Development of A Computer Aided Real-Time Interpretation System for Indigenous Sign Language in Nigeria Using Convolutional Neural Network

Ayodele Olawale Olabanji, Akinlolu Adediran Ponnle

Abstract — Sign language is the primary method of communication adopted by deaf and hearing-impaired individuals. The indigenous sign language in Nigeria is one area receiving growing interest, with the major challenge faced is communication between signers and non-signers. Recent advancements in computer vision and deep learning neural networks (DLNN) have led to the exploration of necessary technological concepts towards tackling existing challenges. One area with extensive impact from the use of DLNN is the interpretation of hand signs. This study presents an interpretation system for the indigenous sign language in Nigeria. The methodology comprises three key phases: dataset creation, computer vision techniques, and deep learning model development. A multi-class Convolutional Neural Network (CNN) is designed to train and interpret the indigenous signs in Nigeria. The model is evaluated using a custom-built dataset of some selected indigenous words comprising of 15000 image samples. The experimental outcome shows excellent performance from the interpretation system, with accuracy attaining 95.67%.

Index Terms — Sign Language, Convolutional Neural Network, Interpretation System, Indigenous Sign Language.

I. INTRODUCTION

Sign language is the most natural and raw system of languages, which predate the early advent of human evolution. The evolution of sign languages follows the early theories in history, with appearance starting even before the emergence of spoken languages. Over time, sign language has been adopted and evolved into an integral part of the daily communication process. Currently, sign languages are extensively applied in international sign use for religious practices, in the sports world, and in workplaces [1].

Sign Language remains an ongoing research study since the arrival of computer vision techniques and has recorded notable progress in the use of several advanced techniques. This progress has seen sign language's identification move from being a not possible research area to one with a growing level of interest. The goal of the sign language interpretation system is to deliver a suitable platform that translates known sign into text and provide support to conversations between signers and non-signers. Besides, the interpretation system plays a critical role in human-computer interaction and graphical applications [2].

In recognition of sign language, the process of determining precise figures and shapes still faces some degree of uncertainty. This uncertainty can be attributed to several factors, including high clutter, image scale, occlusion, low lighting, and more. These factors hinder the possibility of attaining an accuracy level of 100%.

Indigenous sign language in Nigeria is the most popular communication approach for individuals with difficulties in speaking. This approach is used to express emotions and communicate efficiently. However, only a few non signers use or understand the indigenous sign language used in Nigeria. The alternative usage of written communication is considered inconvenient, as the deaf community is generally less skilled in writing the vocal language [3]. Hence, there is a need for the integration of an interpretation system into the communication methods.

In this paper, we present the development of a computer-aided interpretation system for indigenous sign language in Nigeria. The system created helps signers interpret indigenous signs of indigenous words towards communicating with non-signers. A new image dataset for some indigenous sign language for some indigenous words in Nigeria was created. The approach used takes in video sequences and extract multiple frames. The final dataset comprises image samples further processed using image processing algorithms comprising the Hue, Saturation and Value (HSV) thresholding; Histogram Back-projection; Otsu; Gaussian filter; and Median Filter. The dataset is used to train the Convolutional Neural Network (CNN) for feature extraction and classification purposes. The interpretation system was deployed using an interactive graphical user interface to engage users.

II. RELATED WORKS

Multiple research endeavors to create an optimal algorithm have led to the creation of a number of techniques. For example, some techniques developed for the interpretation of various known sign languages include Albanian sign language [4], American sign language [5], Arabic Sign Language [6], [7]; Chinese sign language [8], Korean sign language [9], Mexican sign language [10], and Tamil sign language [11].

A. A. Ponnle, Department of Electrical and Electronics Engineering, Federal University of Technology, Akure, Nigeria. (e-mail: ponnleakinlolu@yahoo.co.uk)
A new technique created by [12] entails YCbCr (Luminance; Chroma: Blue; Chroma: Red) conversion, edge detection, shape enhancement, and CbCr (Chroma: Blue; Chroma: Red) mapping. [13] identified a new technique that combines the TSL (Tint, Saturation, Luminance) skin color model and the improved Kalman filter. The technique applied limits any extra skin color objects through the improved Kalman filter model, which evaluates the motion and its midpoint location. The use of gloves by [14] employs random distributions of 10 full saturated colors in 20 patches. Nevertheless, the need for regular tuning would robustly enhance its operation [15], [16].

Fuzzy C-Means clustering (FCM) and thresholding technique was employed by [17] for segmenting the face and hand. The FCM uses a clustering method that applies an iterative approach to engage fuzzy partitioning. The Fuzzy C-Means algorithm first clusters the image using its color data. The mean value is finally selected by thresholding that uses the possible values available to the skin tone.

[18] applied a method involving two-step hand segmentation. The method applies canny edge detection on the images first to identify the edges. The second phase involves applying the seeded region growing technique to segment the hand from its background.

[19] use morphological operations and HSV (Hue, Saturation, Value) thresholding to segment the hand from its background. Determination of the HSV threshold values is done based on the skin color of the hand before applying it to the images. Morphological operations, including dilation and erosion, are further applied to eliminate noise. [20] used three-hand segmentation algorithms in the development of a recognition system. The three algorithms include the Otsu algorithm, thresholding in YCbCr color space, and the Gaussian Mixture Model (GMM).

[21] employed a heuristic optimization algorithm for the hyperparameter optimization process. The hyperparameters for the CNN structure used are based on the AlexNet model, alongside the outcome from the optimization algorithms. A 98.09% and 98.40% recognition accuracy were achieved on two public datasets used for evaluation. [22] used connected component analysis algorithm on the hand to segment the tip of the fingers as its Region of Interest (RoI). Gesture detection and recognition model using the segmented finger was developed via CNN, which achieved a 96.2% recognition accuracy.

Over the past years, several techniques have been developed for different sign languages used in different countries. For the indigenous sign language of indigenous words in Nigeria, which is possibly adopted and adapted from the ASL, the interpretation system presented is designed to solve the peculiarities concerning indigenous words in Nigeria.

### III. BACKGROUND STUDIES

Most recognition systems focus on three major techniques for hand sign analysis: appearance-based, depth-based, and sensor-based systems. The appearance-based analysis of hand signs combines two key stages:

1. Detection/Tracking
2. Recognition

#### A. Detection/Tracking

Detection plays a key part in the interpretation system. Selective processing and segmentation are required mainly on the raw input samples to aid feature extraction from random/uniform backgrounds. Tracking techniques involve reading the hand movement and its position [23]. Notable steps to the detection of the hands are:

1. Color Space Thresholding;
2. Binarization;
3. Background Removal.

#### B. Recognition

The primary task for an interpretation system is to identify hands-signs correctly for specific positions. Dynamic and static hand signs are two recognition techniques in the research of hand sign movement.

### IV. DATA PROCESSING

Fig. 1 shows the framework for the proposed interpretation system. The interpretation systems for sign language combine three key stages: Segmentation, Feature Extraction, and Classification. The segmentation process is characterized based on the capture system employed and focuses on obtaining the hand shape from input streams. The hand tracking and detection process uses the appearance-based method.
A. Data Collection

To create the database for the interpretation system, the selected indigenous signs and their connotations were obtained, with each containing several samples. The study employed the use of questionnaire, physical interaction, and indigenous sign illustrations [24]. The appearance-based technique was employed to acquire the needed data. This method was chosen because it is user-friendly and a popular option in sign recognition systems.

![Fig. 2. Samples from the created indigenous sign language word in Nigeria.](image)

RGB cameras were used to capture multiple image frames of the hand signs needed to create the dataset. The use of multiple samples per sign collected helps increase the accuracy of the system. The collected signs were adequate to create both the training, validation, and test dataset for the system. The image data and labels are saved into an SQLite database. The collected data served as direct inputs for the machine learning algorithms. The word dataset contains 15 words from the indigenous sign language in Nigeria. Fig. 2 shows some samples from the created dataset. The word dataset has the following features:

i. Each class has 1000 image samples from several signers and is saved in the JPG format.
ii. The total image samples contained in the dataset are 15000 (1000 * 15) images.

B. Preprocessing

From the raw hand-sign samples with normal backgrounds, the hand area is cropped out from the images. The approach applied focuses on the hand, the key element needed to help the interpretation system. The system converts the image samples in RGB (Red, Green, Blue) to HSV color space. The images are converted into binarized colour images using the histogram back-projection and OTSU approach [25] as the automatic thresholding technique. All images obtained are further re-sized by 50×50-pixel size.

C. Augmentation

The use of real-world data usually features the tendency to be random and offers several distortion types, including rotation, shifting, and more. Data augmentation remains a popular strategy [26] used to enhance the performance (with regards to accuracy) of a particular dataset. To enhance the dataset with such situations, image augmentation techniques are applied. The image samples are rotated randomly using 0 and 10 degrees, both in clockwise and anticlockwise directions. The augmentation applied to the system data shows promising results. Fig. 3 shows samples of the augmented images.

![Fig. 3. Augmented samples from the word dataset.](image)

D. Splitting Dataset

The word dataset contains 1000 exemplars for each of its 15 signs. The total number of images in the dataset is 15000. The word dataset is further split into training, validation, and test set. The three sets are split using the ratio of 80:10:10, respectively.

V. DEEP LEARNING MODEL

A deep neural network is engaged in this system to interpret indigenous hand signs. The neural network used is Convolutional Neural Network. The CNN model used, model training, evaluation, and other methodologies are discussed in this section.

A. Model Design

The system uses the created dataset as input to the CNN to classify the indigenous sign language. The CNN is able to perform feature extraction on the dataset. The CNN implementation and model training process were achieved using the Keras library. The CNN architecture comprises of the following layers: 2D Convolutional Layer, Max-pooling Layer, Dense Layer, and Output Layer. A small Region of Interest (RoI) was employed for the image samples to avoid disingenuous performances and overfitting in the learning process.

The CNN model comprises three convolutional layers, two pooling layers, two fully connected layers, combining the ReLU (Rectified Linear Unit) and SoftMax. In addition, the batch-normalization [27] learning approach is employed to avert overfitting. The first convolutional layer takes in the input shape using 8 kernels of size 1×1, followed by a batch normalization layer. Convolution is centered around merging two sets of information, and in the case of CNN, they are the Kernel function and input data. Considering a kernel K[i,j] with dimensions [2I + 1, 2J + 1] where (i,j) = (0,0) is the center of the kernel, the output g[m,n] applying the kernel function on the input a[m,n] is given in (1) and (2).
\[ g[m, n] = K[i, j] \times a[m, n] \quad (1) \]

\[ g[m, n] = \Sigma_{i=-1}^{1} \Sigma_{j=-1}^{1} K[i, j] \times a[m - i, n - j] \quad (2) \]

The second convolutional layers follow with 16 kernels of size \(1 \times 1\), followed by a batch normalization layer. A 2D maxpool layer having \((2,2)\) pool size is added. The maxpool layer decreases the spatial size of the subsequent feature map by accruing the generated features. The third convolutional layer has 32 kernels of size \(3 \times 3\) with an additional batch normalization layer. Another 2D maxpool layer having \((5,5)\) pool size and stride value of 5 is added.

Next, flatten layers and a fully connected dense layer are added to accept output elements. This fully connected layer uses the ReLU activation function. The ReLU function is well expressed by (3).

\[ \text{ReLU}(x) = \text{Max}(0, x) \quad (3) \]

The SoftMax classifier is the final layer in the classification block. The model used the ReLU activation function on every layer and the categorical cross-entropy cost function for loss optimization.

The use of ReLU does not initiate neurons having input values as negative. This approach makes the model more computationally efficient over tanh and sigmoid functions as it creates sparsity. Table 1 shows the layers, kernel size, and the number of neurons in the model.

### B. Model Optimization

The model optimization process is applied to make models more efficient and dependable to input data. The deep learning model that was created employed Stochastic Gradient Descent (SGD) as the optimizer for compiling the model. The SGD functions as a parameter in every training sample and delivers much faster performance. The SGD completes exclusive updates at a particular time.

The cross-entropy provides a better option for optimizing the cost function. For improved classification and prediction outcomes, the cross-entropy cost function is employed in the model.

| Layer Name   | Kernel Size | Number of Kernels /Neurons |
|--------------|-------------|----------------------------|
| Conv2D       | 1 \times 1  | 8                          |
| BatchNorm    | -           | -                          |
| Conv2D       | 1 \times 1  | 16                         |
| BatchNorm    | -           | -                          |
| MaxPool2D    | 2 \times 2  | -                          |
| Conv2D       | 3 \times 3  | 32                         |
| BatchNorm    | -           | -                          |
| MaxPool2D    | 5 \times 5  | -                          |
| Flatten      | -           | -                          |
| Dense        | -           | 128                        |
| Dropout (0.1)| -           | -                          |
| Dense        | -           | 26                         |

### VI. EXPERIMENT AND RESULT

The model is trained using the training and validation set representing 80% and 10% of the dataset distribution, respectively. The neural network completes classification tasks for the indigenous sign language in Nigeria by learning key features for individual classes in the dataset. The classification model attained excellent performance with no transfer learning. The downside to this technique is that if similar data exist for some classes in the subsequent classification task, the possibilities of data leakage will cause biases transfer learning.

In tackling this issue, the model was initialized via the He initialization for classification tasks. The model was trained for 30 epochs, with experimentation performed on multiple hyperparameters, including its regularization, and learning rate. The best values found for the initial learning rate are 0.0001. The model also employs a schedule learning rate which uses step decay to avert divergence and enhance the loss curve. The step decay takes in a learning rate and decreases by a factor of 0.1 after every 5th epoch.

Accuracy, confusion matrix, Recall, Precision, and F1 score were used in evaluating the model performance. Fig. 5 shows the plot of training accuracy against the validation accuracy. Fig. 6 shows the plot of training loss against the validation loss. Both plots were achieved using 30 epochs.

C. Model Training

In training the model, the He normal initialization [28] was applied to initialize every kernel towards achieving quicker convergence. The batch-normalization technique was applied to limit the convolution layers from learning less relevant features and avoid overfitting to the features.

Batch normalization (BN) is a technique to normalize activations amongst connecting layers of deep neural networks to enhance accuracy and training [29]. The addition of the batch-normalization layer aids in the reduction of co-variance shifts within the network’s hidden kernels. The outcome from the previous layer is normalized via BN engaging the present batch statistics. This approach makes the weight distribution of the hidden kernel units or adapts to any data distribution variation. The BN layer also provides a level of regularization effect as weights are scaled depending on the batch mean and variance, which gives the weights stability. Fig. 4 shows the model training flowchart.
Accuracy is calculated using the total sample (1500) minus the TP, FP and FN. Table III shows the TP, FN, FP, and TN values obtained for Niger, Calabar, Enugu, Imo, Bayelsa, Bauchi, and Igbo.

| Word  | True Positive (TP) | False Positive (FP) | False Negative (FN) | True Negative (TN) |
|-------|-------------------|---------------------|---------------------|-------------------|
| Niger | 100               | 3                   | 0                   | 1397              |
| Calabar | 100             | 0                   | 0                   | 1400              |
| Enugu | 100               | 0                   | 0                   | 1400              |
| Imo   | 100               | 0                   | 0                   | 1400              |
| Bayelsa | 100             | 0                   | 0                   | 1400              |
| Bauchi| 100               | 0                   | 0                   | 1400              |
| Igbo  | 100               | 2                   | 0                   | 1398              |

ii. For Ondo, Borno, Abia, Ekiti, Kastina, Kaduna, Anambra, and Are, the outcome for these classes shows varying True Positive (TP), False Positive (FP), and False Negative (FN). Ondo has the lowest TP value with 68. The highest misclass value of 31 was recorded between “Ondo” and “Are.” The values for the TP, FP, TN, and FN for class 1, 6, 7, 8, and 9 are shown in Table IV.

Based on the above observation, the following performance metrics were computed for the following class: Ondo:

**Precision (P) = TP / (TP+FP) = 68 / (68+4) = 0.94**

**Recall (R) = TP / (TP+FN) = 68 / (68+32) = 0.68**
**F1-score** = 2 × \( \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \) = 2 × \( \frac{0.94 \times 0.68}{0.94 + 0.68} \) = 0.79

Niger:

\[
\text{Precision (P)} = \frac{\text{TP}}{(\text{TP} + \text{FP})} = \frac{100}{100 + 3} = 0.97
\]

\[
\text{Recall (R)} = \frac{\text{TP}}{(\text{TP} + \text{FN})} = \frac{100}{100 + 0} = 1.00
\]

\[
\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = 2 \times \frac{0.97 \times 1.00}{0.97 + 1.00} = 0.99
\]

The Precision, Recall, F1-score, and Support for all Word is shown in Table V.

| Word  | Precision | Recall | F1-score | Support |
|-------|-----------|--------|----------|---------|
| Ondo  | 0.94      | 0.68   | 0.79     | 100     |
| Niger | 0.97      | 1.00   | 0.99     | 100     |
| Calabar | 1.00     | 1.00   | 1.00     | 100     |
| Borno  | 1.00      | 0.82   | 0.90     | 100     |
| Enugu  | 1.00      | 1.00   | 1.00     | 100     |
| Abia   | 0.93      | 0.99   | 0.96     | 100     |
| Ekiti  | 0.98      | 0.98   | 0.98     | 100     |
| Imo    | 1.00      | 1.00   | 1.00     | 100     |
| Kastina | 0.83     | 0.99   | 0.90     | 100     |
| Kaduna | 1.00      | 0.97   | 0.98     | 100     |
| Bayelsa | 1.00     | 1.00   | 1.00     | 100     |
| Anambra | 1.00     | 0.98   | 0.99     | 100     |
| Bauchi | 1.00      | 1.00   | 1.00     | 100     |
| Are    | 0.74      | 0.90   | 0.81     | 100     |
| Igbo   | 0.98      | 1.00   | 0.99     | 100     |
| Micro Average | 0.95    | 0.95 | 0.95 | 1500 |
| Macro Average  | 0.96    | 0.95 | 0.95 | 1500 |
| Weighted Average | 0.96    | 0.95 | 0.95 | 1500 |

The micro-averaged score is calculated by considering the total TP, total FN, and total FP of the model. The values of TP, FN, and FP for every class are added to evaluate the micro average score for Precision, Recall, and the F1 score.

\[ \text{Total TP} = (100 + 100 + 100 + 100 + 100 + 100 + 100 + 90 + 98 + 97 + 99 + 98 + 99 + 82 + 68) = 1431 \]

\[ \text{Total FP} = (3 + 2 + 4 + 7 + 2 + 20 + 31) = 69 \]

\[ \text{Total FN} = (32 + 18 + 1 + 2 + 1 + 3 + 2 + 10) = 69 \]

The micro-averaged score for Precision, recall, and F1-score is calculated as:

\[
\text{Micro Precision} = \frac{\text{Total TP}}{(\text{Total TP} + \text{FP})} = \frac{1431}{1431+69} = 0.95
\]

\[
\text{Micro Recall} = \frac{\text{Total TP}}{(\text{Total TP} + \text{FN})} = \frac{1431}{1431+69} = 0.95
\]

\[
\text{Micro F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = 2 \times \frac{0.95 \times 0.95}{0.95+0.95} = 0.95
\]

The macro averaged score for the word model is evaluated to gives \text{Macro Precision} = 0.96, \text{Macro Recall} = 0.95, and \text{Macro F1-score} = 0.95. The weighted-average score evaluated using the weighted mean of the measures gives the \text{weighted – precision} = 0.96, \text{weighted – Recall} = 0.95, and \text{weighted F1 – score} = 0.95.

Based on the performance metric values, the classification model for the word dataset has a precision of 0.96 (on a scale of 0 – 1), Recall of 0.95 (on a scale of 0 – 1), and an F1-score of 0.95 (on a scale of 0 – 1).

The F1 score was used because it emphasizes the false negatives and false positives in its evaluation process. Table 6 shows the average Precision, Recall, and F1 score of the developed model.

| Model | Precision | Recall | F1-score |
|-------|-----------|--------|----------|
| Word Model | 0.95 | 0.95 | 0.95 |

These experiments show the model considerably performed well with the quality input data and training process. The outcomes also validate the robustness and Precision in interpreting indigenous sign language in Nigeria using the proposed model.

### VII. CONCLUSION

This paper presents an efficient real-time-based interpretation system for sign language of some indigenous words in Nigeria. A dataset containing word samples from the indigenous sign language in Nigeria was created to cater for training the model. This paper considered hand movement as the key feature, creating static image samples from continuous video stream subjected to image processing techniques. The model developed uses the convolutional neural network as the classifier. For the interpretation of signs, appearance-based and digital interpretation, the CNN proves to be a better choice. The proposed model recognizes and interprets individual signs as text, supporting the removal of any communication gap amongst signers and non-signers. The method developed has given excellent interpretation accuracies attaining a 95.67% training accuracy and 95.28% validation accuracy. The proposed interpretation system using the CNN classification approach achieves 0.95 of Precision, 0.95 of Recall, and 0.95 of F1-score (on a 0–1 scale). Towards improving the interpretation system, the next step is to gather more data for Nigeria’s indigenous sign language. In consideration of approach and practicality, more focus will go into interpreting and recognition of idioms, phrases, and sentences available in the indigenous signs of Nigeria.

### ACKNOWLEDGMENT

We acknowledge with deep gratitude the special class of the Redeemed Christian Church of God (RCCG) Kings Sanctuary, Akure, Nigeria, and the staff of the School for the Deaf, Akure, Nigeria, for their valuable assistance for this work.
REFERENCES

[1] J.G. Kyle, J. Kyle, B. Wolf, G. Pullen, and F. Maddix. Sign language: The study of deaf people and their language. Cambridge university press, 1988.

[2] M. Billinghamurst. "Put that where? Voice and gesture at the graphics interface." ACM Siggraph Computer Graphics, vol. 32, no. 4, pp. 60-63, 1998.

[3] L. Pigou, S. Dieleman, P.J. Kindermans, and B. Schrauwen, B. Sign language recognition using convolutional neural networks. In European Conference on Computer Vision, Vol.8925, pp. 572-578. Zurich: Switzerland, 2014

[4] E. Gani, and A. Kika. “Albanian Sign Language (AlbSL) Number Recognition from Both Hand’s Gestures Acquired by Kinect Sensors,” International Journal of Advanced Computer Science and Applications (IJACSA), vol. 7, no. 7, pp. 216-220, August 2016.

[5] A. Thalange, and S. Dixit. “Sign Language Alphabets Recognition Using Wavelet Transform.” In International Conference on Intelligent Computing, Electronics Systems, and Information Technology (ICECSE-15), pp. 25-26, Kuala Lumpur, 2015.

[6] K. Assaleh, and M. Al-Rousan. Recognition of Arabic Sign Language Alphabet Using Polynomial Classifiers. EURASIP Journal on Applied Signal Processing, vol. 13, pp. 2136-2145, August 2005.

[7] K. Assaleh, T. Shanableh, M. Fanaswala, H. Bajaj, and F. Amin. “Vision-based system for Continuous Arabic Sign Language Recognition in user dependent mode,” In 2008 5th International Symposium on Mechatronics and its Applications (ISMA08), pp. 1-5, Amman, 2008.

[8] C. Wang, W. Gao, and Z. Xuan, “A real-time large vocabulary continuous recognition system for Chinese Sign Language.” In Pacific-Rim Conference on Multimedia. pp. 150-157, Springer, Berlin, Heidelberg, 2001.

[9] J. S. Kim, W. Jang, and Z. Bien, “A dynamic gesture recognition system for the Korean Sign Language KSL.” IEEE Transactions on Systems, Man and Cybernetics, Part B ( Cybernetics), vol. 26, no. 2, pp. 354-359, April 1996.

[10] F. Solis, D. Martínez, and O. Espinozà, “Automatic Mexican Sign Language Recognition Using Normalized Moments and Artificial Neural Networks.” Engineering. vol. 8, no. 10, pp. 733-740, 2016.

[11] P.S. Rajam, and G. Balakrishnan, G. “Recognition of Tamil sign language alphabet using image processing to aid deaf-dumb people. Procedia Engineering, vol. 30, pp. 861-868, January 2012.

[12] A.Y. Dawod, J. Abdullah, and M.J. Alam, “Adaptive skin color model for hand segmentation.” In 2010 International Conference on Computer Applications and Industrial Electronics, pp. 486-489, Kuala Lumpur, 2010.

[13] S. Mo, S. Cheng, and X. Xing, “Hand gesture segmentation based on improved kalman filter and TSL skin color model.” In 2011 International Conference on Multimedia Technology, pp. 3543-3546, Hangzhou, 2011.

[14] R. Y. Wang, and J. Popović, “Real-time hand-tracking with a color glove.” ACM Transactions on Graphics (TOG), vol. 28, no. 63, pp. 1-8, July 2009.

[15] H. Lu, K.N. Plataniotis, and A.N. Venetsanopoulos, “A full-body layered deformable model for automatic model-based gait recognition.” EURASIP Journal on Advances in Signal Processing, vol. 2008, pp. 1-13, 2007.

[16] T.B. Moeslund, A. Hilton, and V. Kruger, “A survey of advances in vision-based human motion capture and analysis.” Computer Vision and Image Understanding, vol. 104, no. 2-3, pp. 90-126, 2006.

[17] B. Büyüşkaraç, M.M. Bulut, and G.B. Akar, “Sign language recognition by image analysis.” In 2016 24th Signal Processing and Communication Application Conference (SIU), pp. 417-420, Zonguldak, 2016.

[18] C.M. Jin, Z. Omar, and M.H. Jaward, “A mobile application of American sign language translation via image processing algorithms.” In 2016 IEEE Region 10 Symposium (TENSI SyM), pp. 104-109, Bali, 2016.

[19] M. A. Uddin, and S.A. Chowdhury, “Hand sign language recognition for bangla alphabet using support vector machine.” In 2016 International Conference on Innovations in Science, Engineering and Technology (ICISET), pp. 1-4, Dhaka, 2016.

[20] S.C. Agrawal, A.S. Jalal, and C. Bhattacharjya, “Recognition of Indian Sign Language using feature fusion.” In 2012 4th International Conference on Intelligent Human Computer Interaction (IHCI), pp. 1-5, Kharagpur, 2012.

[21] P. S. Neethu, R. Saguna, and D. Sathish, “An efficient method for human hand gesture detection and recognition using deep learning convolutional neural networks.” Soft Computing, vol. 24, pp. 1-10, 2020.

[22] T. Ozcan, and A. Basturk, “Transfer learning-based convolutional neural networks with heuristic optimization for hand gesture recognition.” Neural Computing and Applications, Vol. 31, No. 12, pp. 8955-8970, 2019.

[23] N. Ç, Kılıçoğlu, and U. Güldükbay, ”A hand gesture recognition technique for human–computer interaction,” Journal of Visual Communication and Image Representation, vol. 28, pp. 97–104, April 2015.

[24] K.U. Ebi. Talking Hands: An Introduction to Sign Language in Special Education. Adex Sea Concept, Nigeria. ISBN: 978-978-52439-7-0, 2019.

[25] N. Otou, “A threshold selection method from gray-level histograms.” IEEE transactions on systems, man, and cybernetics, vol. 9, no. 1 pp. 62-66, 1979.

[26] J. Wang, and L. Perez, “The effectiveness of data augmentation in image classification using deep learning.” Convolutional Neural Networks Vis. Recognit, vol. 11, December 2017.

[27] S. Ioffe, and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift.” In Proceedings of the 32nd International conference on machine learning, PMLR, vol. 37, pp. 448-456, 2015.

[28] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification.” In Proceedings of the IEEE international conference on computer vision, pp. 1026-1034, 2015.

[29] J. Bjorck, C. Gomes, B. Selman, and K.Q. Weinberger, “Understanding batch normalization.” arXiv preprint arXiv:1806.02375, June 2018.