ISSN: 2993-8678

Research Article

Current Trends in Mass Communication

An Intelligent Support System for Speech and Hearing-Impaired Persons Using Deep Learning for Nigerian Sign Language

Okorie Emmanuel O[1], Nachamada V Blamah[1] and Gideon Dadik Bibu[1,2]

[1]*University of Jos, Department of Computer Science, Nigeria*

[2]*Higher Colleges of Technology, CIS Division, UAE*

***Corresponding Author**
Okorie Emmanuel O, University of Jos, Department of Computer Science, Nigeria.

Submitted: 2025, Aug 05; **Accepted:** 2025, Sep 30; **Published:** 2025, Oct 09

Citation: Okorie E. Blamah, N. V., Bibu, G. D. (2025). An Intelligent Support System for Speech and Hearing-Impaired Persons Using Deep Learning for Nigerian Sign Language, *Curr Trends Mass Comm, 4*(3), 01- 12.

Abstract

This paper proposes an intelligent support system for speech- and hearing-impaired individuals using deep learning for Nigerian Sign Language (NSL). The system enables real-time communication between hearing/speech-impaired and normal individuals by converting spoken words into text for the impaired, and signs into audio for the normal person. The system utilizes a TensorFlow model trained using transfer learning to recognize NSL signs and convert them into audio signals. The model was trained using a dataset of English words converted to NSL, achieving an average confidence rate of 72.63%. The system comprises two main parts: a sign language recognition module and a speech recognition module. The sign language recognition module recognizes signs made by the hearing/speech-impaired person, while the speech recognition module recognizes words spoken by a normal person. The system has the potential to facilitate communication and reduce discrimination and social exclusion in hearing/speech-impaired individuals.

Keywords: Sign Language, Image Recognition, Text-To-Speech, Audio-To-Text, TensorFlow API, Deep Learning, Convolutional Neural Networks

1. Introduction

According to recent estimates, approximately 430 million people, or over 5% of the global population, suffer from hearing and speech impairment. This number is expected to increase to around 2.5 billion by the year 2050, with at least 700 million requiring hearing rehabilitation [1]. In Nigeria, it is estimated that approximately 8.5 million individuals live with hearing impairments , and another study suggested that over 10 million Nigerians are deaf or hard of hearing [2,3]. These individuals often experience discrimination and social exclusion due to communication difficulties, and 90% of the children are excluded from school for similar reasons. The study also noted that American Sign Language (ASL), which is used in Nigerian schools for the deaf, fails to meet the local language needs of the deaf population because local signs are widely used and more reflective of Nigerian society, but they are routinely ignored. Deaf people around the world communicate using sign language, which is a visual language that involves facial and body movements as a means of communication. Sign languages are not universal, and different sign languages are used in different countries, much like the many spoken languages worldwide. Some countries, such as Nigeria, Belgium, the UK, the USA, and India, may have more than one sign language because no single sign language is universal.Click or tap here to enter text.

2. Literature Review

Numerous studies have been conducted to facilitate communication between individuals with speech and hearing impairments using a two-way communication model. For instance, a real-time system for hand gesture recognition was developed in that can recognize hand gestures and convert them into voice images, and vice versa. However, the system's reliability is limited by its feature extraction and illumination constraints [4]. Another real-time sign language

recognition system was proposed in that uses skin color determination and a convex hull algorithm to determine hand gestures [5]. The K curvature and convex hull algorithms are essential for feature extraction in low-light sign language recognition systems. In a real-time sign language recognition application was developed to detect and process gestures from the Indian Sign Language (ISL) dictionary using a Convolutional Neural Network (CNN) model [6]. The application can detect both static and dynamic gestures and generate Python code for various constructs. The major achievement of this research is the introduction of programming knowledge to hearing and speech-impaired people and improving access to education for them [7]. Proposed the extraction of hand and body from video sequences for Argentinian sign language. The model was pre-trained on the ImageNet VGG-19 network and employed linear dynamic system (LDS) histograms with four stream deep neural networks that consist of stacked LSTM (Long Short-Term Memory) layers [8]. However, difficulties were encountered in extracting accurate skeletal data due to occlusions, even though communication was not performed in real time [9]. In researchers employed a convolutional neural network (CNN) for deep learning to recognize static hand gestures in American and Singaporean sign languages. The CNN eliminates the laborious and computationally extensive feature extraction phases of traditional recognition approaches. However, the model failed to recognize American fingerspelling for letters 'j' and 'z' since they involve motion [10]. In an approach to video-based continuous sign language recognition (CLSR) was proposed, which includes a convolutional neural network (CNN) for spatial feature extraction, stacked one-dimensional temporal convolutional layers (TCL) for short-term temporal modeling, and a bidirectional long short-term memory (BLSTM) unit for global context learning. Although the study showed improved recognition accuracy, it did not extend the method to sign language translation or simultaneously explore the relationships among video, sign language, and spoken language. The study achieved promising results when evaluated on three challenging sign language recognition datasets [11]. In 2018, introduced a real-time alphabet sign language image classification method using deep learning. The network can perform real-time predictions using image frames from a web camera at rates of 50–100 Hz. However, the model couldn't recognize letters j and z, and it wasn't a two-way communication system [12]. Finally, used deep learning to recognize American sign-language gestures by using static images. Although the model demonstrated improved accuracy, it couldn't recognize common words and expressions, symbols involving two hands, or dynamic gestures, as it only detected static finger spellings.

Researchers developed a neural network approach for recognizing common sign languages through hand and sign language recognition [13]. The study focused on a hand location network, a 3D CNN feature extraction network, and an LSTM-based sign language recognition framework. These networks were integrated to create a recognition algorithm for RGB sign language image or video recognition. The study did not address real-time communication as in other studies. Researchers proposed attention-based 3D-CNNs for sign language recognition, which utilized spatial and temporal attention mechanisms [14]. The model achieved 99% accuracy on the Chinese Sign Language dataset and suggested that Capsule Networks would yield better results than Inception Networks. Researchers proposed a model that utilized video sequences to extract temporal and spatial features using Inception, a CNN for recognizing spatial features and an RNN for training temporal features [15]. The model achieved 99% accuracy on a custom American sign language dataset. In 2019, researchers proposed a capsule network for sign language recognition. Compared to LeNet, a widely used deep-learning model, capsule networks achieved 88% accuracy on the test set and outperformed LeNet (82% accuracy) on the MNIST sign-language dataset. By augmenting the training data, the success rate of the capsule network increased to 95%. However, this study only recognized American letters, except for J and Z, as they required movement representation. Another capsule-based deep neural network for American Sign Language (ASL) fingerspelling was proposed in [16]. The framework uses a capsule network with adaptive pooling and achieved 99% accuracy. This approach has potential for nonverbal communication in human-robot interactions but cannot recognize non-static gestures. For Bangla sign language, a deep-learning-based framework was developed to recognize 37 Bengali signs with 96.33% accuracy [17]. However, this framework suffered from misclassifications [18]. utilized sign language recognition through the customized region of interest (ROI) segmentation and convolutional neural networks (CNN) and revealed superior performance in terms of accuracy and real-time detection from video streaming via webcams compared to conventional approaches. However, the system employed Raspberry Pi and lacked a graphical user interface (GUI) for enhanced operation [19]. proposed a Bengali sign language system using convolutional neural networks, categorizing hand skeletal features into standard communicative meanings. However, it failed to concatenate images and words to form expressive words or phrases. In 2018, a real-time system was proposed to convert Indian Sign Language into text, but it only covered numeral signs and used handcrafted features [22]. However, all studies reviewed on Indian sign language have failed to provide two-way communication for hearing/speech-impaired and normal individuals.

Several studies have utilized a combination of handcrafted features and deep learning methods for sign language recognition. For example, used a serial fusion method to combine features from handcrafted methods and VGG-19, a deep learning network, and then fed the resulting data to a classifier, SVM, for classification [29]. However, this approach is more complex and has not been developed for two-way communication proposed a real-time hand gesture recognition system based on shape-based features, while developed a mobile device-based sign language translation system using depth-only images [32-34]. proposed a model for extracting signs from videos with minimally cluttered backgrounds, presenting them as readable text form with 26 ASL letters and three special signs, but was limited to a certain character count and did not wrap the text around the next line. Consequently, this system does not allow long sentences to be written.

In 2019, a deep convolutional neural network was introduced by for recognizing and categorizing the Ghanaian Sign Language [35]. However, it did not facilitate real-time communication between users. Studies on Arabic sign language recognition were conducted by,and Bangladeshi Sign Language was investigated by [36,37]. Nevertheless, developed a real-time sign language detector that utilized American Sign Language (ASL) to enhance communication between the deaf community and the public [38]. The system, which used a pre-trained Convolutional Neural Network (CNN) architecture, SSD Mobile Net V2, recognized selected sign languages with an accuracy of 70-80% under various conditions, including low light and uncontrolled backgrounds. The system captured images of hand gestures using a web camera and OpenCV, while TensorFlow, Tensor Object Detection API, and LabelImg were also employed. The model achieved high accuracy in recognizing symbols, with Yes (88.7%), No (88.6%), Thank You (84.1%), I Love You (82.4%), and Hello (91.0%) being the most accurate. However, limitations such as environmental factors, low light intensity, and uncontrolled backgrounds impacted the accuracy of detection. Additionally, the system did not provide a two-way communication platform. Similarly, proposed a real-time communication system that enables two-way communication between hearing-impaired and non-hearing-impaired individuals through sign language-to-text and text-to-sign language conversion [39]. The system could interpret letters, numbers, and words in Indian Sign Language and achieved an accuracy of 99% in predicting 17,600 test images in 4 s, with an average prediction time of 0.000805.

proposed the use of an intelligent recognition system for static, manual, and non-manual Hausa sign language (HSL) using a Particle Swarm Optimization (PSO)-enhanced Fourier descriptor [2]. A vision-based approach was employed with an RGB digital camera for image acquisition and Fourier descriptors for feature extraction. This paper presents the first study to use deep learning and computer vision techniques to recognize Nigerian local sign languages. The paper also introduces a two-way real-time sign recognition system for the Nigerian Sign Language.

3. Methodology and Experimentation

Recently, deep learning approaches have gained attention for sign language recognition. In this paper, a deep learning approach that is designed to develop a real-time sign language detector using the TensorFlow object detection API and train it through transfer learning using our custom dataset is presented. The architectural design of the system is shown in Figure 2.1 This is a CNN-based SLR system composed of a signer, object detection, and non-signer.

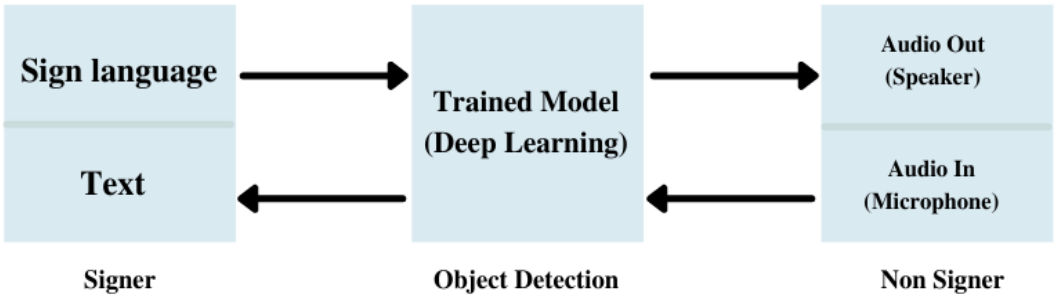

Figure 3: 1 Proposed Architectural Design

- 3.1 Signer
- This part comprises of
- Sign Language
- Text
- 3.1.0 Sign Language

Here, computer vision is used to recognize the signs made by signers (hearing- and speech-impaired). The OpenCV library was used for computer vision. OpenCV (Open-Source Computer Vision) is a library of programming functions for computer vision. OpenCV's deep neural network (DNN) library and TensorFlow Object Detection API offer easy-to-use, open-source frameworks, where pre-trained models for object detection (such as those downloadable from the TensorFlow model zoo) achieve high accuracy in detecting various objects from humans to TV monitors [40]. The OpenCV Library was used for two purposes.
1. Data Acquisition
2. Accepting Signs for prediction

3.1. Data Acquisition

For the data acquisition, images were captured with a webcam using Python and OpenCV. The created dataset consists of signs representing words in Nigeria Sign Language. A total of 100 English words were translated into Nigerian sign language by the two actors. The translated words are shown in Table 2.1. A Python script was used to capture the actors as they made signs for each word; each actor continuously made the sign in front of a web camera for 30 s for each word listed in Table 2.1. Images were captured every 30s to record gestures. Thirty images were generated for each word within 30 seconds. A total of 60 images were generated by the two actors for each word, although some words contained more than 60 images, because each actor generated more signs for a few words. A total of 7963 images were generated and the captured images were stored in their respective

folders. The Uuid library was used to name image files. It helps in the generation of random objects of 128 bits as ids providing uniqueness, as the ids are generated based on time and computer hardware.

Figure 3: 2 LabelImag for letter

hello	basics	hate	walk	basket	beef	bell	bathroom	bedroom	beginner
thanks	iloveyou	bath	become	begin	yes	a	beauty	beg	no
barber	work	my	barbecque	battle	friend	baptism	battery	enemy	coming
hi	ban	brain	bank	jesus	bag	ball	have	bad	bake
school	church	for	welcome	bachelor degree	go	home	are	abandon	because
academics	are you married	come	accept	are you hungry	bachelor	accident	are you deaf	baby	book
class	okay	aware	abuse	you	arrest	female	i	love	able
male	land	are you sick	jealous	jean	jewellry	join	joke	judge	about
jot down	jubilation	keep it	kidnap	killed	kid	she is	kind	kindness	absent
king	queen	backup	above	jail	jesus christ	new	keep	he is	kingdom

Table 3: 1 Data set Library

Once all images were captured, they were labelled individually using the LabelImg package. LabelImg is a free open-source tool for graphically labelling images. The hand-gesture portion of the image was labelled by a gesture in the box or sign, as shown in Figure 2.1. An XML file was created to save the labelled images. The XML files contain all details of the images, including the details of the labelled portion. After labelling all the images, their XML files were available. This was used to create the TF (TensorFlow) records. All images, along with their XML files, were divided into training and validation data at a ratio of 80:20. From the 60 images of each word, 48 (80%) were taken and stored as a training dataset and the remaining 12 (20%) were taken and stored as a validation dataset. This task was performed on all images of 100 words.

3.1.1. Accepting Signs for Prediction

In this prediction phase, a signer (speech and hearing-impaired person) makes a sign to communicate with a non-signer. This sign is captured by the webcam and processed using OpenCV, which in turn serves as input to the trained model to predict the signs made in front of the webcam.

3.1.2. Speech to Text Conversion Methods

The text was displayed on the graphical user interface (GUI) Of OpenCV. This is the direct output of a non-signer block. Spoken words from the microphone by a non-signer were processed using a deep learning model and converted to text. Speech-to-text techniques were used in this study.

Artificial neural network classifier (ANN)-based Cuckoo Search Optimization was used in this research for several reasons; this method is widely adopted by companies such as Google, IBM, and Microsoft, and they have used it to develop speech-to-text recognition APIs that are available for researchers to use. Therefore, we utilized the Google Speech-to-Text API to convert spoken words from the microphone displayed by a non-signer to text displayed on the GUI for a signer to read.

- 3.2 Non-Signer
- This block, like a signer block, consists of two sections.
  - Audio Out (Speaker)
- Audio In (Microphone)

3.2. Audio Out (speaker)

Audio Out is the signer's sign output. Before coming out as text, the deep learning model (object detection block) makes a prediction based on a sign created by the signer. The sign is classified as a word, which is then converted into speech (audio) for a non-signer to hear and respond to. To achieve this, we use a text-to-speech technique.

3.2.1. Text To Speech Conversion

Text-to-speech is a process in which the input text is first analyzed, processed, and understood, and then converted to digital audio and spoken. Figure 1.2 shows a block diagram of the TTS. The figure shows all the steps involved in text-to-speech conversion; however, the main phases of the TTS systems are as follows [41]

3.2.2 Audio In (Microphone)[Heading blue colour]

In this section, a non-signer speaks
on the microphone and the audio is converted to text for a signer to read on the OpenCV GUI. The entire process is explained in section 2.1.1.0.to read on the OpenCV GUI. The entire process is explained in section 2.1.1.0.

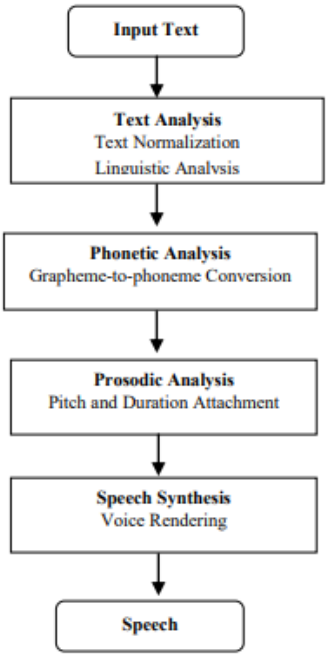

Figure 2: 3 Text to speech System Flow [41]

3.3. Object Detection

Deep learning is a type of machine learning technique whose architectures, such as deep neural networks (DNN) and convolutional neural networks (CNN), have been applied to computer vision, speech recognition, and natural language processing. Computer vision has rapidly developed along with deep learning technology, which has been applied in various industries, such as engineering, agriculture, and medicine. Computer-vision technologies based on deep learning for image classification and detection are classified into classification, object detection, and segmentation. Object detection is the detection of multiple objects in a single image [42].

TensorFlow, an end-to-end open-source machine learning platform, is one of the most popular deep learning frameworks. It provides useful libraries and resources for building and evaluating models. Furthermore, it offers high-performance application programming interfaces (APIs) to build DNN- and CNN-based models, such as voice analysis, natural language processing, and computer vision [42]. Currently, the most commonly used algorithms for object detection in computer vision are CNNs [43], which have been proven to surpass the human-level performance in image classification. The model used in this study was trained to detect objects in real-time. This was achieved using a universal and open-source library, the TensorFlow Object Detection API.

The TensorFlow Object Detection API is an open-source framework built on top of TensorFlow, which facilitates the construction, training, and deployment of object detection models. TensorFlow object detection API is a framework for creating a deep learning network that solves object detection problems [44]. There are already pretrained models in the framework, which are referred to as model zoos. This includes a collection of pretrained models trained on the COCO, KITTI, and open image datasets. These models can be used for inference if we are interested in the categories only in this dataset. They are also useful for initializing models when training on novel datasets. The various architectures used in the pre-trained model are listed in Table 1.1:

Model name	Speed (ms)	COCO mAP	Outputs
CenterNet HourGlass104 512x512	70	41.9	Boxes
CenterNet HourGlass104 Keypoints 512x512	76	40.0/61.4	Boxes/Keypoints
CenterNet HourGlass104 1024x1024	197	44.5	Boxes
CenterNet HourGlass104 Keypoints 1024x1024	211	42.8/64.5	Boxes/Keypoints
CenterNet Resnet50 V1 FPN 512x512	27	31.2	Boxes
CenterNet Resnet50 V1 FPN Keypoints 512x512	30	29.3/50.7	Boxes/Keypoints
CenterNet Resnet101 V1 FPN 512x512	34	34.2	Boxes
CenterNet Resnet50 V2 512x512	27	29.5	Boxes
CenterNet Resnet50 V2 Keypoints 512x512	30	27.6/48.2	Boxes/Keypoints
CenterNet MobileNetV2 FPN 512x512	6	23.4	Boxes

Table 3: 2 TensorFlow Pretrained Models

Creating accurate machine-learning models capable of localizing and identifying multiple objects in a single image remains a core challenge in computer vision. However, the TensorFlow object detection API was used. [40] *stated that* neural network-based classifiers were used together with other object-detection techniques. The TensorFlow Object Detection API can be used for training and testing an SSD using the MobileNet model. The objective is to detect each sign in a video (webcam) and to localize every detection identified together in space and time.

In most cases, training a complete convolutional network from scratch is time consuming and requires massive datasets. Accordingly, this problem can be resolved by utilizing the power of transfer learning with a pretrained model using the TensorFlow API[45]. Because we used a small dataset, we adopted transfer learning to train our model using SSD. The SSD applies a smooth L1-Norm to determine the location loss. During the classification process, SSD performs object classification. Hence, for every predicted bounding box, collections of N-category predictions are calculated for each likely category in the dataset. Furthermore, feature maps describe the features of interest of an image at various scales; hence, working MultiBox on multiple feature maps increases the likelihood of any object being large or small, which is eventually detected, localized, and properly classified.

The TensorFlow API writes model performance-related logs and optimizer states using the tfevents format. There are two main tfevents that you want to keep track of: **training-related** and evaluation-related. The training event was limited to loss- and learning-rate tracking. It tracks the number of steps per epoch so that one can see how quickly one's training job is going. When viewed from TensorBoard, we can see the loss of a component-level breakdown (separately for classification and localization) as well as the total value. It becomes especially useful when faced with a problem, and aims to examine the model to find its root cause. In addition, we can keep track of how the learning rate changes over time and how many steps per second our training job requires. However, the evaluation tfevent is like the training tfevent; it consists of a loss component with the same breakdown. Additionally, the evaluation metrics were tracked.

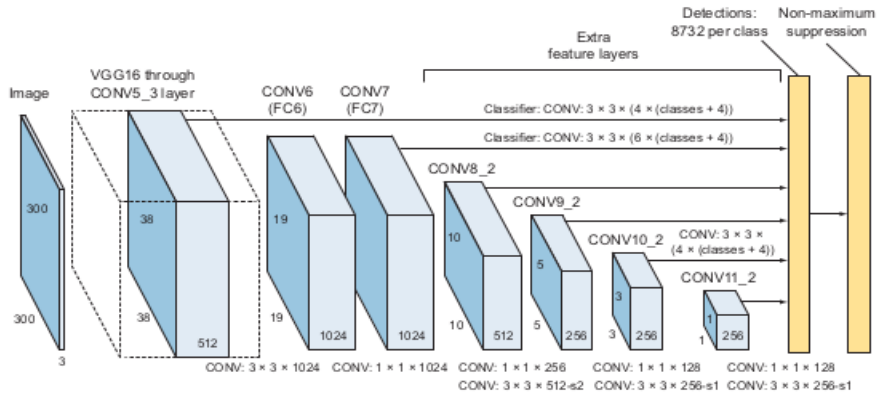

Figure 3: 4 SSD Architecture [46]

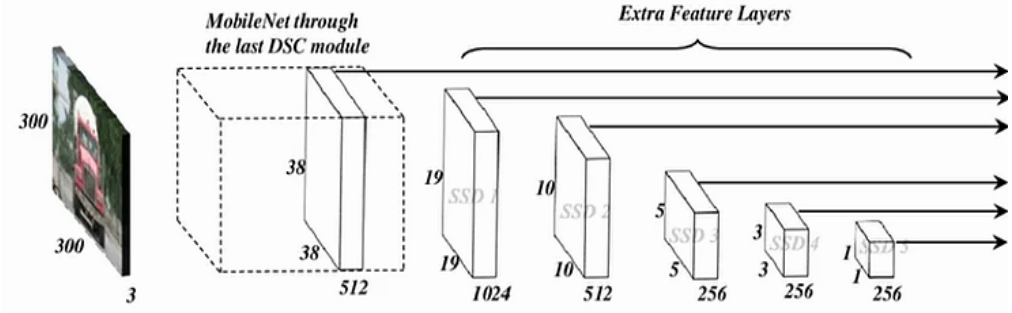

Figure 2: 5 MobileNet Architecture [47]

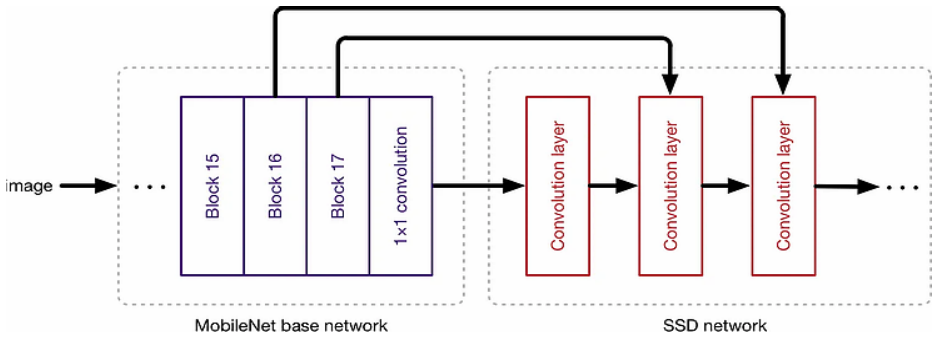

Figure 3: 6 MobileNet SSD Overview [48]

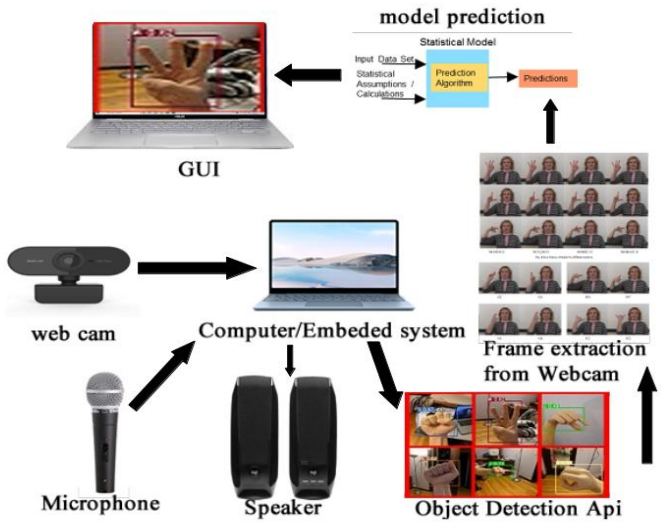

Figure 3: 7 Graphical Representation of the Proposed Framework

4. Experimental Results

4.1. Dataset and Experimental Setup

The dataset was created for Nigeria Sign Language, where the signs are words of the English language. The dataset was created using the data acquisition method described in Section 3.1.1. Two actors from a non-governmental organization called the Deaf Technology Foundation voluntarily contributed to the acquisition of data, which helped the speech- and hearing-impaired people in society to communicate seamlessly. Each actor made 60 signs for each word, within an interval of 60s. A total of 1608 images were obtained. The dataset consisted of 1608 images, which were divided into training and test datasets, 80% for trains, and 20% for tests. For all the image sizes, the XML generated for training was approximately 1.6GB. The experiment was conducted on a PaperSpace cloud server, with the specifications shown in Figure 3.1.

The programming environment included Python (version 3.7.3), Jupyter Notebook, OpenCV (version 4.2.0), and the TensorFlow Object Detection API.

Parameter	Value
Machine Type:	P5000 1 Upgrade Deactivate
Billing Type:	hourly
Region	NY2
Hostname	ps1hvy8gj
Private IP	10.64.57.37
Public IP	64.62.141.159 Remove Public IP
Memory	30 GB
CPUs	8
Storage	100 GB
GPU	16 GB
Network	Paperspace

Figure 3: 1 Paper Space Server Configurations/Specification

4.2. Results and Discussion

The model was trained for a total of 20000 epochs, as shown in Figures 4.2 to 4.8, which depict the training process. The training was completed in approximately 2h owing to the use of a GPU. The model was trained for 20000 steps (epochs) with a batch size of 64, which required approximately two hours for completion. Figures 4.2 - 4.6 illustrate the evolution of the total loss during training, while Figure 4.7 shows the individual steps of the training process. Figure 4.8 displays the learning rate of the model, which was randomly generated by the TensorFlow object detection API from the configuration settings. The use of a variable learning rate, rather than a constant rate, can help achieve greater accuracy in a shorter amount of time. From the loss graphs, the total loss decreased rapidly owing to the pretrained model. This is a positive outcome, because it indicates that the training loss in a good model continues to decrease as the model is trained. A minimum loss is desirable, and a decreasing value indicates that the model is learning during the training. Training can be discontinued at any time if loss ceases.

As shown in Figure 4.2, the total loss value rapidly decreases when training commences from a pre-trained checkpoint rather than from scratch. Although the total loss values fluctuated, they exhibited an overall decreasing trend. It is important to note that the faded orange lines represent the actual total loss values, whereas a darker orange line was obtained after smoothing with a factor of 0.6. Nonetheless, a crucial observation is that the loss values decreased throughout the training process. As the values of the classification loss approach zero, the classification accuracy is high and the detector performance is efficient. In addition, the localization loss curve in Figure 4.5 illustrates the predicted bounding box that matches the ground-truth bounding box. As the loss value decreased, there was less error in the action detection, and the Intersection over Union increased, indicating that the action detection was in the correct direction.

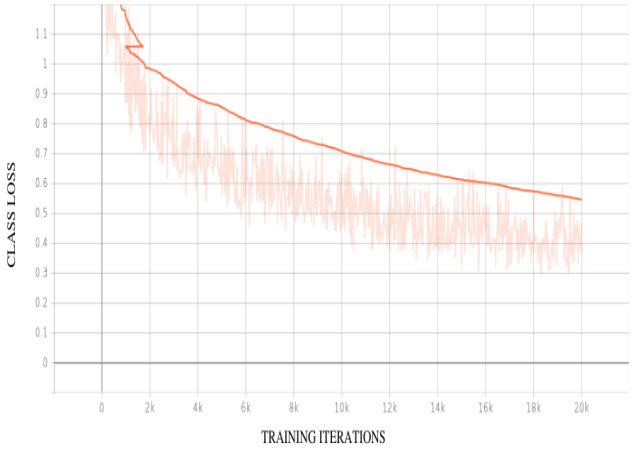

Figure 4: 2 Decline of Total Loss when Fine-Tuning SSD-MobileNet

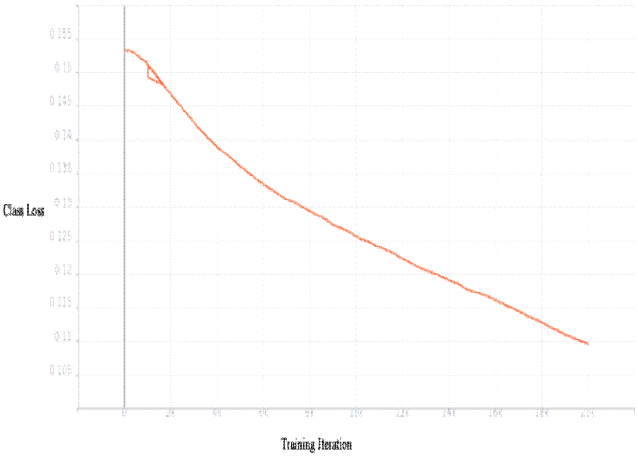

Figure 4: 3 Decline of Regularization Loss when Fine-Tuning SSD-MobileNet

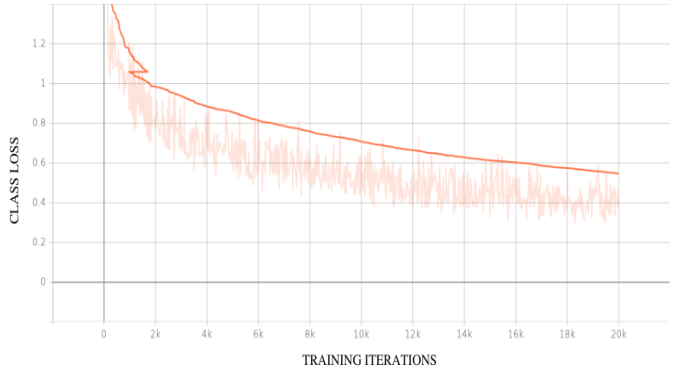

Figure 4: 4 Decline of Normalized Loss when Fine-Tuning SSD-MobileNet

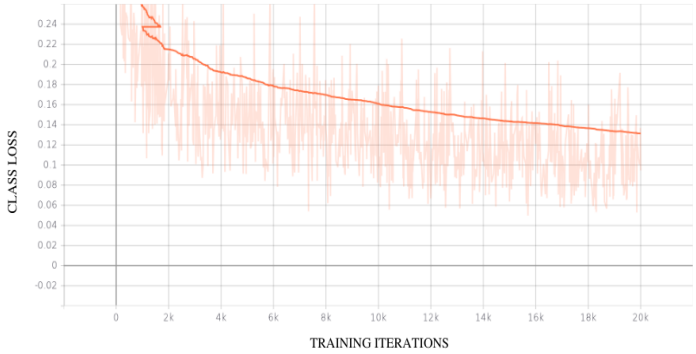

Figure 4: 5 Decline of Localization Loss when Fine-Tuning SSD –MobileNet

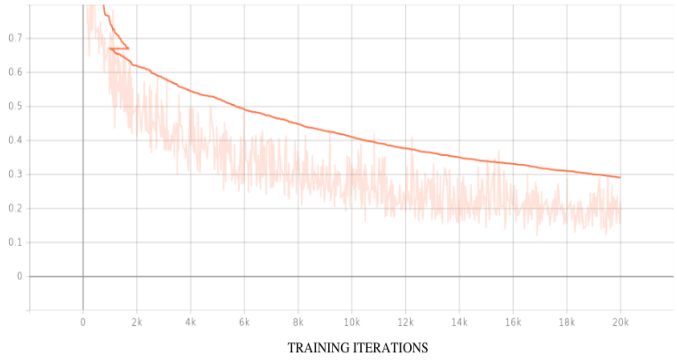

Figure 4: 6 Decline of Classification loss when Fine-Tuning SSD-MobileNet

Figure 4: 7 Steps Per Sec

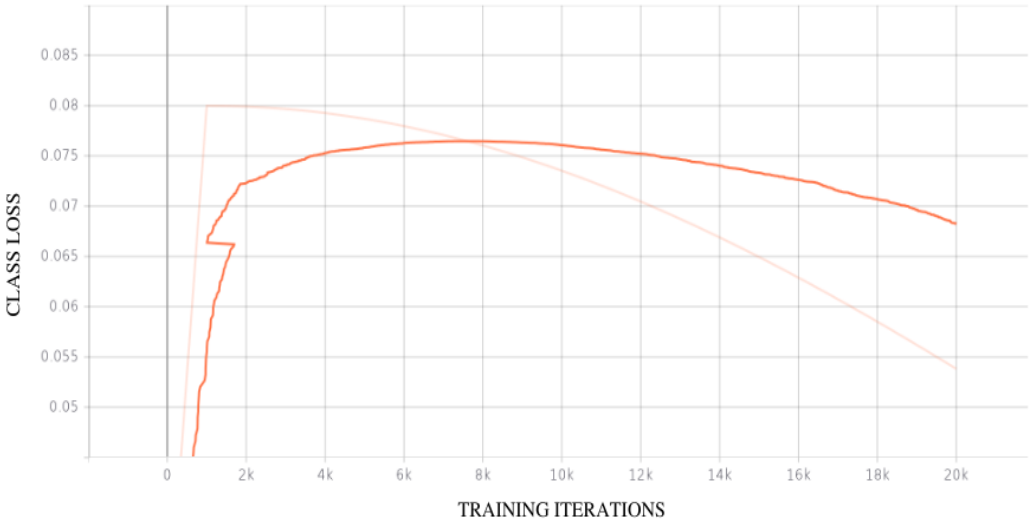

Figure 4: 8 Learning Rate

4.3. Validation and Making Predictions

We validated our trained models and made predictions by signing words in front of a camera. The checkpoint of the trained model was loaded, and OpenCV was used to obtain the signs made in real time in front of the camera, depending on the sign that OpenCV displays on the screen, the word in the text, and the percentage

accuracy on the GUI. Note: When the script is run it takes couple of minutes before the GUI display because of the loading time of the trained model. Figure 3.9 an example

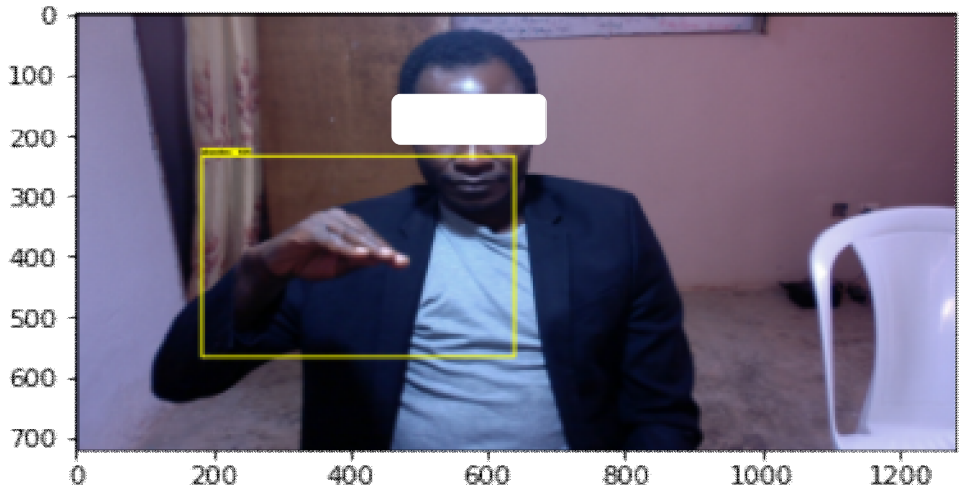

Figure 3: 9 Display GUI

The developed system can detect Nigerian sign language alphabets in real time. The system was created using TensorFlow object detection API. The pre-trained model taken from the TensorFlow model zoo was SSD MobileNet v2 320 × 320. It was trained using transfer learning on the created dataset, which contained 1608 images in total, with 60 to 80 images for each word. The total loss incurred during the last part of the training at 20,000 steps was 0.556107, localization loss was 0.12637174, classification loss was 0.33312747, regularization loss was 0.0966078, and learning rate was 0.0538146.

hello	basics	hate	walk	basket	beef	bell	bathroom	bedroom	beginner
70%	90%	80%	76%	50%	49%	90%	65%	89%	78%
thanks	iloveyou	bath	become	begin	yes	a	beauty	beg	no
65%	75%	67%	60%	55%	40%	76%	87%	76%	77%
barber	work	my	barbecue	battle	friend	baptism	battery	enemy	coming
70%	72%	65%	73%	65%	51%	50%	65%	71%	80%
hi	ban	brain	bank	jesus	bag	ball	have	bad	bake
87%	62%	50%	70%	64%	82%	76%	77%	54%	55%
school	church	for	welcome	bachelor degree	go	home	are	abandon	because
78%	57%	48%	60%	90%	77%	56%	89%	67%	88%
academics	are you married	come	accept	are you hungry	bachelor	accident	are you deaf	baby	book
91%	87%	89%	50%	45%	78%	80%	88%	67%	87%
class	okay	aware	abuse	you	arrest	female	i	love	able
80%	65%	56%	90%	92%	78%	78%	88%	90%	76%
male	land	are you sick	jealous	jean	jewellery	join	joke	judge	about
67%	88%	56%	89%	67%	89%	59%	78%	59%	89%
jot down	jubilation	keep it	kidnap	killed	kid	she is	kind	kindness	absent
78%	49%	50%	77%	78%	89%	70%	67%	71%	87%
king	queen	backup	above	jail	jesus christ	new	keep	he is	kingdom
88%	67%	69%	89%	88%	78%	80%	79%	59%	90%

Table 3:1. Confidence Rate of Each Word

The results of the system were based on the confidence rate, and the average confidence rate of the system was 72.63%. For each word, the confidence rate is recorded and tabulated in the result as shown in Table 4.1. The confidence rate of the system can be increased by increasing the size of the dataset, thereby increasing the recognition ability of the system. Therefore, the results of this system could be improved and enhanced. Despite the small size of the dataset, the proposed system achieved an average confidence rate of 72.63%.

When making real-time predictions, we discovered that the model misclassified some of the words. This is logical because we trained the model using only a small dataset. To create a more generalized and robust prediction, more datasets are required, which is one of the limitations of AI.

5. Conclusion

Communication is a vital means for exchanging ideas, thoughts, feelings, and information among individuals. However, this can be challenging for hearing- and speech-impaired people because of their disabilities. Sign language serves as the primary mode of communication between individuals, and hearing/speech is abled. Unfortunately, not everyone is proficient in sign language, which restricts their communication. The field of sign language communication is still being explored and promising results are emerging. This study utilized the TensorFlow object detection API to develop a real-time communication system for hearing/speech impaired and normal individuals. The neural network was trained using the Nigerian sign language word dataset. The system detects sign language in real time and converts signs to audio and audio to text, which are then displayed on a graphical user interface (GUI). For data acquisition, the images were captured using an iMac Time HD camera. Although the quality of the images was not optimal compared to that of high-definition cameras, Python and OpenCV libraries made capturing the images easier and more affordable. The system achieved an average confidence rate of 72.63%. Although the model demonstrated a high level of accuracy, it was trained by using a limited dataset.

In conclusion, the developed system shows great potential for enhancing the communication between hearing/speech-impaired and normal individuals. However, further research and training on larger datasets are necessary to improve the performance.

References

1. World Health Organization. (2021).
2. Salami, H., Abolarinwa, J. A., Salihu, B., David, M., & Enenche, P. (2018). Intelligent sign language recognition using image processing techniques: a case of Hausa sign language.
3. Ajavon, P. (2023). "A sign language for Nigeria," Paulina Ajavon, Accessed.
4. Shinde, S. S., Autee, R. M., & Bhosale, V. K. (2016, December). Real time two way communication approach for hearing impaired and dumb person based on image processing.
5. Taskiran, M., Killioglu, M., & Kahraman, N. (2018, July). A real-time system for recognition of American sign language by using deep learning.IEEE.
6. Thanasekhar, B., Kumar, G. D., Akshay, V., & Ashfaaq, A. M. (2019, December). Real Time Conversion of Sign Language using Deep Learning for Programming Basics. In *2019 11th International Conference on Advanced Computing (ICoAC)* (pp. 1-6). IEEE.
7. Konstantinidis, D., Dimitropoulos, K., & Daras, P. (2018, June). Sign language recognition based on hand and body skeletal data. In *2018-3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video* (3DTV-CON) (pp. 1-4). IEEE.
8. Dimitropoulos, K., Barmpoutis, P., Grammalidis, N. (2017). "Higher Order Linear Dynamical Systems for Smoke Detection in Video Surveillance Applications," IEEE Transactions on Circuits and Systems for Video Technology, vol. 27, no. 5, pp. 1143-1154.
9. Adithya, V., & Rajesh, R. (2020). A deep convolutional neural network approach for static hand gesture recognition. *Procedia computer science, 171*, 2353-2361.
10. Papastratis, I., Dimitropoulos, K., Konstantinidis, D., & Daras, P. (2020). Continuous sign language recognition through cross-modal alignment of video and text embeddings in a joint-latent space. *IEEE Access, 8,* 91170-91180.
11. Daroya, R., Peralta, D., & Naval, P. (2018, October). Alphabet sign language image classification using deep learning.
12. Das, A., Gawde, S., Suratwala, K., & Kalbande, D. (2018, January). Sign language recognition using deep learning on custom processed static gesture images.
13. He, S. (2019, October). Research of a sign language translation system based on deep learning. In *2019 International conference on artificial intelligence and advanced manufacturing (AIAM)* (pp. 392-396). IEEE.
14. Huang, J., Zhou, W., Li, H., & Li, W. (2019). Attention-based 3D-CNNs for large-vocabulary sign language recognition. *IEEE Transactions on Circuits and Systems for Video Technology, 29*(9), 2822-2832.
15. Bantupalli, K., & Xie, Y. (2018, December). American sign language recognition using deep learning and computer vision.
16. Jalal, M. A., Chen, R., Moore, R. K., & Mihaylova, L. (2018, July). American sign language posture understanding with deep neural networks.
17. Hossen, M. A., Govindaiah, A., Sultana, S., & Bhuiyan, A. (2018, June). Bengali sign language recognition using deep convolutional neural network.
18. Khan, S. A., Joy, A. D., Asaduzzaman, S. M., & Hossain, M. (2019, April). An efficient sign language translator device using convolutional neural network and customized ROI segmentation.
19. Hossain, S., Sarma, D., Mittra, T., Alam, M. N., Saha, I., & Johora, F. T. (2020, July). Bengali hand sign gestures recognition using convolutional neural network.
20. Aich, D., Al Zubair, A., Hasan, K. Z., Nath, A. D., & Hasan, Z. (2020, July). A deep learning approach for recognizing bengali character sign langauage.

21. Hasan, M. M., Srizon, A. Y., & Hasan, M. A. M. (2020, June). Classification of Bengali sign language characters by applying a novel deep convolutional neural network.
22. Sajanraj, T. D., & Beena, M. V. (2018, April). Indian sign language numeral recognition using region of interest convolutional neural network.
23. Rao, G. A., Syamala, K., Kishore, P. V. V., & Sastry, A. S. C. S. (2018, January). Deep convolutional neural networks for sign language recognition.
24. Sruthi, C. J., & Lijiya, A. (2019, April). Signet: A deep learning based indian sign language recognition system.
25. Chhabria, K., Priya, V., & Thaseen, I. S. (2020, February). Gesture recognition using deep learning. In *2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)* (pp. 1-4). IEEE.
26. Patil, U., Nagtilak, S., Rai, S., Patwari, S., Agarwal, P., & Sharma, R. (2020). Sign language translator using deep learning. *Easychair*, (2541).
27. Srivastava, S., Gangwar, A., Mishra, R., Singh, S. (2022). "Sign Language Recognition System using Tensor Flow Object Detection API".
28. Nguyen, H. B., & Do, H. N. (2019, April). Deep learning for american sign language fingerspelling recognition system.
29. Rajan, R. G., & Leo, M. J. (2020, February). American sign language alphabets recognition using hand crafted and deep learning features. In *2020 International conference on inventive computation technologies (ICICT)* (pp. 430-434). IEEE.
30. Tolentino, L. K. S., Juan, R. S., Thio-ac, A. C., Pamahoy, M. A. B., Forteza, J. R. R., & Garcia, X. J. O. (2019). Static sign language recognition using deep learning. *International Journal of Machine Learning and Computing, 9*(6), 821-827.
31. Gupta, D., Mohanty, J. P., Swain, A. K., & Mahapatra, K. (2019, December). AutoGstr: Relatively Accurate Sign Language Interpreter. In *2019 IEEE International Symposium on Smart Electronic Systems (iSES)(Formerly iNiS)* (pp. 322-323). IEEE.
32. Bachani, S., Dixit, S., Chadha, R., & Bagul, A. (2020). Sign language recognition using neural network. *International Research Journal of Engineering and Technology, 7*(4), 583-586.
33. Park, H., Lee, J. S., & Ko, J. (2020, January). Achieving real-time sign language translation using a smartphone's true depth images.
34. Kurhekar, P., Phadtare, J., Sinha, S., & Shirsat, K. P. (2019, April). Real time sign language estimation system.
35. Odartey, L. K., Huang, Y., Asantewaa, E. E., & Agbedanu, P. R. (2019, August). Ghanaian sign language recognition using deep learning.
36. Saleh, Y., & Issa, G. (2020). Arabic sign language recognition through deep neural networks fine-tuning.
37. Ahmed, S. T., & Akhand, M. A. H. (2016, December). Bangladeshi sign language recognition using fingertip position.
38. Pathak, A., Kumar, A., Priyam, P., Gupta, P., Chugh, G., Awasthi, K., ... & Jain, L. (2022). Real time sign language detection.
39. Bohra, T., Sompura, S., Parekh, K., & Raut, P. (2019, November). Real-time two way communication system for speech and hearing impaired using computer vision and deep learning.
40. Mustamo, P. (2018). *Object detection in sports: TensorFlow Object Detection API case study* (Bachelor's thesis, P. Mustamo).
41. Mache, S. R., Baheti, M. R., & Mahender, C. N. (2015). Review on text-to-speech synthesizer. *International Journal of Advanced Research in Computer and Communication Engineering, 4*(8), 54-59.
42. Yoon, H., Lee, S. H., & Park, M. (2020). TensorFlow with user friendly Graphical Framework for object detection API.
43. Wan, L., Zeiler, M., Zhang, S., Le Cun, Y., & Fergus, R. (2013, May). Regularization of neural networks using dropconnect. *In International conference on machine learning* (pp. 1058-1066). PMLR.
44. Alakh, S. (2020). "Build your Own Object Detection Model using TensorFlow API".
45. Al-Azzoa, F., Taqia, A. M., & Milanovab, M. (2018). Human related-health actions detection using Android Camera based on TensorFlow Object Detection API. *International Journal of Advanced Computer Science and Applications, 9*(10).
46. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016, September). Ssd: Single shot multibox detector. In *European conference on computer vision* (pp. 21-37). Cham: Springer International Publishing.
47. Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. [Online]. Available:
48. Matthijs, H. (2021). "MobileNet version 2".

Copyright: *©2025 Okorie Emmanuel O, et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.*