Development of a Sign Language E-Tutor Using Convolutional Neural Network

*[1]Opeyemi Adanigbo and [2]Oyeyemi T. Oyewole
[1]Department of Computer Engineering, Federal University, Oye-Ekiti, Nigeria
[2]Department of Computer Engineering, Elizade University, Ilara-mokin, Nigeria
opeyemi.adanigbo@fuoye.edu.ng|temitayo.oyeyemi@elizadeuniversity.edu.ng

Received: 01-JAN-2022; Reviewed: 01-APR-2022; Accepted: 15-JUN-2022
http://doi.org/10.46792/fuoyejet.v8i2.1055

ORIGINAL RESEARCH

Abstract- Deaf and hearing-impaired people typically use sign language as their primary form of communication. This study designed a Convolutional Neural Network-based Sign Language e-tutor which removes language barriers between people who are deaf and use sign language and people who can hear and speak. Thus, giving deaf people a way to communicate with hearing people in real time, with no need to write notes or use a human sign language interpreter. The method used is comprised of four major phases: data collection, data preprocessing, model training and model evaluation. The Model Precision, Accuracy, Recall and f1 score were 0.977, 0.985, 0.99, and 0.99 respectively.

Keywords- Accuracy, Convolutional Neural Network, Precision, Recall

1 Introduction

According to Treat (2016), 23.7% of Nigerians have hearing impairment and may require the use of sign language for communication. Hence, human-computer interaction (HCI) systems will prove to be a reliable and consistent solution for such persons. Sign language is a non-verbal language that involves the movement of fingers, hands, arms, heads, and bodies, as well as facial expressions, to express one's message (Chong and Lee, 2018). People are connected through communication because it allows them express their inner feelings, and exchange ideas verbally or non-verbally. The deaf community, on the other hand, is unable to communicate verbally. The introduction of sign language was intended to help hearing-impaired people communicate their sentiments to others. After more than 30 years of rigorous sign language research, the majority of sign languages around the world are still poorly documented or even known (Zeshan, 2004). In Nigeria, sign language varies across the region and 8.5 million citizens live with hearing impairments (Hassan *et al.*, 2017).

2 Literature Review

Sign language is a visual language that consists primarily of three main components: finger writing, meaning to spell words (Bayati and Hussein, 2010), and word-level associations, which include hand gestures conveying the meaning of words. Second, it recognizes the word-level symbol vocabulary composed of words or the entire gesture of the alphabet through image classification. (dynamic input/ video classification) and finally non-passive features including facial expression, tongue, mouth and body position.

2.1 Approaches to Sign Language Recognition

There are two major approaches to Sign Language Recognition (SLR), namely: Digital image processing-based Appro-ach for SLR and Data Glove-based Approach for SLR (Bhatia, 2020). The latter is not so commonly used as they are not suitable for sign language recognition due to their high cost or unavailability of specific sensors. This study focuses on DIP-based approach.

2.2 Sign Language Symbols

This contains lingual information which includes different symbols and letters. Sign language symbols are able to indicate all the sign parameters that include hand shapes, movement, location and palm orientation. SL symbols are classified into single handed and double handed signs. Furthermore, these signs are classified into static and dynamic signs. There are two broad classes of signs namely: One handed signs and two handed signs. Sign language consists of manual and non-manual elements as well. In manual signs, only hands are used to express any sign and in non-manual signs body postures, mouth gestures and face expressions are used.

2.3 Related Works

Hassan *et al.*, (2017) presented an intelligent recognition of static, manual and non-manual HSL. A Red Green Blue (RGB) digital camera was used for image acquisition, Fourier descriptor was used for features extraction and Artificial Neural Network (ANN) was used for classification. Thereafter, particle swarm optimization algorithm (PSO) was used to optimize the features based on their fitness in order to obtain high recognition accuracy. The optimized features selected gave a higher recognition accuracy of 90.5% compared to the manually selected features that gave 74.8% accuracy. However, the features extracted were chosen manually before being fed into the ANN algorithm which was used for classification which is subjective.

Rosero-montalvo *et al.*, (2018) presented an intelligent electronic glove system able to detect numbers of sign language in order to automate the process of

*Corresponding Author

Section B- ELECTRICAL/COMPUTER ENGINEERING & RELATED SCIENCES
Can be cited as:
Adanigbo O. and Oyewole T. (2023). Development of a Sign Language E-Tutor Using Convolutional Neural Network, FUOYE Journal of Engineering and Technology (FUOYEJET), 8(2), 192-196. http://dx.doi.org/10.46792/fuoyejet.v8i2.1055

communication between a deaf-mute person and others. This is done by translating the hand movement sign language into an oral language. The system is inside to a glove with flex sensors in each finger that they are used to collect data that are analyzed through a methodology involving the following stages: Data balancing with the Kennard-Stone (KS), Comparison of prototypes selection between CHC evolutionary Algorithm and Decremental Reduction Optimization Procedure 3 (DROP3) to define the best one and subsequently, the K-Nearest Neighbors (KNN) classification. As a result, the amount of data reduced from the first stage from storage within the system is 98%. Also, a classification performance of 85% is achieved with Cross-generational elitist selection, Heterogeneous recombination, and Cataclysmic mutation (CHC) evolutionary algorithm.

Akin-ponnle, (2021) presented a method of Machine Language Algorithm, which is deep learning, was adopted to classify and arrive at convenient model, by using a holistic Convolutional Neural Network (CNN) to recognize digits shown by human hands by an interpreter in a worship center in Nigeria. The CNN is known for both extracting a comprehensive feature representation from the input image and learning a classifier for each desired output simultaneously. How to implement a CNN for hand sign language digit classification from scratch in Google-colab and train the model in cloud environment is described in this study. The cloud-based training permits the use of the powerful GPUs for training, which reduces the training time in comparison with training on CPU. Experimental results show 93.5% accuracy on the test set.

Jebali *et al.*, (2021) presents a computer vision-based system to recognize the signs in continuous sign language video. The system is based on two main phases; sign words extraction and their classification. To separate sign words from video sequences, they present a new algorithm able to detect accurate words boundaries in a continuous sign language video. Using hand shape and motion features, this algorithm isolates signs from video. In the recognition phase, the extracted signs are classified and recognized using Hidden Markov Model (HMM) and other approaches such as Independent Bayesian Classifier Combination (IBCC) were also tested. This system had a recognition accuracy of 95.18% for one and 93.87% for two hand gestures respectively. This dataset contained 33 isolated signs.

3 METHODOLOGY

Convolutional Neural Network (CNN) is a deep neural network that consists of several layers of neural network. CNN is primarily used to perform image classification, object recognition and detection. CNN is comprised of learnable weights and biases of neurons that work by receiving inputs and performing dot product computation, which then determines the output of the network (Rakibul *et al.*, 2021). Unlike regular Neural Networks, the neurons of CNN architecture are arranged in three dimensions known as width, height and depth. Depth of CNN does not resemble the number of layers in the network of CNN, but refers to the dimension of the activation volume instead. CNNs have a sequence of layers that perform different functions (Amsaad,2023). There are three main types of layers in CNN architecture, which are: convolutional layer, pooling layer and fully connected layer.

3.1 YOU ONLY LOOK ONCE (YOLO)

Yolo is a CNN-based algorithm that performs object detection in real time. Object detection requires identifying things and locating them on the picture (Viswanatha *et al.*, 2022). A single-stage object detection was performed using the YOLOv7 algorithm. The three primary components of this single-stage object detector are: the model backbone which pulls crucial characteristics from an image, the model neck which mostly employs feature pyramids to aid generalized object scaling for improved performance on unknown data. The model head is in charge of the actual detection, which involves applying anchor boxes to features that create output vectors. The class probabilities, object scores, and bounding boxes are all included in these vectors. The object detector uses input photos to produce features, which are then sent into the prediction system, which draws boxes around objects and predicts their classes.

3.2 MATHEMATICAL MODEL

The head anticipates the object's bounding box and produces the xcenter, ycenter, w, h center coordinates, width, and height. The following is the equation for the expected bounding box:

$$b_x = (t\ ) + c_x \quad (1)$$

$$b_y = (t_y) + c_y \quad (1)$$

$$b_w = p_w \cdot et_w \quad (2)$$

$$b_h = p_h \cdot et_h \quad (3)$$

where p_w and p_h represent the width and height of the prior bounding box, respectively. (c_x, c_y) is the coordinate of the top left corner of the image.

3.3 LOSS FUNCTION

This study made use of IOU (Intersection over Union) to determine the degree of overlap between the predicted and the ground-truth bounding box. IOU is represented as follow:

$$IOU = M \cap N M \cup N \quad (4)$$

where M is the prediction bounding box, represented by (xcenter, ycenter, w, h). N is the ground-truth bounding box (x, y, w, h). However, this optimization method has the disadvantage of not being able to optimize non-overlapping parts. Therefore, the generalized GIOU loss function is represented as follows:

$$GIOU = IOU - |Ac - U| \ |Ac| \quad (5)$$

where Ac represents the minimum bounding box between the predicted bounding box and the ground-truth bounding box. U is the union of the predicted and the ground-truth bounding boxes, i.e., M ∪ N. The loss function not only pays attention to the overlapping area,

http://doi.org/10.46792/fuoyejet.v8i2.1055 http://journal.engineering.fuoye.edu.ng/

but also focuses on the non-overlapping area of the two kinds of boxes which better reflects the overlap of the two boxes. The bounding box regression loss function used in this article is:

$$LossGIOU = 1 - GIOU \quad (6)$$

The value range of GIOU is (-1, 1). The higher the overlap of the bounding box M and N, the closer the GIOU is to 1. When M and N do not overlap, optimization can still be performed, which benefits from the existence of the smallest bounding boxes, in contrast, this advantage is missing from IOU.

3.3 Dataset

The dataset used in this project is a primary dataset, consisting of sign images taken by the author, consisting of many classes with each class containing 50 to 100 images. The dataset consists of images of characters, A to Z and commonly used words. The images were annotated using labelimg, the object recognition was done using the You only look once version 7(YoloV7) algorithm and Streamlit. The design system was developed with python programming language.

The dataset consisted of 11,340 jpeg images of 200 x 200 resolution, the number of sign classes were 54, the channel was 3(RGB) and file sizes ranged between 14kb and 30kb. Data preprocessing was carried out on the images before becoming input to the training process. The preprocessing data includes resizing to 416 × 416, conversion from RGB to grayscale, and adding data by rotating the available images by 1 and 2 degrees clockwise and counterclockwise. Preprocessing is done so that the data varies and equates the size of the training data.

3.4 Model Development

The phases involved in the model development are highlighted in the following sub-sections:

3.4.1 Preprocessing

LabelImg was used for annotating the images. And for producing the manual bounding box labels on the source pictures. After that, each of the photos and bounding box coordinates went through an augmentations pipeline, which scaled the images to 1024 × 1024-pixel squares and added probability for various modifications. A sample picture of the image annotation is displayed in Figure 1.

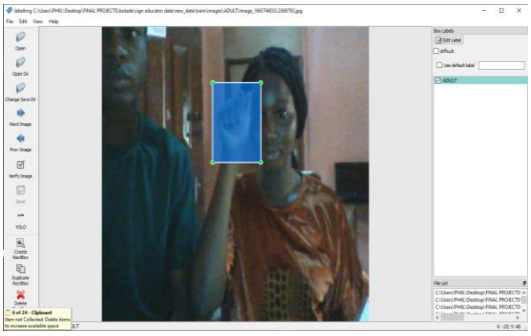

Fig. 1: Sample screen showing the image annotation

Data transformation techniques which involved specific degrees of image rotation, changes in picture positions, blurring, horizontal flips, random erase, and a number of additional color modifications were used.

3.4.2 Model Training and Testing Phase

The training set contained a total of 8,100 images with 54 Classes and 150 image categories. The validation set contained a total of 3240 images with 54 Classes and 60 image categories. The overall number of images contained in the dataset was 11, 340. The training was done on Google colab using the GPU runtime with Yolov7 and the web application was designed with streamlit.

3.4.3 Model Evaluation Phase

This phase is executed after the image dataset model training using yolov5 object detection model. The web application was developed using streamlit API on a web application. The app created then makes prediction with object detection, visualize those predictions at a given confidence level.

4 Results and Discussion

The model is trained using the training and validation sets representing 71.4% and 28.6% of the dataset distribution, respectively. The Neural network completes classification tasks by learning key features for individual classes in the dataset. The classification model attained excellent performance with no transfer learning. In order to avoid the possibility of data leakage which can cause biased transfer learning, the model was initialized for classification tasks. The model was trained for 50 epochs and 4 batch sizes, with experimentation performed on multiple hyperparameters, including its regularization, and learning rate. The best values obtained for the initial learning rate was 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.

The training was done on Google colab using the GPU runtime with Yolov7 and deployed on the web application with streamlit. The web application comprises of the four different menus, which are the home, sign up, login and about. After the sign up, a user can login and access different features such as options for different sign language and also option to test the sign language either with an image, video, or webcam.

4.3 Evaluation with Precision and Recall

The metrics in Table 1 are indicators of the models' performances. Precision (P) measures the quality of a positive prediction made by the model, while recall (R) describes the percent of positives that are classified correctly. The mean average precision (mAP) is a metric used to evaluate object detection models. Using a precision-recall curve, the mAP was calculated by the weighted mean of precisions at each intersection-over-union (IoU) threshold. The IoU indicates the overlap of the predicted bounding box compared to the ground truth box. Therefore, a higher IoU indicates that the

http://doi.org/10.46792/fuoyejet.v8i2.1055 http://journal.engineering.fuoye.edu.ng/

predicted bounding box is very close to the ground truth box coordinates. The most significant metric is the mAP@0.5 which is interpreted as the accuracy. The mAP@0.5:0.95 averages the mAP at each IoU threshold within the interval, but at higher thresholds like 0.95, the margin of error is unnecessarily rigorous. As depicted in Table 1, the YOLOv7 and YOLOv7x have significantly higher mAP@0.5 and faster runtimes than the other models.

Table 1. Evaluation Result after 50 Epoch Training

Epoch	Precision	Recall	mAP@0.5	mAP @0.5:0.95
1	0.4802	0.4622	0.2458	0.2076
2	0.3815	0.5508	0.2852	0.2408
3	0.1792	0.7488	0.3465	0.2888
4	0.1222	0.7647	0.2787	0.2176
5	0.3588	0.5545	0.3755	0.3099
6	0.4281	0.6025	0.4497	0.3717
7	0.2866	0.7059	0.4668	0.3939
8	0.3494	0.7691	0.551	0.4672
9	0.4153	0.7446	0.5921	0.503
10	0.427	0.7776	0.6484	0.5443
11	0.5193	0.7842	0.7035	0.5938
12	0.518	0.7759	0.7176	0.614
13	0.5994	0.7852	0.7317	0.626
14	0.5381	0.7962	0.7342	0.6313
15	0.5799	0.8353	0.7749	0.6595
16	0.7258	0.7781	0.8111	0.6917
17	0.53	0.8398	0.809	0.67
18	0.4682	0.757	0.7485	0.5804
19	0.598	0.8615	0.8097	0.6596
20	0.6086	0.853	0.8355	0.6744
21	0.7388	0.8555	0.8855	0.7354
22	0.7475	0.9253	0.9233	0.7731
23	0.8504	0.8906	0.9314	0.7886
24	0.8153	0.9253	0.9234	0.7731
25	0.8719	0.9555	0.9553	0.811
26	0.8595	0.9338	0.936	0.7936
27	0.9016	0.9592	0.957	0.8199
28	0.884	0.9563	0.947	0.8161
29	0.8414	0.9175	0.9415	0.8043
30	0.9247	0.9142	0.967	0.8239
31	0.863	0.9521	0.9533	0.8149
32	0.8689	0.9217	0.9604	0.8144
33	0.8366	0.9316	0.9502	0.7892
34	0.7952	0.9377	0.9333	0.7716
35	0.8369	0.9194	0.9461	0.7788
36	0.9148	0.8996	0.9531	0.7982
37	0.893	0.9292	0.9644	0.8157
39	0.9099	0.9317	0.9626	0.8171
40	0.9275	0.9469	0.9621	0.8277
41	0.95	0.9207	0.9761	0.8403
42	0.963	0.9521	0.9533	0.8149
43	0.9646	0.9201	0.9579	0.803
44	0.9701	0.9265	0.9587	0.7973
45	0.972	0.9193	0.9518	0.7969
46	0.9729	0.9357	0.9428	0.7855
47	0.975	0.9294	0.9508	0.793
48	0.977	0.9012	0.9577	0.8042
49	0.98	0.9517	0.9643	0.8222
50	0.9836	0.9727	0.9723	0.8316

Performance Evaluation Curve for Precision, Recall and mAP are displayed in Figure 2. The Neural Network was trained for 50 epochs and that took time on GPU on Google colab. Key values that were regularly checked during the training were: the loss function value, precision, recall, mAP and average IoU. As seen from the Table 1, all of those values continually improved until the 50th epoch, when the model started over-fitting and then the training was stopped. performance. Figure 2 shows the plot of training accuracy against the validation accuracy and Figure 3 shows the individual model plots for precision and recall. It can be seen that the model performed optimally at about the 40th epoch.

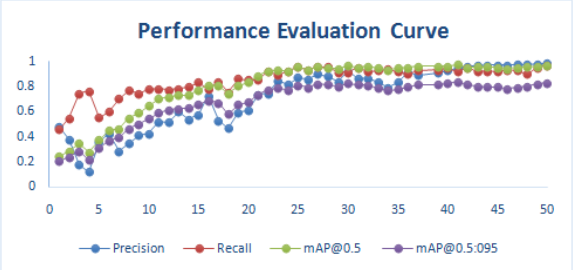

Fig 2: Performance Evaluation Curve for Precision, Recall and mAP

The individual plots for precision and recall are as depicted in Figure 3. The YOLO algorithm processed input pictures from the dashboard camera with the frame resolution 200 x 200 pixels at the average of 23 frames per second. The model mAP which is the same as accuracy is shown in Figure 4.

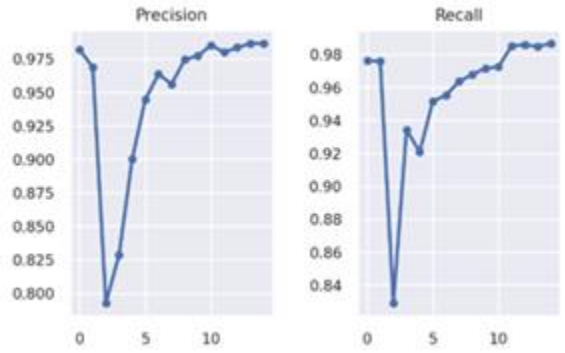

Fig 3: Graph plots showing model precision and recall

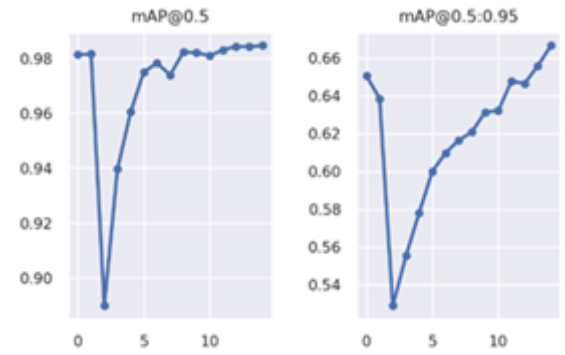

Fig 4: Graph plots showing model mean average precision (mAP) @ 0.5 and @ 0.5:0.95

http://doi.org/10.46792/fuoyejet.v8i2.1055 http://journal.engineering.fuoye.edu.ng/

5 Conclusion and Future Work

The developed system implemented 54 signs on the web application for the end users. The proposed sign language recognition system can be further extended to recognize gestures and facial expressions. Instead of displaying letter labels, it could be designed to display sentences as more appropriate translation of language. The sign language can later be deployed on the mobile application.

References

Akin-ponnle, A. E. (2021). *Cloud - Based Human Sign Language Digit Classification Using CNN : A Case Study of*. *7*(5), 3899–3903.

Amsaad, F., Prasanna, P.,Pravallika, T., Mamatha, G., Raviteja, B., Lakshmi, M., Alsaadi, N., and Tashtoush, Y. (2023). Toward Secure and Efficient CNN Recognition with Different Activation and Optimization Functions. 10.1007/978-3-031-33743-7_45.

Bayati, M., and Hussein, K. (2010). Comparison Between Modes of Communication for the Deaf and Dumb via e-Learning Through Case Study (sign language and finger spelling in review).. *JDCTA. 4. 36-39. 10.4156/jdcta.vol4.issue2.4.*

Bhatia, P., Verma, S., and Kaur, S. (2020). Sign Language Generation System Based on Indian Sign Language Grammar. ACM Transactions on Asian Language Information Processing.

Chong, T., and Lee, B. (2018). *American Sign Language Recognition Using Leap Motion Controller with Machine Learning Approach*. https://doi.org/10.3390/s18103554

Hassan, S. T., Abolarinwa, J. A., Alenoghena, C. O., Bala, S. A., David, M., and Farizamin, A. (2017). *Intelligent Sign Language Recognition Using Enhanced Fourier Descriptor : A Case of Hausa Sign Language*.

Jebali, M., Dakhli, A., and Jemni, M. (2021). Vision - based continuous sign language recognition using multimodal sensor fusion. *Evolving Systems*, *2016*. https://doi.org/10.1007/s12530-020-09365-y

Rakibul, H., Al Mahmud,A., Mokhlesur, Yasin, Hossain, and Razib,H . (2021). A Review on the Convolutional Neural Networks (CNN) and Hand- Written Digit Recognition Using Deep Convolutional Neural Network. 6. 1-18.

Rosero-montalvo, P. D., Godoy-trujillo, P., Flores-bosmediano, E., Carrascal-garc, J., Otero-potosi, S., Benitez-pereira, H., & Peluffo-ord, D. H. (2018). *Sign Language Recognition Based on Intelligent Glove Using Machine Learning Techniques*. 5–9.

Treat, S. (2016). Deaf education: Gallaudet University: How does education and special education is being advanced in Nigeria.

Viswanatha,V., Chandana, R K, and Ramachandra, A.. (2022). Real Time Object Detection System with YOLO and CNN Models: A Review. *Xi'an Jianzhu Keji Daxue Xuebao/Journal of Xi'an University of Architecture & Technology. XIV*. 10.37896/JXAT14.07/315415.

Zeshan, U. (2004). Sign Languages of the World. *Elsevier Ltd.*, *1989*, 358–365.