Real-Time Sign Language Fingerspelling Recognition using Convolutional Neural Network

Abiodun Oguntimilehin
Department of Computer Science, Afe Babalola University, Nigeria
ebenabiodun2@yahoo.com

Kolade Balogun
Department of Computer Science, Afe Babalola University, Nigeria
alibalogun996@gmail.com

Abstract: *Sign language allows mute people to communicate, problem occurs when a conversationalist fails to understand it. Despite efforts to address this problem, an effective solution is not yet found. In this work, Convolutional Neural Network (CNN) was trained on two different datasets separately- binary and Red Blue Green (RGB), each contains 25,900 images of Nigerian Sign Language. A deep neural pre-trained module was used to detect hand gestures in the video feed which tackled the issue of complex backgrounds, also showed excellent detection in dimly lit areas. The accuracies of (98.95%, 76%) and (98.87%, 98.85%) were obtained respectively on the training and the validation sets. The real time system developed implemented both models as a single system which makes it a unique one.*

Keywords: *Communication, deep learning, sign language, hand gesture recognition.*

Received July 6, 2023; accepted December 20, 2023
https://doi.org/10.34028/iajit/21/1/14

1. Introduction

Communication is an important aspect of human existence that defines interaction between people [27], it makes interaction among people possible [1]. Sign language is a communication language; it is a nonverbal method of communication using bodily movement mostly. It commonly involves the use of hand and harm when verbal communication is not possible or inappropriate [26]. Sign language varies by places with many varieties around the world [5, 17, 23, 27]. Sign language is considered a visual and natural language which allows interaction and communication between deaf and dumb and the non-mute people [20, 25].

Due to the fact that non-mute people are less likely to learn Sign language, mute people find it difficult to communicate with them [16]. The problem extends to mutes from different communities as sign language varies from place to place. This is where pattern recognition algorithms play an important role, as it is used in automating the process of recognizing signs to help non-signers comprehend and interact with mute people. Pattern recognition is an artificial intelligence area that entails feeding a piece of data, such as an image into a prediction model that returns information about existing forms, such as an object, a number, or a letter and so on [6]. Human hands were recognised to be one of the earliest means of communication [28], hand gesture recognition constitutes natural intelligent and more so convenient method valuable in the interaction of humans with computers, the advantage of which is taken in developing sign language recognition systems [13, 17, 30].

The hand signs collected as photos or in real time are used as the input data in sign language pattern recognition [6]. Technological issues and interdisciplinary inclusiveness are among the problems associated with development of sign language translation systems. Artificial intelligence has played a prominent role in achieving automated sign language translation systems [7]. Frantic efforts must be made to enable more people understand sign language [19].

Despite many research efforts in producing promising systems for sign language recognition, it is quite worrisome that some challenges are not yet resolved. These include how to interpret combination of subtle movement and expressions like hand pose and movement, facial expressions, and body posture among others. Also, the language has at least thousands of words, including words with very similar hand poses and this constitutes a significant problem as current gesture recognition systems mostly cover minor subset of gestures that are defined appropriately. In addition to this, the degree of contact between signer's fingers is mostly used to differentiate some of the signs. Varied camera angles and signer's postures give room for a lot of variations [11].

This work focuses static fingerspelling in Nigerian Sign Language (NSL) which is coined from the American Sign Language (ASL). It consists of a dataset of sign language which includes the signing of the letters A-Y, numbers 1-9 excluding the letters J and Z because they are dynamically signed, number 0 because it resembles the signing for the letter O. With recent improvement of Graphical Processing Unit (GPU), Convolutional Neural Network (CNN) has been employed in solving several computer vision problems. Therefore, this work takes the advantage of convolutional neural networks to achieve a real-time and commendable sign language recognition system.

2. Review of Related Literature

This section describes the literature associated with gesture recognition as well as image processing. It discusses the methods used by some authors to accomplish the related systems.

The system developed in [3] used Approximate String Matches (ASM) and K-Means classifiers in the work. K-Means was used to encode trajectory of hand joints to make character sequence. Analysis of character sequence was carried out with the aid of ASM. The authors reported the system recorded 93.52% accuracy.

Histogram of Oriented Gradient (HOG) was used in [14] to resolve the problem associated with complexity of hand signs as well as changing background in some systems developed to manage real-time interpretation of sign language. HOG and Support Vector Machine (SVM) were combined for the detection of hand movement and its direction. The system got 91.7% detection when tried on six standard motions.

Liwicki and Everingham in [15] developed a system that can recognize finger spelled words from a video, considering British Sign Language (BSL). Histogram Of Gradients (HOG) was used to detect hand, employing the variation in colour of the hand and the non-hand pixels. Hand shapes were categorized with the use of Hidden Markov Model (HMM). Palacios *et al*. [18] developed a hand sign recognition system utilizing RGB-D sensors. This system took the advantage of depth data to resolve the issues associated with lightning situations as well as backgrounds that are clustered. Recognition of gesture was carried out using Feature Based Decision Tree. The detection accuracy of 97% was recorded.

In the study carried out in [31], Bayesian Inference Rule and K-Nearest Neighbour (KNN) were engaged in the work. KNN was used for the estimation of the two geometric invariants and the bayes decision tree was employed for the recognition. This method achieved 95% classification accuracy. Chourasia *et al*. [9] presented a motion recognition system to recognize Indian sign language. KNN and SVM classifiers were employed for sign letter classification while extraction of hand features and detection of handshape were executed by combining Hu invariant with moments and speed-up robust features. Classification accuracy of 96% was recorded.

A sign language recognition system based on skin colour was developed in [27] using CNN. Performance of this system is subject to a background with uniformity and correct lightening. Accuracies of 90.04%, 93.44% and 97.52% were recorded for alphabetic recognition, number recognition and static word recognition respectively, giving an average of 93.67% detection rate. The system developed in [21] used a vision-based approach which avoids the use of artificial devices during interaction. CNN was used for gesture classification, classifying full ASL alphabets. The system reported 92% accuracy.

A system to recognize hand gesture and translate it into speech was developed in [24]. CNN was used to translate 26 alphabets, 11 phrases and 10 numbers in ASL using 47 folders, each containing 2400 greyscale images of each gesture captured using varied positioning. Coloured hand glove was used in the work and hand segmentations depend majorly on the colour of the hand.

This work differs from these earlier works in a few ways. First, being the first NSL fingerspelling recognition system to classify a total of 37 alphabets and numbers compared with the approaches that were able to classify only 24 classes reported in the literature. Secondly, the work incorporated artificial signs to represent punctuation and symbols to improve usability. CNN was used in this work.

3. Problem Statement

The deaf-dumb community's main social problems are communication barriers between the hearing majority and the deaf-dumb minority, which prohibit them from obtaining basic and needed life services. Usually, people do not learn sign language if there is no mute person in their relation circles or if it is not required for their job. When they engage with a mute person, the communication can be hard and tedious. Despite the fact that the problem has been addressed by advances in automatic sign language detection, an effective solution is yet to be found due to a number of difficult issues. The majority of existing research focuses on developing vision-based hand recognition systems implemented using high-resolution cameras and depth sensors, as well as segmentation algorithms that are limited to basic backgrounds. However, when working with a vast sign vocabulary collected in complex and uncontrolled background settings, the efficiency of such methods is severely constrained.

4. Basic Concept of CNNs

CNNs are neural networks that replace one of its fully connected layers with convolutional operation [12]. It comprises convolutional filters, pooling layers as well as activation blocks [29]. It is capable of working on input images to differentiate between one image and the other. It is a layered network consisting of a convolutional layer for feature extraction, layered upon a pooling layer for compression, connected to a fully connected network of artificial neurons for recognition

[2]. Figure 1 better explains the workings of CNN, shows the basic architecture of CNN.

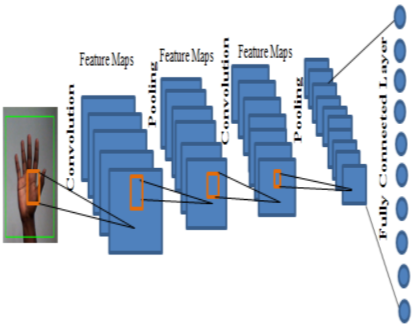


Figure 1. CNN Architecture.

4.1. Input Layer

The major building block of a computer image is the combination of its pixels. After capturing image, the pixels which range from 0-255 form a matrix like object of the digital image. Layers in CNN are automated to process and detect simpler patterns like lines and curves before moving to the identification of complex ones like faces [8].

4.2. Convolutional Layer

This layer comes after the input layer; it is made up of convolutional kernels. Every neuron is associated with a kernel window in the convolution layer. The convolution kernel is determined from the weight of each linked neuron. A series of N pictures, one designated for each of the N neuron is the result of this convolution procedure. The new images may have negative values due to convolution. In order to avoid this, a Rectified Linear Unit (ReLU) is used to substitute negative values with zero. This layer's outputs are christened feature maps [2, 4, 8, 22]. Stride, filter size and zero padding or padding specify each convolutional operation of the convolutional layer. Stride is a non-negative whole number value which determines the sliding step. Filter size is important; it is fixed across all filters that are used in the convolutional operation. Zero rows or zero columns are added to the original image matrix in order to get the required output's feature map size. Zero padding strips the network from becoming smaller and gives room for voluminous deep layers in the network [2]. Example of zero padding is shown in Figure 2.

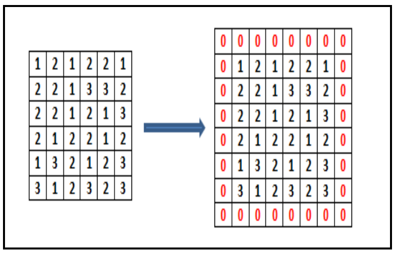


Figure 2. Padding of a 2D image array.

4.3. Pooling Layer

After a convolution layer, comes the pooling layer. Pooling reduces the size of feature maps which subsequently reduces the time used to train the network. Max pooling is often employed in extracting sub-regions of the feature map like 2x2 pixel tiles [2, 22].

4.4. Fully Connected Layer

This is made up of a limited number of neurons that receives a vector as input and outputs another vector. This layer is added to ensure an inexpensive learning of nonlinear combinations of high-level information of the output of the convolutional layer. It ensures the input image becomes flattened to a column vector after it has been transformed into a format for multi-level perceptron. Backpropagation is used by every round of training to send the flattened output to feed forward neural network. Low level features and dominating ones in images over some epochs can be distinguished using this model. Softmax classification approach can be employed for the classification. The output of the fully connected layer is the final output of the CNN [2, 22].

5. Proposed Method

5.1. Collection and Description of Dataset

Normal webcam was used to collect two different datasets used in this research. The first data set consists of 25,900 images of 6 different signers that signed 37 static gestures, which were then converted to binary images to train the model by using binary inverse threshold and Ostu binarization techniques. Since (2/V) and (6/W) are differentiated based on minute differences, there were some difficulties encountered when differentiating them after they were converted to binary images. The second set consists of the same 25,900 images, but left in Red Blue Green (RGB) format. To collect more balanced datasets, images were captured from different viewpoints. The data sets were collected while subjects were moving their hands around both on image plane and along z-axis.

5.2. Hand Segmentation

The hand images were captured by first detecting the hand from the webcam video feed and then segmenting it by cropping it out. Hand detection was achieved using a Google model called Mediapipe. Mediapipe is a Machine Learning (ML) solution developed by Google; it recognizes hands using two sub models namely, palm detection and hand landmark model:

1. Palm Detection Model: the palm detection model was designed using a Single Shot Detector (SSD) which is the fastest object detection Machine

Learning (ML) model currently. This model was trained using a data set of different hands of different pixels. The palm detector was used to detect the location of the hand.

2. Hand Landmark Detection Model: this follows palm detection model. It does exact key point localization of 21 3D hand-knuckle coordinates, making use of regression. This model works effectively for hands that are partially visible as well as self-occlusion [10].

Using this method for segmentation addresses the problems in detecting hand gestures from a complex background faced by some of the existing researches reviewed.

5.3. Data Pre-processing

The data pre-processing module has sub modules and this is where the data was processed before the modelling phase. The sub modules in the pre-processing phase are:

5.3.1. Resizing the Image

When these images come, they might be irregular in shapes, so here the images were resized to maintain perfect shapes throughout the data set. All the images were resized to similar shapes so that no image is bigger than the other. By re-sizing the pixels of input image, processing time and dimensions of all images were made equal. The images of the first dataset were resized to 64x64 dimensions being binary, while the images of the second data set were resized to 128x128 dimensions. Each vector was reshaped into an array of selected size using the NumPy python package.

5.3.2. Normalizing the Image

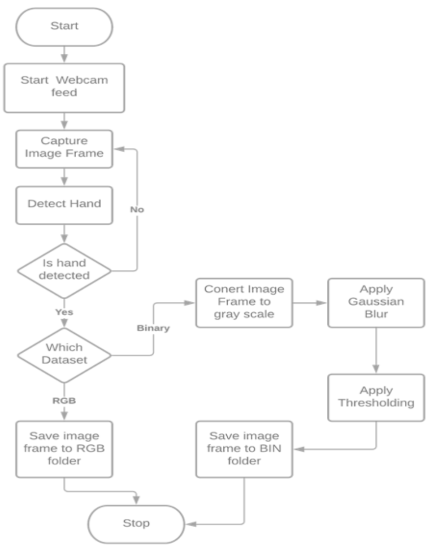
The resized images were then normalized to get values ranging from 0 to 1. This was done so that data was divided by 255 to get a value that spans between 0 and 1, nothing lesser and nothing higher in order to balance the weights of the data for further training.

5.4. Data Augmentation

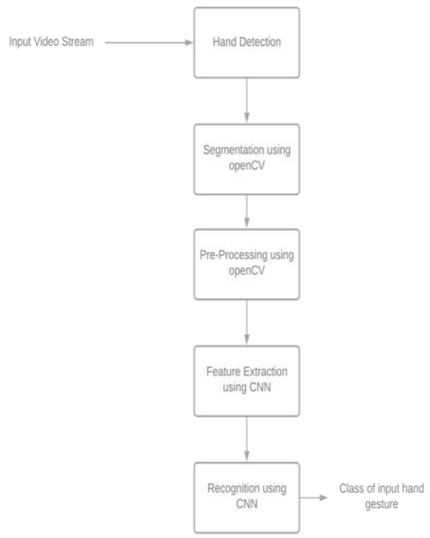
Data augmentation was performed on the image to increase the number of images in the dataset. There are many techniques used in data augmentation, with Shear image transformation, flipping and zooming used in this work. It enables better learning due to the bigger amount of the training set and allows the algorithm to learn from different conditions of the object in question.

5.5. Modelling

The model was created using an input layer1(convolution layer) along with rectifier linear unit as the activation function for the artificial neurons and max pooling layers for feature extraction, a flatten layer, and a final fully connected output layer saddles with responsibility to classify gestures. Input layer takes the threshold image or RGB image of hand postures to further layers for feature extraction and classification. The two datasets, each containing 25,900 images were each divided into 18,130 training set and 7,770 validation set. Each image was assigned a class label from alphabets, numbers and symbols. The model was trained over 6 epochs with batch size of one. System's modelling process is shown in Figure 3.



a) Data capturing process.



b) Training and Recognition Process.

Figure 3. System's modelling process.

6. Results, Discussion and Development of the Real Time Sign Language System

6.1. Results

This section describes the results of the experiment carried out using the CNN configuration. The experimental result shows that the model which was trained on the coloured image data set achieved 98.87% accuracy on its training set and 98.85% accuracy on its validation which is better compared with the model trained on the threshold (Black and White) images data which achieved 98.95% on its training set, but with a lower result of 76% accuracy on its validation set. This was to be expected as some signed letters cannot be told apart by just the shapes alone. The results of the two different datasets are shown in Figures 4 and 5 respectively.

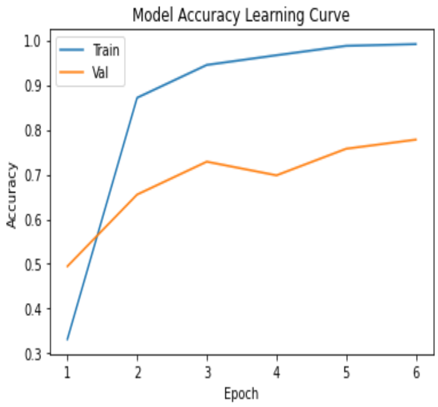


Figure 4. Accuracy curve for thresh (Black and White) model.

Comparing the two graphs in Figures 4 and 5 accuracies of the Thresh (black and white) and RGB models, it was observed that the accuracy value of the RGB model started at the 77.5% range for the first epoch as opposed to the Thresh model (black and white) which achieved 33% at its first epoch. After subsequent epochs, both models started showing steady improvement in accuracies on their training sets. Although the RGB model had a higher starting accuracy on its training set which achieved 98.87% on its last epoch contrary to the thresh (black and white) model which started with a low accuracy of 33% at its first epoch, but subsequently became higher than the RGB model with an accuracy of 98.95%.

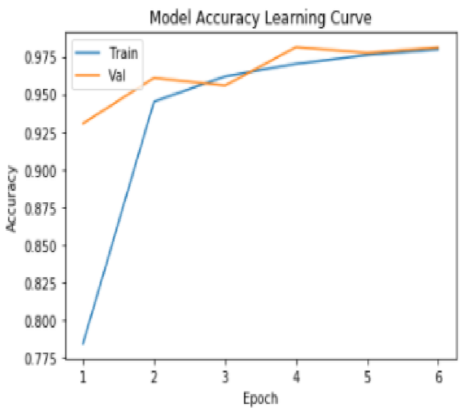


Figure 5. Accuracy curve of RGB model.

The RGB model had a higher accuracy with respect to its validation set at 93% at the starting epoch when compared with the thresh model which achieved 50% accuracy at the starting epoch. After subsequent epochs, the improvement of the RGB model was observed as it achieved a validation accuracy of 98.85% as opposed to the thresh model which did not achieve any NOP improvement in its validation accuracy as it ended with 76% accuracy. This can also be as a result of the difference in diversity and the lack of information in the training set as the Thresh data set contains only silhouettes of the hands as compared with the RGB data set that contains all hand features.

The thresh model showed good results when used on the real system, but was affected by changes in the lightening. The RGB model showed very good result in varying background and light situations but had problems correctly classifying gestures made from hands lighter than the ones it was trained on, concluding that both models were successful in classifying hand signs with different advantages. That is why our proposed solution takes advantages of both models to create a more robust system. The developed system implemented both models as a single one.

6.2. Development of the Real Time Sign Language System

The front end of the application was developed using a python package called PysimpleGUI which is an open source package used for the development of Graphical User Interfaces (GUIs), coding was done using python programming.

The main page of the system shown in Figure 6 contains a centre screen where the webcam feed is shown; an exit button to close the application also exists here. It contains a start button to start the video feed and a text box to show the output of the recognized gesture. It also has a drop-down menu for the selection of the model (BIN/RGB) to be used; this is shown in Figure 7.

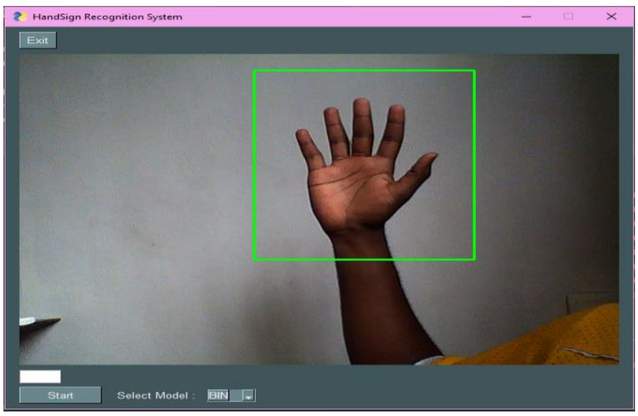


Figure 6. Main screen.

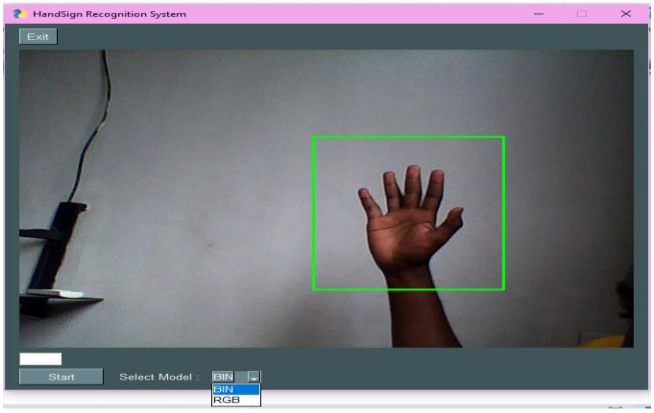


Figure 7. Drop down menu options.

The main page of the system shown in Figure 6 contains a centre screen where the webcam feed is shown; an exit button to close the application also exists here. It contains a start button to start the video feed and a text box to show the output of the recognized gesture. It also has a drop-down menu for the selection of the model (BIN/RGB) to be used; this is shown in Figure 7.

7. Conclusions

This work explores the opportunities and challenges in the recognition of hand gestures by the creation of not one but two data sets: black and white (binary) and coloured RGB images. The two different datasets were made use of to train a CNN model of the same architecture differently, generating two different models. This paves way for the development of a better robust detection system, although the system can recognize gestures successfully, some improvements are still possible, such as increasing the overall speed of the system and to further improve the system's dynamic signed words and sentences.

Acknowledgements

The authors are grateful to the founder of Afe Babalola University, Ado-Ekiti, Nigeria for the financial support towards the publication of this work.

References

[1] Akansha T. and Bansal S., "Hybrid FiST_CNN Approach for Feature Extraction for Vision-Based Indian Sign Language Recognition," *The International Arab Journal of Information Technology*, vol. 19, no. 3, pp. 403-411, 2022. https://doi.org/10.34028/iajit/19/3/15

[2] Albawi S., Bayat O., Al-Azawi S., and Ucan O., "Social Touch Gesture Recognition Using Convolutional Neural Network," *Computational Intelligence and Neuroscience*, pp. 1-10, 2018, https://doi.org/10.1155/2018/6973103

[3] Alonso D., Teyseyre A., Soria A., and Berdun L., "Hand Gesture Recognition in Real World Scenarios Using Approximate String Matching," *Multimedia Tools and Applications*, vol. 79, pp. 20773-20794, 2020. https://doi.org/10.1007/s11042-020-08913-7

[4] Alzubaidi L., Zhang J., Humaidi A., Al-Dujaili A., and Duan Y., "Review of Deep Learning: Concepts, CNN Architectures, Challenges, Applications, Future Directions," *Journal of Big Data*, Springers, vol. 8, no. 53, 2021. https://doi.org/10.1186/s40537-021-00444-8

[5] Amal H., Reny R., and Prathap B., "Hand Kinesis in Indian Sign Language Using NLP Techniques with SVM Based Polarity," *International Journal of Engineering and Advanced Technology*, vol. 9, no. 4, pp. 2044-2050, 2020. https//doi: 10.35940/ijeat. D8483.049420

[6] Ayachi R., Said Y., and Atri M., Study and Implementation of Deep Learning Algorithms for Object Detection, M.Sc Physics Thesis, University of Monastir, 2018.

[7] Baumgartner L., Jauss S., Maucher J., and Zimmermann G., "Automated Sign Language Translation: The Role of Artificial Intelligence Now and in Future," *in Proceedings of the 4th International Conference on Computer-Human Interaction Research and Applications*, Budapest, pp. 170-177, 2020. https://www.scitepress.org/PublishedPapers/2020/101438/101438.pdf

[8] Bhatt L., Patel C., Taisania H., Patel J., and Vashela R., "CNN Variants for Computer Vision: History, Architecture, Application, Challenges and Future Scope," *Electronics*, vol. 10, no. 20, pp. 1-28, 2021. https://doi.org/10.3390/electronics10202470

[9] Chourasia N., Kanchan D., and Supratim S., "Analysis on Hand Gesture Spotting Using Sign Language Through Computer Interfacing," *International Journal of Engineering Science and Innovative Technology*, vol. 3, no. 3, pp. 132-139, 2014. Doi: https://doi.org/10.1007/s11263-018-1121-3

[10] Google, Mediapipe, https://mediapipe.dev, Last Visited, 2023.

[11] Kang B., Tripathi S., and Nguyen T., "Real-Time Sign Language Fingerspelling Recognition Using Convolutional Neural Networks from Depth Map," *in Proceedings of the 3rd Asian Conference on Pattern Recognition*, Kuala Lumpur, pp. 1-5, 2015. https://doi.org/10.48550/arXiv.1509.03001

[12] Ketkar N., *Deep Learning with Python- A Hand on Introduction*, India, 2017. Doi: 10.1007/978-1-4842-27 66-4

[13] Kour K. and Mathew L., "Literature Survey on Hand Posture Techniques for Sign Language Recognition," *International Journal of Technical*

Research and Science, vol. 2, pp. 431-433, 2017. https://www.ijtrs.com/uploaded_paper/Literature%20Survey%20on%20Hand%20Gesture%201.pdf

[14] Lin J. and Ding Y., "A Temporal Hand Gesture Recognition System Based on Hog and Motion Trajectory," *Optik-International Journal for Light and Electron Optics*, vol. 124, pp. 6795-6798, 2013, https//doi:10.1016/j.ijleo.2013.05.097

[15] Liwicki S. and Everingham M., "Automatic Recognition of Fingerspelled Words in British Sign Language," *in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, Miami, pp. 50-57, 2009, https//doi:10.1109/cvprw.2009.5204291

[16] Lopez I., Hand Gesture Recognition for Sign Language Transcription, M.Sc Computer Science Thesis, Boise State University, 2017. https://scholarworks.boisestate.edu/cgi/viewcontent.cgi?article=2322&context=td

[17] Oguntimilehin A., Olatunji K., Abiodun A., Adeyemo A., and Balogun A., "A Review of Sign Language Recognition Techniques," *International Journal of Information Systems and Computer Sciences*, vol. 10, no. 6, pp. 1-8, 2021.

[18] Palacios J., Sagüés C., Montijano E. and Llorente S., "Human-Computer Interaction Based on Hand Gestures Using RGB-D Sensors," *Sensors*, vol. 13, no. 9, pp. 11842-11860, 2013. https://doi.org/10.3390/s130911842

[19] Rao K., Rajpal B., Prasad N., Pradeep N., and Shatri R., "A Survey on the Techniques Applied to The Recognition and Conversion of Indian Sign Language," *Journal of Applied Research on Industrial Engineering*, vol. 8, pp. 399-411, 2020. https//doi:dx.doi.org/10.22105/jarie.2020.259290.1214

[20] Roller O., Zargarian S., Ney H., and Bowden R., "Deep Sign: Enabling Robust Statistical Continuous Sign Language Recognition Via Hybrid CNN-Hmms," *International Journal of Computer Vision*, vol. 126, pp. 1311-1325, 2018.

[21] Rumana R., Rani R., and Prema R., "A Review Paper on Sign Language Recognition for the Deaf and Dumb," *International Journal of Engineering Research and Technology*, vol. 10, no. 10, pp. 329-332, 2021.

[22] Saleem M., Senan N., Wahid F., Aamir M., and Samad A., "Comparative Analysis of Recent Architecture of Convolutional Neural Network," *Mathematical Problems in Engineering*, pp. 1-9, 2022. https://doi.org/10.1155/ 2022/7313612

[23] Schönström K., "Sign Languages and Second Language Acquisition Research: An Introduction," *Journal of European 2nd Language Association*, vol. 5, no. 1, pp. 30-43, 2021. https//doi: doi.org/10.22599/jesla.73

[24] Sharma S., Saxena V., and Satish K., "Comparative Analysis on Sign Language Recognition System," *International Journal Scientific and Technology Research*, vol. 8, pp. 981-990, 2019.

[25] Shivashankara S. and Srinath S., "American Sign Language Recognition System: An Optimal Approach," *International Journal of Image Graphics and Signal Processing*, vol. 8, pp. 18-30, 2018. https//doi: 10:5815 /ijigsp.2018.08.03

[26] Singh S., Gupta A., and Singh T., "Sign Language Recognition Using Hybrid Neural Networks," *International Journal of Innovative Technology and Exploring Engineering*, vol. 9, no. 2, pp. 1092-1098, 2009. https://www.ijitee.org/portfolio-item/L33491081219/

[27] Tolentino L., Serfajuan R., Thio-ac A., Pamaboy M., and Forteza J., "Static Sign Language Recognition using Deep Learning," *International Journal of Machine Learning and Computing*, vol. 9, pp. 821-827, 2019. DOI:10.18178/ijmlc.2019.9.6.879

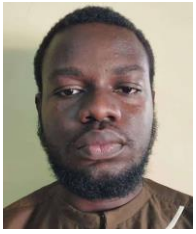
[28] Tukhtaev S. and Whangbo T., "A Combined Method of Skin-and Depth-Based Hand Gesture Recognition," *The International Arab Journal of Information Technology*, vol. 17, vo. 1, pp. 137-145, 2020. doi:10.34028/iajit/17/1/16.

[29] Yildiz A., Adar N., and Mert A., "Convolutional Neural Network Network Based Hand Recognition in Sophisticated Background for Humanoid Robot Control," *The International Arab Journal of Information Technology*, vol. 20, no. 3, pp. 368-375, 2023, doi.10.34028/iajit/20/3/9

[30] Zahid H., Rashid M., Hussain S., Azim F., and Syed S., "Recognition of Urdu Sign Language: A Systematic Review of the Machine Learning Classification," *Peej Computer Science*, vol. 8, pp. 1-25, 2022. https//doi: 10,7717/peej-cs.833

[31] Ziaie P., Muller T., and Knoll A., "A Novel Approach to Hand-Gesture Recognition in A Human-Robot Dialog System," *in Proceedings of the 1st Workshops on Image Processing Theory, Tools and Applications*, Sousse, pp. 1-8, 2008. DOI: 10.1109/IPTA.2008.4743760

Abiodun Oguntimilehin (Ph.D) is an Associate Professor of Computer Science, Department of Computer Science, Afe Babalola University, Nigeria. He obtained Ph.D in Computer Science from the Federal University of Technology, Akure, Nigeria. He is a chartered member of Computer Professionals (Registration Council of Nigeria), Member, International Association of Engineers (IAENG) and Nigeria Computer Society. His research interests are Medical Informatics, Data Science and Machine Learning. He has a number of publications in both reputable local and international journals.



Kolade Balogun is a graduate member of the Computer Professionals (Registration Council of Nigeria), he obtained B.Sc Computer Science from the Afe Babalola University, Ado-Ekiti, Nigeria. He is at present an Information Technology expert. His research interest is Data Science.