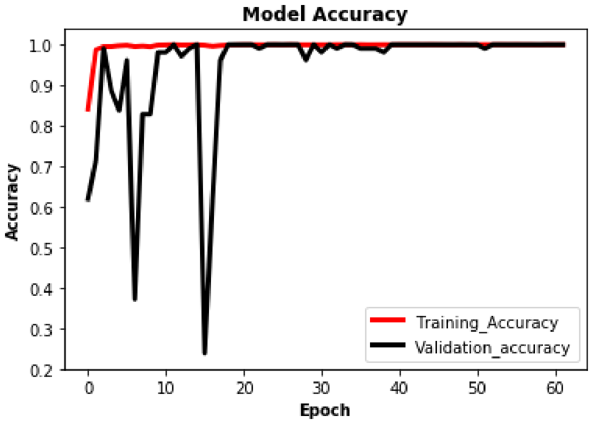

Fig. 4: Performance curve of the proposed model for recognition of alphabet of the MU HandImages ASL dataset. From the figure, it is seen that both training and validation performance curve converges to 100% at epoch 50. Hence, the model converges very rapidly.

TABLE II: **PERFORMANCE OF THE PROPOSED MODEL WHEN DIGIT AND ALPHABET DATASETS ARE COMBINED**

Dataset	Category	Accuracy(%)
MU HandImages ASL	Alhanumeric	99.92
Sign Language Digits and ASL Alphabet	Alphanumeric	99.90
Sign Language Digits and Finger Spelling	Alphanumeric	99.90

VII. Conclusion

American Sign Language is one of the most popular Sign Languages in the World. In this study, we proposed a convolutional neural network model (SLRNet-8) for automatic recognition of ASL, and its performance has been evaluated on

TABLE III: **COMPARISON WITH PREVIOUS RESEARCH**

Model	Dataset	Accuracy(%)
CNN [4]	MU HandImages ASL	91.70
	Self-generated	89.75
CNN [5]	MU HandImages ASL and Finger Spelling	91.63
CNN (Proposed)	**MU HandImages ASL**	**99.92**
	Finger Spelling	**99.99**

four ASL datasets of digits and alphabet. The performance of the proposed model for ASL has been compared with some prominent works already reported on the same dataset. The proposed model has significantly improved the recognition accuracy of the ASL of some prominent works already exists in the literature. The model predicts every sign with 100% accuracy. In this study, we have considered only isolated digit or letters of ASL from static images. In the future, the model can be applied for the recognition of sentence-level continuous words of ASL or for recognition of ASL from the video.

Acknowledgment

We are thankful to the NVIDIA corporation for supporting the study by providing a GPU card.

References

[1] World Health Organization, WHO. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss

[2] J. Cummins, "Bilingualism and second language learning," *Annual Review of Applied Linguistics*, vol. 13, pp. 50–70, 1992.

[3] National Association of Deaf. [Online]. Available: https://www.nad.org/

[4] V. Bheda and D. Radpour, "Using deep convolutional networks for gesture recognition in american sign language," *arXiv:1710.06836*, 2017.

[5] B. Garcia and S. A. Viesca, "Real-time american sign language recognition with convolutional neural networks," *Convolutional Neural Networks for Visual Recognition*, vol. 2, 2016.

[6] A. Barczak, N. Reyes, M. Abastillas, A. Piccio, and T. Susnjak, "A new 2d static hand gesture colour image dataset for asl gestures," 2011.

[7] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[8] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 1096–1103.

[9] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, 1998.

[10] R. Vaillant, C. Monrocq, and Y. Le Cun, "Original approach for the localisation of objects in images," *IEE Proceedings-Vision, Image and Signal Processing*, vol. 141, no. 4, pp. 245–250, 1994.

[11] C.-L. Huang and W.-Y. Huang, "Sign language recognition using model-based tracking and a 3d hopfield neural network," *Machine vision and applications*, vol. 10, no. 5-6, pp. 292–307, 1998.

[12] H.-I. Suk, B.-K. Sin, and S.-W. Lee, "Hand gesture recognition based on dynamic bayesian network framework," *Pattern recognition*, vol. 43, no. 9, pp. 3059–3072, 2010.

[13] M. S. Islalm, M. M. Rahman, M. H. Rahman, M. Arifuzzaman, R. Sassi, and M. Aktaruzzaman, "Recognition bangla sign language using convolutional neural network," in *2019 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)*, Sep. 2019, pp. 1–6.

[14] T. Starner, J. Weaver, and A. Pentland, "Real-time american sign language recognition using desk and wearable computer based video," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 20, no. 12, pp. 1371–1375, 1998.

[15] F. Beşer, M. A. Kizrak, B. Bolat, and T. Yildirim, "Recognition of sign language using capsule networks," in *2018 26th Signal Processing and Communications Applications Conference (SIU)*. IEEE, 2018, pp. 1–4.

[16] A. Deza and D. Hasan, "Mie324 final report: Sign language recognition."

[17] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv:1502.03167*, 2015.

[18] A. F. Agarap, "Deep learning using rectified linear units (relu)," *arXiv:1803.08375*, 2018.

[19] 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.

[20] S. Park and N. Kwak, "Analysis on the dropout effect in convolutional neural networks," in *Asian Conference on Computer Vision*. Springer, 2016, pp. 189–204.

[21] B. Ko, H.-G. Kim, K.-J. Oh, and H.-J. Choi, "Controlled dropout: A different approach to using dropout on deep neural network," in *Big Data and Smart Computing (BigComp), 2017 IEEE International Conference on*. IEEE, 2017, pp. 358–362.

[22] X. Cui, V. Goel, and B. Kingsbury, "Data augmentation for deep convolutional neural network acoustic modeling," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4545–4549.

[23] M. M. Rahman, M. S. Islam, R. Sassi, and M. Aktaruzzaman, "Convolutional neural networks performance comparison for handwritten bengali numerals recognition," *SN Applied Sciences*, vol. 1, no. 12, p. 1660, Nov 2019. [Online]. Available: https://doi.org/10.1007/s42452-019-1682-y

[24] S. Mannor, D. Peleg, and R. Rubinstein, "The cross entropy method for classification," in *Proceedings of the 22nd international conference on Machine learning*. ACM, 2005, pp. 561–568.

[25] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv:1412.6980*, 2014.