Design Studies and Intelligence Engineering
L.C. Jain et al. (Eds.)

doi:10.3233/FAIA220050

Deep CNN for Static Indian Sign Language Digits Recognition

Jennifer Eunice R[1] and D Jude Hemanth[1]
[1]*Department of ECE, Karunya Institute of Technology and sciences, Coimbatore, India*

Abstract. Sign language recognition (SLR) is a significant solution for the hearing and speech disabled to connect with the people. However, SLR system faces complexities such as low accuracy, overfitting, hand occlusions, and high interclass similarities. In this paper, a deep learning-based Convolution Neural Network model is proposed for Sign language recognition to address the issues. Our model uses Indian Sign Language dataset which comprises 10 class with a total of 2072 static digit gestures ranging between 0 to 9. Each class has 207 images. The proposed model generated desired outcome and the results are evaluated with varied optimizers such as Adam, RMS Prop, Stochastic gradient descent (SGD) optimizers. CNN model with SGD achieved training and validation accuracy of 99.72% and 98.97% respectively. The training and validation loss were comparatively minimum for our model. Further, the performance evaluation of the proposed model was analyzed based on precision, recall, F-score value. Our method shows its effectiveness over other machine learning models with a recognition rate of 99%.

Keywords. Sign Language Recognition, Convolutional Neural Network, Indian Sign Language

1. Introduction

Unlike common man, hearing and speech impaired community face social challenges in their day to day activity. They are hesitant to socialize with the society because of their inabilities to express their emotions and feelings, their hardships in facing job interviews, struggle for Education, trouble in making voice calls or video calls through mobile phones, etc., Sign Language (SL) is one of the elemental forms of communication for the hearing and speech impaired community to interact with the society. Though Sign Language is a basic mode of communication for the disabled, only very few understand sign language. It is not a universal language and has different variants such as Indian Sign Language (ISL), American sign language (ASL), British Sign language (BSL), etc., Hence, to interact with this disabled people, there is a need to promote a system that can bridge over this communication gap[1]. Sign language is a visual mode of communication that includes articulated hand gestures along with the body, head, lip movement, and facial expressions, whereas one should not perturb sign language with body language[2].

The conventional methods involve a professionally trained linguist who acts as an interpreter between the hearing majority and the disabled community. However, this method is inefficient since there is a shortage in number of trained human translators. Whereas, written communication is also cumbersome since the hearing and speech impaired are less skilled in writing spoken language. For example, in case of any emergency, the fastest way to approach is through a spoken conversation where written communication is not appropriate[3]. The objective of this paper is to find a suitable for designing a reliable and affordable sign language recognition system. There have been multiple approaches implemented in the field of sign language.

The sensor and glove based approach is one among them which substitutes human translators[4]. This technique is a computer run application that facilitates easy interaction between signers and non-signers. Organizations like Google and Microsoft are exploring similar technology to interpret signs using sensors. Sensors track hand, finger, facial expressions, and gestures to interpret the sign. In this technique, the connection is established using external hardware which makes this method expensive and non-handy in the case of real-time recognition.

Automatic sign recognition is a tedious multidisciplinary issue yet needs to be solved. Further research works in sign language have been reported using machine learning techniques where feature extraction is not automatic and generates less accurate results. Machine learning [5] algorithms are suitable for small size data, but it fails when it comes to training large data. These techniques consume maximum time in developing and evaluating features to enhance model accuracy. In the case of problem-solving approach, this technique divides the problems into steps and solves them individually and finally combines them to obtain results. Certain Machine learning algorithms like decision tree[6], backpropagation[7], Support vector machine (SVM) [8][9] and K-Nearest Neighborhood (KNN)[10] generates accurate results in sign language recognition but they succumb to environmental changes and is applicable where limited and more structured data is available.

On the contrary deep learning can automatically learn the set of features from given raw data rather than manual process of hand-crafted feature engineering. Through deep learning techniques, the computational models are trained layer-wise to learn on data mimicking human brain as of how it perceives and understands multimodal information[11]. These techniques outperform well in several tasks and manage complex data from different sources like visual, medical, audio, social, and sensor with ease. Deep learning networks are inspired by Biological architectures and learning algorithms [12].

The reason behind this work is to examine a suitable method to develop an automated sign language recognition system (ASLR)[13] that transcribes sign language into speech or text. In this paper, a CNN model with SGD optimizer is proposed for Indian Sign language dataset which shows a promising outcome.

2. Related Works

Sign language recognition (SLR) can be divided into two categories they are Isolated Sign language recognition (ISLR) and Continuous sign language recognition (CSLR). Static and dynamic gestures fall under ISLR, where alphabets and numerical are static gestures and dynamic gestures include words. CSLR is used in the recognition of sentences which is the combination of static and dynamic gestures. The two major constraints in feature extraction from static gestures include noisy images and variations in background. A CNN based hybrid Scale-invariant transformation(hybrid SHIFT)[14] was proposed to address the issues in feature extraction from static gestures which outperformed well when compared with CNN using Adaptive thresholding technique. Raimundo et al. [15] also proposed a similar CNN Architecture with additional pre-processing steps by applying morphological filters, contour generation, polygonal

Approximation, and segmentation for recognizing static gesture. However, these models are not extended to dynamic and continuous sign recognition.

Movement Epenthesis, extraction of spatial-temporal features are the key problems identified in dynamic feature extraction. A novel 3D CNN architecture that captures spatial-temporal features from the raw images without any pre-trained knowledge was proposed by Zhi et al.[16]. A simple 3D convolution classifier is used in this article to extract temporal features from dynamic gestures where there are different advanced techniques are available that can be incorporated for temporal feature extraction in a dynamic SLR system.

In continuous sign language recognition (CSLR) system, sign sequence segmentation and sign recognition are major challenges. Dynamic time warping based Level Building algorithm [17] is proposed to address the challenges. However, it suffers from bad similarity function and high computation. Wenwen et al., [18] proposed a hidden Markov model to calculate the similarity between the signing model with good accuracy and reduced computational cost. This technique achieved superior recognition rate along with lower computational constraints with a runtime of 8.28s for each sentence and 12.20% of error rate when compared with LB-HMM and LB-DTW. On the other hand, Javed et al. [19] proposed a three-stream CNN architecture using a kernel-based extreme learning machine (KELM) Fusion classifier [20] to learn complementary features from three different motion templates namely Motion History Image(MHI), Dynamic Image(DI), and RGB motion image(RGBMI) to achieve better performance as well as recognition accuracy in real-time recognition. Though this technique attained accurate outcomes than traditional SoftMax, these models excluded handcrafted features in determining its recognition accuracy. Kumar et al.[21] infused Joint angle topographic descriptor (JATD) and joint distance topographic descriptor (JDTD) in the two-stream CNN classifier to enhance the prediction accuracy along with test prediction speed. A two-layer CNN was proposed by Suresh et al.[22] to address the recognition accuracy using 6 different datasets with two optimizers Adam and Stochastic gradient descent. The outcome of CNN with Adam Optimizer achieved an accuracy of 99.51%. Table 1 highlights the issues identified using different deep learning techniques in previous research works.

Table 1. Detailed survey on merits and demerits of different approaches based on the issues identified from previous works.

Reference	SLR Category	Data sets	Issues Identified	Technique	Accuracy	Drawbacks
[23]	CLSR	NYU, RKH, First-person	1.Bacground complexity 2.Hand Occlusion 3.Hand pose estimation	3DCNN + LSTM + SSD	99.8%	Achieved accuracy only with described datasets.
[24]	CLSR	LSA64	Inefficient Traditional handcrafted feature extraction methods	3D CNN	93.9%	1. Models not trained with large datasets 2. Lack of testing with diverse gestures and situations
[25]	CLSR	RWTH-PHOENIX - Weather	1. Overfitting 2. Shortfall in precise frame level supervision in training CNN.	RNN + CNN+ Staged optimization	-	Needed algorithm extension for multi-modal versions

[26]	Dynamic SLR	RWTH-BOSTON-50 ASL, ASLLVD Corpus	1. Partial hand occlusions	CNN + Particle filter + HEI	89.33%	HEI method subsided in recognition accuracy with other datasets
[27]	Dynamic SLR	ArSL	1. Improve accuracy	CNN + RNN	95.2%	RNN learns some additional features
[28]	Static SLR	ASL Benchmark & NTU Digit	1.High interclass similarities 2.Overfitting 3.Occulsion 4. Large Intraclass variation	CNN + Multiview augmentation + inference fusion	93%	Failures in classification of similar sign pairs
[29]	Static SLR	TFS	1. Complication in identification of invalid sign from valid sign	CNN + HoG	91.26%	Uncertain if multiple invalid signs occur in single frame.

ArSL – Argentinian Sign Language, SSD – Single Shot Detector, LSTM – Long Short-Term Memory, HEI – Hand Energy Image, HoG – Histogram of oriented gradient, RNN- Recurrent Neural network, TFS – Thai Finger spelling.

CNN [30] based model outperforms all other handcrafted feature extraction models. The main objective of our proposed work is to develop a completely versatile Deep learning-based sign language recognition system. Our proposed CNN Model with SGD optimizer is compared with other existing models that outperform well.

3. Proposed System Architecture.

The proposed system design has 3 stages. In the first stage, the input raw data is preprocessed. We introduce this preprocessing technique in the earlier step systematically to enhance CNN performance and accuracy. The entire flow diagram of the system is depicted in Fig. 1.

In the proposed system architecture, we use Indian Sign Language static gesture datasets. The RGB images of dataset are pre-processed using image resizing, segmentation, and normalization. After performing normalization, the pre-processed images are store in the RGB data store for further usage. If the pre-processing is not satisfied, again it goes back to the first stage for feature extraction. In the next stage, the pre-processed RGB images from the data store are divided into two categories. One for the testing phase and the other is for the training phase. In the training phase, 80% of datasets are used to train the proposed CNN model. The trained CNN is used in the testing phase for classification. If the trained CNN classifier fails to classify the images in the testing phase, then fine-tuning of hyperparameters in the CNN model is compassed and again the model undergoes training process. After training phase, the trained CNN classifier is classified using 20% of test data and the classified RGB test images are compared with pre-processed RGB images in the data store to substantiate the accuracy of the output image.

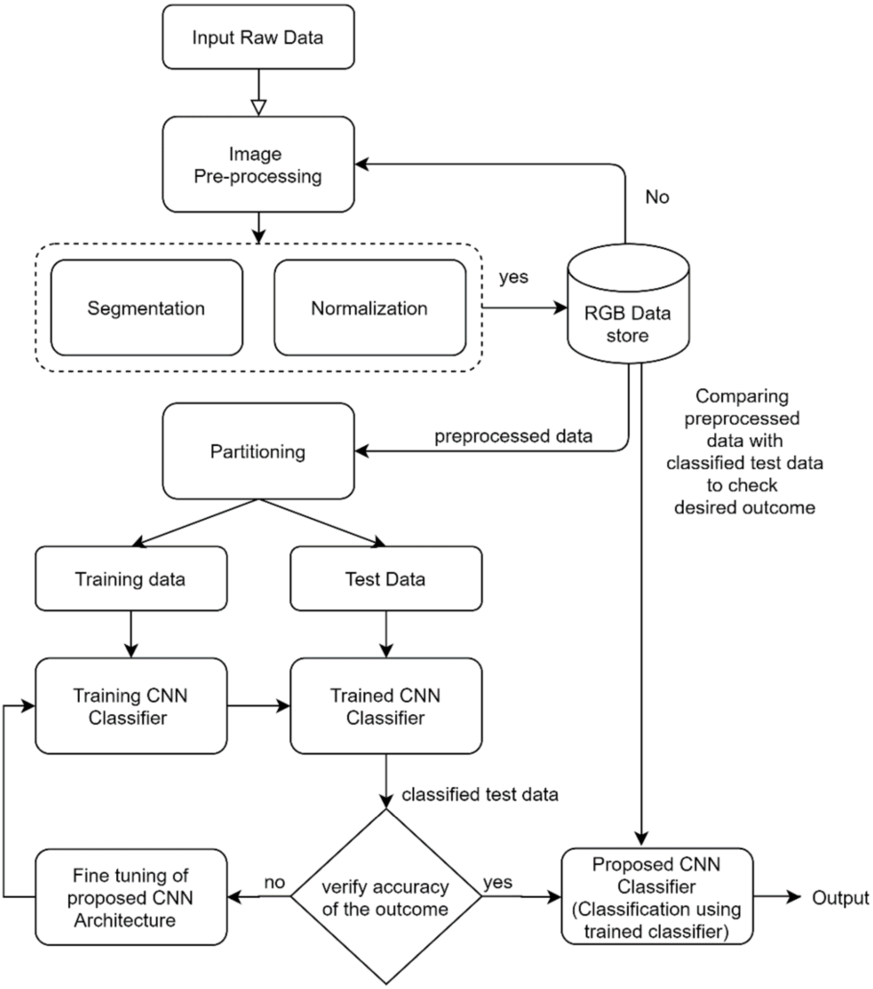

Input Raw Data
Image Pre-processing
No
Segmentation
Normalization
yes
RGB Data store
Partitioning
preprocessed data
Comparing preprocessed data with classified test data to check desired outcome
Training data
Test Data
Training CNN Classifier
Trained CNN Classifier
classified test data
Fine tuning of proposed CNN Architecture
no
verify accuracy of the outcome
yes
Proposed CNN Classifier (Classification using trained classifier)
Output

Figure 1. Proposed system architecture.

3.1 Proposed CNN Model

Convolution neural network is the best-chosen model to solve image classification problems in computer vision. They are widely used in object detection, image recognition, image classification, face detection, and so on. In deep learning, CNN is a self-learning model that classifies on its own using forward and backward propagation rather than programming. The convolution layer has a different layer namely input layer, convolution layer, pooling layer, and fully connected layer.

In General, for training and testing a CNN model, the input values pass through a series of different convolution layers, pooling layer, fully connected layers and finally, the SoftMax function is applied to classify an object under the probabilistic values of 0 and 1. The first step involved in CNN is to read the input image and break the input into small fragments. The first layer of convolution recognizes some parameters like colors, horizontal vertical lines, etc. The second layer further recognizes some complex parameters and the fully collected layer finally learns the complete object from the different layers and classifies the object. Let ‘F’ be the frame size of an image, ‘W refers to input size, and let ‘S’ denotes the stride of kernel and ‘P’ denote padding then the output size of the convolution layer is defined in equation (1).

$$\text{Output of convolutional layer } = +1 \tag{1}$$

The convolution layer is the basic building block in a CNN model where a filter strides over our image and transforms it based on the filter values. For example, let us consider that input has “x” feature maps as input, “k” as kernel, and “y” as output and let “n x m” be the filter size. Feature mapping is done using the following equation (2),

$$(\mathrm{m}, \mathrm{n}) \;=\; (\mathrm{x} \,*\, \mathrm{k})\;[\mathrm{n\,x\,m}] \;= \sum\sum_{i,j} k[i,j]x[n-i,m-j] \tag{2}$$

For example, if we convolute a 6 x 6 image with 3x3 kernel then the feature map obtained will be of 4 x 4. Repeating the convolution will end up with a shrinking image so it has to be limited for a particular number so that the image does not completely disappear. For this reason, padding is done. In pooling layer, no particular parameters are learned but they are meant to reduce the size of image dimensions. After final pooling there comes to flatten layers since the outcome after convolution will be in the form of a matrix. The flatten layer unrolls all the matrix values into vectors and are mapped to fully connected layers. This fully connected (FC) layer is a feed-forward neural network and finally, SoftMax is applied at the end of FC layer to get a probabilistic classification.

The proposed CNN model which has five convolution layers with an input image dimension of 64 x 64 x 3. A 3 x 3 kernel is used to stride through the convolution layer. Max pooling of 2 x 2 is employed and achieves max out of (30, 30). In addition to it, a dropout of 0.5 is added which a powerful regularization technique to avoid overfitting and enhance the accuracy of the results by randomly nullifying few neurons during the training phase. Finally, our model has two dense layers with 512 units. The proposed CNN model is compared with various machine learning techniques and has achieved a state of art results.

4. Experiments and results

The performance of our proposed model is evaluated under two steps using ISL digit dataset. In our first step, model is tested by trying different optimizers and finding a suitable optimizer. Secondly, the performance evaluation is done using different machine learning algorithms such as KNN, Logistic regression, and Neural Networks.

4.1 Dataset

Indian Sign language emerged to solve the interaction issue of abled people with hearing and speech impaired community. Here in this paper, our model is trained using the Indian Sign Language Digit dataset which has 10 classes with 207 RGB color images under each class. An image dimension of 100 x 100 is fetched as an input from the ISL digit dataset to train the model. Digits from 0 to 9 were considered with a total of 2072 images. A sample of ISL digit Dataset is shown in Fig. 2.

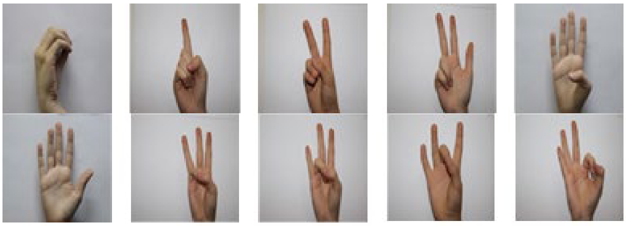
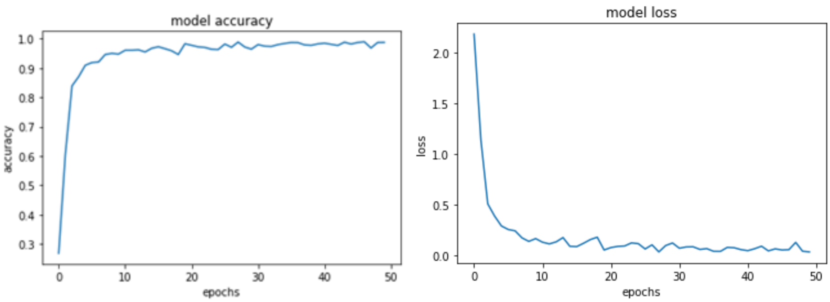


Figure 2. Sample Single Handed static ISL Digit dataset

4.2 Experimental Analysis

In this paper, our model has experimented with different optimizers such as RMS Prop, Adam, and SGD optimizer. Optimizers are used to minimize the losses in models by changing the learning rate and weights in a neural network. A maximum of 50 epochs with a batch size of 5 is used to train the model. The model accuracy and model loss are plotted in Fig. 3. While testing the proposed model with different optimizers, Stochastic gradient descent stands out over Adam and RMS Prop optimizers with training, validation accuracy of 99.72%, and 98.97%. The detailed experimental reports of colored images concerning optimizer are shown in Table 2.



(a) (b)

Figure 3. (a) proposed model training accuracy, (b) proposed model loss

Table 2. Experimental Analysis with respect to optimizer.

Optimizer	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss
RMS Prop	0.9899	0.0456	0.9735	0.0593
Adam	0.9928	0.0166	0.9842	0.0328
SGD	**0.9972**	**0.0123**	**0.9897**	**0.0246**

4.3 Performance Analysis

Performance analysis of the Indian Sign language digit dataset is tested on various machine learning models such as KNN, Logistic regression, and neural networks. Table

3. shows the comparative analysis of the proposed CNN model with existing machine learning techniques. The quality of the classifier is assessed using certain evaluation metrics such as precision, recall, and F-score values. While testing, Neural network achieved better accuracy of 95% when compared with KNN and logistic regression which achieved only 62% and 75% recognition accuracy. Above them all, deep learning-based CNN model outperformed well with an accuracy of 99% for ISL dataset. Precisions are the number of correct positive predictions that occurred in runtime. In other words, precision can be calculated as the ratio of exact prediction of positive values to the total number of positive predicted examples. Equation (3) defines the precision as,

$$Precision = \frac{true\ positives}{true\ positives + false\ positives} \tag{3}$$

Recall, also known as “sensitivity” is defined as the ratio of retrieved instances out of relevent instances. Equation (4) denotes the function of recall,

$$Recall = \frac{true\ positives}{true\ positives + false\ negatives} \tag{4}$$

F-score is a parameter to measure the accuracy of a model which is the harmonic mean of precision and recall. The expression of F-score is given in equation (5).

$$F - Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{5}$$

Table 3. Performance Analysis pf proposed CNN model with various machine learning models.

Model	Precision	Recall	F1 Score	Accuracy
KNN	0.66	0.63	0.64	62%
Logistic regression	0.75	0.75	0.75	75%
Neural network	0.92	0.93	0.93	95%
CNN	**0.96**	**0.95**	**0.95**	**99%**

5. Conclusion and Future works

In this work, a deep learning-based CNN model is developed for the recognition of static Indian Sign language digits. Our proposed CNN model is build using 5 convolution layers, 2 Dense layers, and 2 dropout layers. An input image of 64 x 64 x 3 is fetched to our convolutional model. This model is trained and tested on the ISL digit dataset with a total of 2072 images and has 10 classes of static hand digit gestures representing from 0 to 9. It is observed that our CNN model with stochastic gradient descent optimizer achieved the highest training and validation accuracy of 99.72 % and 98.97% when compared with other optimizers. The performance of our proposed model is also evaluated with different machine learning models based on performance metrics such as precision, recall, and F – score, where our model proved to be the best by achieving a state-of-the-art result. Our model achieved 99% accuracy when compared to KNN, Logistic regression, and Neural networks.

Sign language recognition faces challenges when it comes to real-time recognition. Also incorporating a huge number of manual signs in the processing with minimum error rate is another challenge. Our future scope is to train our proposed architecture with different datasets and extend our model to recognize continuous sign language.

References

[1] P. Kaur, P. Ganguly, S. Verma, and N. Bansal, "Bridging the Communication Gap : With Real Time Sign Language Translation," pp. 485–490, 2018.

[2] G. A. Rao, K. Syamala, P. V. V Kishore, and A. S. C. S. Sastry, "Deep Convolutional Neural Networks for Sign Language Recognition," pp. 194–197, 2018.

[3] L. P. B, S. Dieleman, P. Kindermans, and B. Schrauwen, "Sign Language Recognition Using Convolutional Neural Networks," pp. 572–578, 2015, doi: 10.1007/978-3-319-16178-5.

[4] C. Oz and M. C. Leu, "Engineering Applications of Artificial Intelligence American Sign Language word recognition with a sensory glove using artificial neural networks," *Eng. Appl. Artif. Intell.*, vol. 24, no. 7, pp. 1204–1213, 2011, doi: 10.1016/j.engappai.2011.06.015.

[5] K. K. Dutta, "Machine Learning Techniques for Indian Sign Language Recognition," *2017 Int. Conf. Curr. Trends Comput. Electr. Electron. Commun.*, pp. 333–336, 2017.

[6] G. Fang, W. Gao, and D. Zhao, "Large Vocabulary Sign Language Recognition Based on Fuzzy Decision Trees," vol. 34, no. 3, pp. 305–314, 2004.

[7] R. Achkar, G. A. Haidar, D. Salhab, A. Sayah, and F. Joubran, "Sign Language Translator using the Back Propagation Algorithm of an MLP," 2019, doi: 10.1109/FiCloudW.2019.00019.

[8] S. Limkar, "Available on : Elsevier-SSRN Indian Sign Language Recognition using SVM Classifier," 2019.

[9] J. Ekbote, "Indian Sign Language Recognition Using ANN And SVM Classifiers," 2017.

[10] F. Utaminingrum, I. K. Somawirata, and G. D. Naviri, "Alphabet Sign Language Recognition Using K-Nearest Neighbor Optimization," vol. 14, no. 1, pp. 63–70, 2019, doi: 10.17706/jcp.14.1.

[11] O. K. Oyedotun and A. Khashman, "Deep learning in vision-based static hand gesture recognition," *Neural Comput. Appl.*, 2016, doi: 10.1007/s00521-016-2294-8.

[12] P. Janssen, S. Kalkan, and M. Lappe, "Deep Hierarchies in the Primate Visual Cortex : What Can We Learn for Computer Vision ?," vol. 35, no. 8, pp. 1847–1871, 2013.

[13] M. V Beena, "Automatic Sign Language Finger Spelling Using Convolution Neural Network : Analysis," vol. 117, no. 20, pp. 9–15, 2017.

[14] A. Dudhal, H. Mathkar, A. Jain, O. Kadam, and M. Shirole, *Approach for Indian Sign Language Recognition System Based on CNN*, vol. 2018. Springer International Publishing, 2018.

[15] R. F. P. Jr, C. D. B. Borges, M. A. Almeida, and C. P. Jr, "Static Hand Gesture Recognition Based on Convolutional Neural Networks," vol. 2019, 2019.

[16] Z. H. I. J. I. E. L. Iang, S. H. B. I. N. L. Iao, and B. I. N. G. Z. H. U, "3D Convolutional Neural Networks for Dynamic Sign Language Recognition," no. May, 2018.

[17] R. Yang, S. Sarkar, S. Member, and B. Loeding, "Handling Movement Epenthesis and Hand Segmentation Ambiguities in Continuous Sign Language Recognition Using Nested Dynamic Programming," vol. 32, no. 3, pp. 462–477, 2010.

[18] W. Yang, J. Tao, and Z. Ye, "PT US CR," *Pattern Recognit. Lett.*, 2016, doi: 10.1016/j.patrec.2016.03.030.

[19] O. Koller, H. Ney, and R. Bowden, "Deep Hand : How to Train a CNN on 1 Million Hand Images When Your Data Is Continuous and Weakly Labelled."

[20] G. Huang, S. Member, H. Zhou, X. Ding, and R. Zhang, "Extreme Learning Machine for Regression and Multiclass Classification," vol. 42, no. 2, pp. 513–529, 2012.

[21] E. K. Kumar, P. V. V Kishore, M. T. K. Kumar, and D. A. Kumar, "3D sign language recognition with joint distance and angular coded," *Neurocomputing*, no. xxxx, 2019, doi: 10.1016/j.neucom.2019.09.059.

[22] S. Suresh, T. P. Mithun Haridas, and M. H. Supriya, "Sign Language Recognition System Using Deep Neural Network," *2019 5th Int. Conf. Adv. Comput. Commun. Syst. ICACCS 2019*, pp. 614–618, 2019, doi: 10.1109/ICACCS.2019.8728411.

[23] R. Rastgoo, K. Kiani, and S. Escalera, "Hand sign language recognition using multi-view hand skeleton," *Expert Syst. Appl.*, vol. 150, p. 113336, 2020, doi: 10.1016/j.eswa.2020.113336.

[24] G. M. R. Neto, G. B. Junior, J. D. S. de Almeida, and A. C. de Paiva, "Sign Language Recognition Based on 3D Convolutional Neural Networks," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018, vol. 10882 LNCS, pp. 399–407, doi: 10.1007/978-3-319-93000-8_45.

[25] R. Cui, “Recurrent Convolutional Neural Networks for Continuous Sign Language Recognition by Staged Optimization,” pp. 7361–7369.

[26] K. M. Lim, A. Wee, C. Tan, C. Poo, L. Shing, and C. Tan, “Isolated sign language recognition using Convolutional Neural Network hand modelling and Hand Energy Image,” 2019.

[27] S. Masood, A. Srivastava, H. C. Thuwal, and M. Ahmad, *Real-Time Sign Language Gesture (Word) Recognition from Video Sequences Using CNN and RNN*. Springer Singapore.

[28] W. Tao, M. C. Leu, and Z. Yin, “Engineering Applications of Artificial Intelligence American Sign Language alphabet recognition using Convolutional Neural Networks with multiview augmentation and inference fusion,” *Eng. Appl. Artif. Intell.*, vol. 76, no. September, pp. 202–213, 2018, doi: 10.1016/j.engappai.2018.09.006.

[29] P. Nakjai and T. Katanyukul, “Hand Sign Recognition for Thai Finger Spelling : an Application of Convolution Neural Network,” 2018.

[30] K. Simonyan, “Two-Stream Convolutional Networks for Action Recognition in Videos,” pp. 1–9.