Implementing Computer Vision Techniques to Recognize American Sign Language (ASL) Hand Signals

Alvaro Martin Grande[a,1,], Rodrigo E. Ayala[b], Stuart Isteefano[c], Tauheed Khan Mohd[d]

[a] *Department of Math and Computer Science, Augustana College, Rock Island, IL, USA*
[b] *Department of Math and Computer Science, Augustana College, Rock Island, IL, USA*
[c] *Department of Math and Computer Science, Augustana College, Rock Island, IL, USA*
[d] *Department of Math and Computer Science, Augustana College, Rock Island, IL, USA*

Abstract

American Sign Language is a popular language for deaf individuals. Communication is made easier for these people through sign language. However, in a digital era like today, there is a need for these people to be able to communicate online, and even get help from technology to communicate in person with non sign language speakers. This research will present a program able to translate American sign language to plain English. This study aims to use the OpenCV library to recognize hand signals, also a trained model to identify images so that the program can then translate them to words and letters. The program uses a data set of over 2000 images which will be in this case the largest data set available. With over 90% of accuracy it results in a basic computer program with the largest data set available that would make possible for users to communicate with a wide variety of words and expressions.

Keywords:
Datasets, Neural Networks, Hand Detection, Text Tagging

1. Introduction

American sign language, also known as ASL, is the most used form of communication for the deaf in the United States. American sign language uses approximately 6000 different gestures for common words and finger spelling for other nouns, etc. [1]. Individuals must invest considerable time and effort to communicate with other individuals. Additionally, different sign languages exist for each region of the world, making it more challenging. With advances in modern computer vision technologies, real-time communication between the deaf and the non-deaf is no longer just a thought. Many programs are available that allow for small communication between the two parties; however, they are still being improved by the day; we are inching closer towards complete software that implements and can translate the entire sign-language vocabulary. A concurrent translating software program will open a plethora of opportunities for the deaf community, who struggle to land and obtain full-time jobs.

Communication between a deaf individual and a non-ASL-speaking person can be nearly impossible; even if they are familiar with the language, it still may be difficult to understand if they are not fluent in sign language.

Email addresses: alvaromartingrande20@augustana.edu (Alvaro Martin Grande), rayalaguerrero18@augustana.edu (Rodrigo E. Ayala), stuartisteefanos17@augustana.edu (Stuart Isteefano), tauheedkhanmohd@augustana.edu (Tauheed Khan Mohd)

The purpose of ”teaching” computers sign language using computer vision is to enable deaf individuals further to communicate with the world around them. If our goal as a society is to better include and communicate with the disabled, implementing computer vision could create a real-time interpreter [2]. This study was used to determine the effectiveness of an AI system in reducing the communication barriers between the deaf and the gesture recognition system. Seven participants evaluated the system, and all concluded that the system was effective and efficient. It was also stated that the participants were motivated and desired to be involved with the system. [2]. These systems could give the disabled new opportunities that they may not have had before. According to the Yang-Tan Institute at Cornell University’s analysis of 2016 American Community Survey data, fewer than 40 percent of deaf individuals work full time. Lacking the ability to communicate makes it difficult for these individuals to land jobs; even with improvements in technologies and accommodations, employers are still unwilling to take the risk. Therefore, having a real-time interpreter who could immediately communicate with others could drastically improve the opportunities for those with hearing disabilities.

Research has shown that the framework for a computerized real-time ASL interpreter is there. Many studies and projects have been conducted in the last few years to work on this solution; one specific case proposed a real-time hand-gesture based recognition system based on the American Sign Language (ASL) dataset and capturing data through a BGR webcam and processing it using Computer Vision (OpenCV). This project used 29 different gestures from the ASL language, and the goal was to use OpenCV to train the dataset to recognize these gestures. This study showed a 99.999 percent success rate, consisting of over 87,500 RGB samples. The following steps moving forward and the basis of our research is to further expand on the gestures and create coherent sentences and words using the entire American sign language library. We would then like to develop on this matter and implement the different languages used across the world in our project. While we recognize it would be a near-impossible task to include all 300 of the universal sign languages in our project, barring the resources available (Sign solutions). Our goal is to implement as many as possible, given our time frame.

This project aims to create a meaningful product that uses technologies with vast research potentials, such as OpenCV in this case and the entire computer vision approach for new applications. With computer vision, you can manipulate the software to make it identify faces, objects, hands, etc. The idea was to create a program capable of recognizing and translating American Sign Language. This approach is really using the potential of computer vision technology to help people that have the necessity to communicate with ASL. Most computer programs and applications don’t have support for sign language. Most of the time, we can see sign language translations in some broadcast events, but it is not a common practice. Nowadays, everything is digital and even more after COVID-19. Deaf-mute individuals have a disability that will affect their interpersonal relationships, not just in person with other people. But also online, such as in video calls, online classes, etc. Some type of translating program should be developed to accommodate those types of disabilities and via recognition of hand gestures and interpretation. Our motivation is to make life easier for deaf-mute people that struggle every day with life going digital and the way things are now being held for social and professional relationships. They should be able to communicate with others digitally even if other people don’t have the ability to understand or use sign language. It is indeed rare to encounter real accommodations for deaf-mute individuals in the virtual world, such as applications or social networks that are able to translate sign language and break the language barrier. In addition, technology came up to make people’s lives easier. For the general public, it has been easy to adapt but not so much for people with disabilities. They have restrictions to do certain activities that are now basic and that you could consider a necessity.

2. Related Work

In order to design interpreters for American Sign Language, it is important to be aware of how computers interact with gestures and movements. To start, the hand and the arm of the client have to be measurable so that the computer can read those inputs and process them to then interpret a specific gesture. Before vision-based detection, mechanical devices or gloves were used to directly measure arms and joints. But vision-based systems implemented recognition of static hand gestures or postures. Models were trained using store images of possible hand patterns, geometric moments, contours, silhouettes, and 3D skeletal models [3]. This is scalable not just to hand and arm gestures but also to identify facial expressions. The algorithms implemented are also used for facial recognition purposes and complemented with other actions. Several projects are being developed integrating desktop and mobile applications with computer vision technology.

OpenCV is a python library that is used for computer vision practices and human-computer interaction. The base exercise before recognizing complicated gestures is to detect finger counting. There are eight steps involved in the interpretation of finger movements and relating them to counting. It typically involves taking an image, editing it to identify contour points and boundaries of the hand, finally finding the fingertips and counting the number of fingers [4]. From this point, transitions and edits are made to not only detect fingertips and typical boundaries of a hand. Some other constraints are implemented to increase the precision with more complicated signs and sequences of gestures. Just to identify the number of fingers, the study mentioned succeeded with a 92% accuracy. However, the percentage decreases when gestures get more complicated and even with combined gestures with both hands and facial expressions. Indeed processing sign language with the help of computer vision tools can be a tough challenge; pattern recognition techniques and neural networks can be used to train models in OpenCV and build a system of American Sign Language recognition and translation in real-time [5].

There exist multiple advanced techniques to recognize hand gestures. However, our literature review points out Computer Vision and Deep Neural Networks are the most popular and effective methods [6]. Using computer vision to recognize a wide variety of hand signs such as numbers or signals from the American Sign Language alphabet requires dealing with techniques of hand segmentation, advanced classification algorithms, and large datasets [7]. Recent studies have achieved 82.55% accuracy on hand signals recognition by using Transfer Learning. Transfer Learning is a common technique used in Deep Learning. Neural networks are pre-trained, and weights are established. Overall, the best result showed an accuracy of 94.44% by using these networks [6].

Segmentation algorithms used in Computer Vision attempt to detect hand features such as skin color, appearance, motion, depth, or skeleton. Implementation of tracking techniques highly improves the accuracy and performance of these techniques [7]. Nevertheless, segmentation presents critical limitations: a high-resolution video camera must be connected to a machine, and the subject must stand close to this one. Therefore, to create functional Computer Vision algorithms - such as the ones developed in the popular python library OpenCV - the next factors must be taken into consideration: color recognition, appearance recognition, motion recognition, skeleton recognition, depth recognition, 3D-model recognition, and deep learning recognition [7].

The American Sign Language alphabet is composed of dynamic and static gestures. Dynamic gestures require motion and therefore must be videotaped, including various angles and perspectives. On the other hand, static hand signals can be photographed. A rich ASL dataset must be composed of both types of gestures. Although, there exist Human-Computer Interaction projects which implement wearable glove-based and camera vision-based sensors [8]. These gloves use various sensors to capture hand motion and position, as well as to detect the correct location of the fingers and the palm. This device must be constantly connected to a computer. While in this case, large datasets are not required, rigorous classification algorithms must be implemented. Glove-based sensors have clear limitations: this technology may be uncomfortable for older people and children and challenging to be used by individuals suffering from different disabilities.

Utilizing large video datasets to implement OpenCV segmentation algorithms on training photage is a recurrent strategy widely used in Computer Vision. The combination of these two fields, Computer Vision and Deep Learning, appear to be the most powerful and promising approach to identify hand signals correctly. There exist various American Sign Language dictionaries which pictorially describe each of the signals and combinations in the language. As an example, the American Sign Language Hand shape Dictionary organizes multiple signs based on the initial hand shape. However, it isn't easy to navigate this dictionary to find specific signs among 1600 that were included. Therefore, the ASL dataset must be versatile, providing quick access to thousands of signs. Building video classifiers and caption generators for comprehensive sets of ASL signs must include instant sign recognition as their key element.

The dataset used by Athitsos et al. contains at least one video example per sign from a native singer for 3000 signs. The signers were recorded from different locations. Side wise and face wise, the videos were captured at 60 frames per second at a resolution of 640x480 pixels per frame. Frontally, videos were captured at 30 frames per second at a resolution of 1600x1200 pixels per frame. High-resolution videos facilitate hand-pose tracking and finger recognition. Every hand signal was labeled by Athitsos' team since there are no exact written transcriptions associated between English and American Sign Language. These approximations are named "glosses." This dataset can be found at: www.bu.edu/asllrp/lexicon.

Additionally, as previously mentioned, the usage of convolutional neural networks is required to process real-time video data from a sign language conversation into the English language. The implementation of a such a neural network is challenging due to several reasons: environmental concerns (lighting sensitivity, background), occlusion (hands or finger out of the camera view), sign boundary detection (when a sign ends and the next one begins), co-articulation (sign is affected by the preceding or succeeding sign) [9]. Deep Neural Networks have been trained and applied to American Sign Language alphabet recognition with accuracy consistently over 90%. As previously mentioned in this proposal, users must access a specific framework to utilize this device. In order to create a tool useful for users around the world, these ones must be connected to the Internet, and the Deep Neural Network must be allocated in a web server constantly receiving and outputting data. In addition, transfer learning plays an important role: various neural networks have been trained and tested with multiple datasets.

Moreover, pre-trained neural networks such as GoogleNet and Caffe could be used to achieve this objective [10]. Following this idea, we propose to utilize and create various convolutional neural networks of multiple layers to study the behavior of these ones when Athitsos' dataset is passed. However, [10] manifest the importance of the differences between the training data images of these pre-trained networks, and our data. It was critical for them to test the effectiveness of altering a variety of pre-training weights at different depths. They initialized the GoogLeNet network as well as they reinitialized all the classification layers with Xavier and properly adjusted the dimensions to match the data size. Various learning rates for different layers were changed. It can be increased how the validation accuracy increases at the same rate that the number of epochs grows. Validation accuracy does not reach 0.5 unless, at least, one epoch has been trained. The results can be found on the next figure:

Furthermore, other authors have built complex datasets based on images instead of videos, and built their own convolutional neural networks such as using 224x224 images [11]. Later on, these images were normalized and transformed into NumPy vectors of 32x32x2. As it can be observed, the size of the images was extremely reduced and the RGB channels were kept. In order to train the model, 61,614 images from the dataset were used, with around 2200 for each class. For the validation process, 18,480 images were used. Tested the accuracy for each letter of the ASL alphabet. The letter I obtained the highest accuracy of 99.57%. An accuracy of 99.99% was obtained for the letter. However, the lowest accuracy was 97.32% and 97.37%, for letter M and N, respectively.

3. Experimental Setup

This research was done using a method known as transfer learning and data augmentation that allowed the creation of a deep learning model that represented some of the ASL dictionaries. Transfer learning is a research method that is implemented via storing knowledge. This knowledge is gained while solving problems and applying them to different problems with similar structures. This machine learning style tries to emulate human learning by using their experiences to solve relevant problems. ”Human learners appear to have inherent ways to transfer knowledge between tasks. That is, we recognize and apply relevant knowledge from previous learning experiences when we encounter new tasks. The more related a new task is to our previous experience, the more easily we can master it.” [12]. The closer machine learning gets to imitating human learning, the more successful the research will be.

Data augmentation was also used in this research proposal. Data augmentation is a technique that is used to increase the amount of data by adding similar copies of existing data that are slightly modified. This helps to reduce the amount of over fitting when training a data model. According to the University of Cape town Computer Science department research, ”Data augmentation overcomes this issue by artificially inflating the training set with label preserving transformations. Recently there has been extensive use of generic data augmentation to improve Convolutional Neural Network (CNN) task performance.” [13]. These Machine learning methods helped improve the model's accuracy and created a dependable dataset of the ASL language.

The dataset used was trained in Kaggle and composed of over 12,000 videos focused on American sign language signs. The specific dataset used was the WLASL set which recognizes over 2,000 known ASL words. In order to train the model to be a better fit for real-world recognition, the dataset was trained using different brightness scenarios, ranging from twenty percent lighter/to darker, and was zoomed in and out up to 120 percent to represent the real-time use better.

After training the model, it is then opened and loaded in an IDE; in this specific case, PyCharm was used. OpenCV was also used for the video feed to be detected; code was implemented to be able to detect hand movements in the video feed using OpenCV. The hand signals are then presented to the camera and are recognized by the model. This process is done by capturing the signs in a frame and then processing the specific frame through the model; the model then predicts the sign. There are different possibilities, and the model predicts it at different rates, Low confidence, Medium confidence, and finally, high confidence. The program has a specific threshold that needs to be met for a word to be produced, the word is produced, and an accuracy rating is given along with the word [14].

4. Framework

The general outline for this project to be feasible will consist of constantly adding features and images to recognize numbers, letters, words, and then combinations of gestures to construct and translate full sentences. OpenCV will be the Python library that will be used for the computer vision aspect of the project. We might also employ some other libraries or databases in the future to store all the information required to pair hand gestures to sign language. The mediapipe module from OpenCV will be required to make the computation to identify hands through the webcam of the computer. The way that this module works is through a detectHand() method that gathers data of over 30,000 images from hand gestures. To identify different parts of the hand it implements 3D modeling and coordinates over the axis of the camera.

The module creates a 3D model based on images that creates points and coordinates that form a hand that is easily identifiable. It divides the structure of the hand in different parts that are numbered. There are 20 points that conform to the structure of the hand, this is helpful to recognize different gestures even if your fingers and joints are not in a flat and static position. If we run the code using the methods from the mediapipe modules, a list of coordinates will be printed into the console with x, y, and z coordinates. The z coordinate represents the depth and this is very important in cases where some fingers are overlapping the others or some parts of the hand. To increase accuracy, the module renders a significant amount of synthetic hand models and then they map it to the exact coordinates of the hand. In the image below we can see some examples of how the outline shapes the hand perfectly and the adaptability that it has. This is possible because of the depth recognition of the 3D models and how it can recognize even the fingers that are hidden and overlapped. With that being said, these models will be paired with images of commonly used gestures of ASL, so that it will be able to recognize such patterns. After that we are interested in translating each gesture into English. It can be possible to create in the camera window a dedicated space for the translations that will update sentences and words in real time as the gestures are changing. The final step will be to use that module to recognize face landmarks and associate them with the gestures to create more complex sentences. At the end the output of our program will be a sign language translator that virtually translates ASL [15].

5. Methods

In this section, we will discuss the different approaches utilized throughout the project. While some approaches were more successful than others, all of them were extremely useful to obtain more accurate results and improve the predictions of our code. Therefore, we believe it is not only necessary to test various methods, but it is enriching since it gave us the opportunity to find the weaknesses and strengths of our project.

The methods used will be described chronologically, from the beginning to end, explaining their respective importance.

5.1. Sign Language Recognition (MediaPipe) – Basic Neural Network Model

The first strategy implemented to recognize hand gestures and signals utilized the mediapipe library and TensorFlow. MediaPipe is a popular framework in the Artificial Intelligence field (AI) created to build machine learning pipelines for processing time-series data like video or audio. This cross-platform framework works in Desktop/Server, Python, Android, iOS, and embedded devices like Raspberry Pi and Jetson Nano. It is important to highlight this framework belongs to OpenCV (most popular Computer Vision package in Python). Additionally, TensorFlow is a well-known software library for machine learning and artificial intelligence commonly used to design, build, and train deep neural networks.

To initially run this program, it was necessary to install the latest version of TensorFlow, MediaPipe and OpenCV. This script could perform segmentation techniques by using TensorFlow tensors on real-time video. The mediapipe detection algorithm takes an image (raw data from the laptop's webcam) plus a "pkl" model. The image is passed as OpenCV object, as well as the "pkl" model. This function processes the images and returns the results of the detection algorithm plus the image. Secondly, the draw styled landmarks function draws the detected landmarks from the previous function on the real-time video. It overwrites the image in real-time to place the segments over the detected hands. Finally, this script uses a speak function that uses the machine's speaker to "read out loud" the detected word/sentence.

5.2. Sign Language Recognition (MediaPipe) – Basic Neural Network Model + Sentece Recognition

This script presented several weaknesses. Firstly, it could only detect words, one by one. Secondly, the program constantly crashed after a few detections. Thirdly, the model did not contain sufficient words for the script to be useful.

Therefore, the first fix was to put together several detected words into "sentence" arrays, and to output that as a result. The only problem with this strategy is that a hand signal to finish a sentence (like a punctuation symbol was needed). Secondly, we were able to fix that bug by making a few changes on the script to make it compatible with MacOS. Thirdly, we added more features to the model. These new words were fed to the model in form of images and videos. Finally, the model was retrained and its performance clearly improved. Now, the model can detect a wide variety of hand signals.

5.3. WLAS (World Level American Sign Language) Video Dataset from Kaggle

Secondly, a more recent dataset was found in Kaggle. Kaggle is a platform sponsored by Google and TensorFlow to manage and distribute datasets and practice machine learning and deep learning with these ones. This World Level American Sign Language set counts with over 2000 gestures. These gestures were recorded and compiled in mp4 format. To use this dataset and train a neural network model, we successfully upload it to Google Collab. Once the dataset is ready to be used, we converted into multiple pandas' data frames for easier manipulation. Lately, we divided this dataset into three different categories: test set, training set and validation. Our model will be trained over the training set (largest set out of the three). Moreover, the new-learnt weights of the neural network will be tested with the testing set. Finally, to verify the accuracy and performance of our neural network architecture, we will use the validation set to double check the results.

5.4. Sign Language Convolutional Neural Network Framework

This framework uses a dataset of 44 hand gestures by using OpenCV and TensorFlow. While this dataset is considerably smaller, the training time is extremely faster. Additionally, by using this framework, the data extraction and training by the convolutional neural network is up to ten times faster. For each gesture, 1200 50x50 pixels images were captured. These images were in grayscale to optimize the training time. Additionally, some of these images were rotated and slightly modified to increase the size of the set, and therefore, obtain better accuracies after the training process [16].

Furthermore, by using TensorFlow and Keras, a Convolutional Neural Network was created. This neural network shares similarities with the classical MNIST architecture. Multiple layers were added to the model to handle the data and improve the training process. Finally, the model is exported out of Keras (and Google Collab/ Google Cloud Computing Platform) to perform some live hand-gestures recognition over some videos [17].

Finally, after obtaining the highest accuracy for both of the models using their respective datasets, we expect to compare their accuracy, training time and performance results [18].

5.5. A Real-Time System For Recognition Of American Sign Language By Using Deep Learning.

Investigators at Yildiz Technical University have developed Convolutional Neural Networks (CNN) capable of classifying 28 pixels by 28 pixels images of hand signals and gestures. This CNN consists of an input layer, two 2D convolutional layers (images are processed in grayscale), and additional layers for pooling and flattening. Softmax-based loss functions, as well as Rectified Linear Unit (ReLU) activation functions, are used for this neural network [19]. This neural network has real-time applications capable of successfully detecting hands' bounds, skin, and colors by using multiple algorithms. Open CV is used to calculate convex hull and convexity detections. Different polygons are shaped over the hands and fingers denoting the region of interest. Then, the algorithm picks up on these regions and classifies the hand gesture [19].

The dataset used by Yildiz researchers was originally created by Massey University, This dataset counts 900 images including 25 samples for each of 36 characters consisting of 26 letters and 10 numbers in the dataset. After multiple training processes of the neural network, it reached an accuracy of 98.05 on successfully classifying hand gestures and signals from the dataset [19].

5.6. American Sign Language Recognition System Using Wearable Sensors with Deep Learning Approach.

Researchers at Keimyung University developed smart wearable sensors with Bluetooth capabilities and battery modules to be adapted to users' hands. IMU sensors were placed on each of the fingerprints of the user's hand to record figures and hand movement data. The data collection process consisted of multiple subjects signing various hand signals and gestures in American Sign Language (ASL). IMU sensors are crucial to detect movement and orientation since some hand signals are similar. Several fingerspelling words were recorded, as well. These sensors recorded and computed the next parameters: acceleration, rotational angle, orientation, and quaternion (the quotient of two directed lines or vectors in a 3D space). This data was normalized based on mean and standard deviation features in order to reduce processing time and data complexity.

A long short-term memory classification model (LSTM) was used for this data. Thai recurrent neural network introduces functional gates and sigmoid functions to successfully decided what information to pass through its layers. This model counts with six hidden layers, two drop-out layers, and three dense layers. ReLU and Softmax activation functions were used with a learning rate of 0.001. The accuracy obtained from the neural network was 99.67 in predicting the values for the mean and standard deviation for the gestures. The algorithm did a slightly better performance in predicting the standard deviation of the data. While most of the data sensors' predicted accuracy turned out to be low, the gyroscope predicted values turned out to have an accuracy of 96.95 [**?**].

5.7. Classification of Sign-Language Using MobileNet.

Multiple datasets obtained from Kaggle were converged for the training phase, and from 43,500 to 87,000 images were used [20]. This data contains 29 different classes containing all the letters of the alphabet. To process all this data, a MobileNet neural network was utilized for data detection and classification [20]. This complex neural network is constituted of multiple layers to process images of 64 pixels by 64 pixels, all in RGB color. The learning rate of the model was 0.0001, and softmax was used as the activation function, as well as the adam optimizer [20]. The accuracy of the model was 95.41 by using a MobileNet pre-trained model. The model was finally validated with the remaining data from the training vast dataset [20].

5.8. Deep Learning for American Sign Language Fingerspelling Recognition System.

Researchers at Vietnam National University utilize the Massey database to train and test their models. This dataset includes 2525 images of static alphabetical hand signals and gestures from a to z. Their model first performs a skin segmentation of the image, then crops and resizes this one, to later adjust weights, and eventually train the classifier. Researchers use more specifically a HOG feature descriptor combined with an SVM classifier to detect hand signals [21]. This algorithm generates gradients with a magnitude and orientation over the image to detect changes in hand gestures [21]. A histogram of gradients is generated corresponding to each image cell. normalization is applied to make the algorithm less sensitive to light changes. Additionally, an LBP (grayscale invariant texture operator) generates multiple histograms after normalizing blocks of cells [21].

A Support Vector Machine (SVM) is one of the machine learning algorithms utilized by researchers. SVMs attempt to generate optical decision boundaries in order to maximize the margin between two different classes. Three different kernels are used in this paper: Radial Basis Function (RBF) kernel, Polynomial kernel, and Linear Kernel for the multi-class SVM classifier. Moreover, the second model utilized for this experiment is a CNN architectured with multiple dropout layers to reduce overfitting. Data augmentation is performed to obtain better results [21]. Cross-validation is applied to analyze the results of the various models on the dataset. Input images of 150 pixels by 150 pixels, RGB color, are sued to train the different models. HOG-SVM and LBP-SVM get an accuracy of 97.49 and 98.23, respectively [21]. On the other hand, HOG-LBP-SVM and CNN reached an accuracy of 98.36 and 97.08, respectively [21].

5.9. Evaluation of Deep Learning based Pose Estimation for Sign Language Recognition.

Researchers at the University of Texas used the American Sign Language Image Dataset (ASLID), with images extracted from Gallaudet Dictionary Videos and the American Sign Language Lexicon Video Dataset [22]. This rich dataset counts over 1000 examples of hand signals and gestures. Annotations for arms are joints were provided by the researchers and successfully included in the dataset [22].

The model proposed by the researchers includes transfer learning to optimize the algorithm uses and get higher accuracies. By using deep learning techniques, the researchers evaluated single frames from the dataset videos to estimate concrete poses and gestures on the images. This neural network classifies context body joints (previously annotated) [22]. A quantitative evaluation measure is used for joint detection of seven upper-body joint locations. If the distance established between the detected joint and the ground truth is less than a threshold, the estimation will be determined to be correct [22]. By using this technique, researchers obtained multiple accuracies for the different joint groups (head, hands, shoulder, and elbow) [22].

5.10. Deep learning in vision-based static hand gesture recognition.

The dataset utilized by researchers at University contains 24 different hand gestures. These hand gestures are signed over a light black background for easier recognition and classification. These images are in grayscale and have a size of 248 pixels by 248 pixels [23]. They will be later prepared for segmentation. Additionally, they are filtered using an algorithm to segment hand-occupied regions [23].

The models are trained on 1440 processed hand gesture images with a segmented size of 32 pixels by 32 pixels, in grayscale [23]. Multiple convolutional neural networks (CNNs) are trained with various architectures: two hidden layers (CNN1), three hidden layers (CNN2), and four hidden layers (CNN3) [23]. Additionally, as more layers are stacked, different autoencorders such as stacked denoisining sutoencoders (SDAEs) of multiple depths are trained to observe any improvement in performormance: SDAE1 has 90 and 70 neurons, in the first and second hidden layers of the network, respectively. Learning rates of 0.2 and 0.4 are used for greedy layer-wise pretraining and fine-tuning, respectively, and a momentum rate of 0.5 is used as inertia for the error Gradient [23]. The architecture for SDAE2 with three hidden layers is shown below in Table 2. It will be seen that SDAE2, achieved a lower MSE compared to SDAE1. Lastly, a SDAE3 with four hidden layers is trained; the network architecture and training parameters are given in Table 2. It can be seen that SDAE3 achieved the lowest MSE out of the three trained SDAEs [23].

5.11. A Real-Time American Sign Language Recognition System using Convolutional Neural Network for Real Datasets.

This research project implemented a hand-gesture recognition CNN and a Human-Computer Interface system to communicate with the user. The multi-class recognition CNN was fed with a total of 26 alphabets [24]. The convolutional architecture used was VGG Net for high-scale image recognition. This neural network was trained to classify 28 classes of static fingerspelling in ASL. Images were normalized and scaled (224 pixels by 224 pixels) for training purposes [24]. A total of 61614 images were used for the training dataset, with more than 2000 images per each alphabet. Each image was captured in different backgrounds and with multiple lightnings. Softmax was used as the activation function, and the learning rate was 0.0001 with 500 epochs [24].

A real-time python script was built to assist the deaf-mute individuals by integrating this CNN. An accuracy of 100 was obtained for the validation set [24]. American Sign Language Character Recognition with Capsule Networks.

5.12. ASL-3DCNN: American sign language recognition technique using 3-D convolutional neural networks.

ASL Video sequences are preprocessed and then analyzed by using cascaded 3D CNNs.To successfully train these neural networks, the videos are preprocessed which reduced the chances of CNNS beijing trained on ineffective sets. Each video is converted into multiple frames, and then the color frome is transformed into gray-scale format. The illuminations variations in the fram are canceled out using histogram equalization. The size of each normalized frame is 512 pixels by 384 pixels [25].

After completing this process, the new video sequences are manually trimmed to ensure that only hand gestures and motions are present in these videos. After using a ReLU as a an activation function, the accuracy of the trained neural network is 96.0 on training data, 97.1 on testing, and 96.4 on validation [25].

5.13. Enhanced Sign Language Transcription System via Hand Tracking and Pose Estimation.

Hand tracking is a challenging task due to the difficultness of computantioanlly describing hands' shapes. Therefore, the dataset used for this experiment fixed the color of the hands and the background color was removed [26].

To predict the depth map from a single frame, the Eigen map prediction tool was used. This tool provides prediction models that consists of two deep networks: the first one predicts the input image, and the second one enhances the first prediction [26].

To estimate hand poses, more detail during the recognition process is required. Increasing the number of joins on the same frame requires better recognition techniques, and makes depth mapping a difficult task. The Zhou model was utilized to fully exploit hand model geometry enabling the researchers to obtain all joint positions in both hands [26].

These models were used on the ASL Image Dataset to evaluate the accuracy of tracking hands. This dataset contains 479 images captured from the American Sign Language Lexicon Video Dataset [26]. The accuracy obtained for hand tracking was close to 100, and the one obtained for pose estimation was between 57.3 to 91.2 [26] for various poses.

Table 1: American Sign Language (ASL) Deep Learning Classifiers

Num.	Year	Paper	Wearable	Real-time detection	Media type	Neural Network	Accuracy	Learning rate
1	2016	Evaluation of Deep Learning based Pose Estimation for Sign Language Recognition	NO	NO	IMAGE	Multipe (CNN and Transfer Learning)	N/A	N/A
2	2016	Enhanced Sign Language Transcription System via Hand Tracking and Pose Estimation	NO	YES	IMAGE	CNN	91.20	N/A
3	2017	Deep learning in vision-based static hand gesture recognition	NO	YES	IMAGE	CNN	N/A	0.2/0.4
4	2018	A Real-Time System for Recognition of American Sign Language by using Deep Learning	NO	YES	VIDEO	CNN	98.89	N/A
5	2018	MS-ASL: A Large-Scale Data Set and Benchmark for Understanding American Sign Language	NO	NO	VIDEO	CNN	N/A	N/A

S. No	Year	Paper	Wearable	Real-time detection	Media type	Neural Network	Accuracy	Learning rate
6	2019	Deep Learning for American Sign Language Fingerspelling Recognition System	NO	NO	IMAGE	Supervised Learning CNN	98.03	N/A
7	2019	American Sign Language Character Recognition with Capsule Networks	YES	YES	IMAGE	CNN	N/A	N/A
8	2020	American Sign Language Recognition System Using Wearable Sensors with Deep Learning Approach	YES	NO	IMAGE	Long Short-term Memory (LSTM)	96.95	0.001
9	2020	A Real-Time American Sign Language Recognition System using Convolutional Neural Network for Real Datasets	NO	YES	IMAGE	CNN	100	0.0001
10	2021	American Sign Language Static Gesture Recognition using Deep Learning and Computer Vision	NO	YES	IMAGE	Vision Transformer Model	N/A	N/A

S. No	Year	Paper	Wearable	Real-time detection	Media type	Neural Network	Accuracy	Learning rate
11	2021	ASL-3DCNN: American sign language recognition technique using 3-D convolutional neural networks	NO	YES	VIDEO	3DCNN	97.10	N/A
12	2022	Classification of Sign-Language Using MobileNet - Deep Learning	NO	NO	VIDEO	MobileNet (Transfer Learning)	95.41	N/A

S. No	Year	Paper	Wearable	Real-time detection	Media type	Neural Network	Accuracy	Learning rate

6. Results

This data set was trained using over 2000 videos that deciphered signals in American Sign Language. After training the data set, code was written to translate the signals into letters, words, and eventually with the latest release, fully formed sentences. By studying these videos, the model can successfully create over 1000 words and can also form over 100 sentences. When a user runs the program, they will be able to admit sign language signals into the webcam and create a plethora of different words/sentences. This software enables people to freely communicate with others through the use of a webcam. The proposed research shows room to expand the current environment surrounding real-time automated sign-language translators. There are already programs that can correctly identify certain gestures and less-complex words in some cases. There has not been a sufficient number of projects that have successfully formed complex sentences or incorporated slang, punctuation, etc. While adding on to these programs would be a great improvement, there is room for more significant expansion. There lacks a phone application that gives people real-time assistance, and in today's society, that is a necessity. People will not have the luxury of bringing a computer with a program everywhere they go; therefore, the proposed research exhibits that creating an application is not only a possibility but a necessity for expanding on this solution. The current infrastructure that many of these programs contain is sufficient for translating basic words and gestures. However, to further advance the software, sentence building and other complex grammar structures are needed for the programs to benefit society.

7. Discussion

Based on the results, the current dataset with over 2000 videos with words and expressions in American Sign Language will give this program the ability to identify a large number of words and translate them to plain English. Research has been numerous last couple of years, however the results end up being quite similar [27]. Most projects try to focus on just hand recognition using a small data set to represent the alphabet and numbers [28]. Also they don't incorporate ways for the user to actually create sentences with sign language. And with a small data set it is difficult to actually communicate with such application. Also, projects are not up to date using previous versions of neural network packages such as TensorFlow [29] and this creates added difficulties for further researchers who might want to develop a better version of the program [30]. For further research, there is a need of a better user interface for the application and eventually a mobile application that could quickly translate sign language and create a voice assistant that can translate the signals to maintain a conversation with a non sign language speaker. This could totally change the way that people with speech and hearing problems would communicate on a daily basis [31]. Also implementation of different types of sign languages could be appreciated for non-English speakers that might use different signs to communicate [32]. Indeed, it would be always important to keep updating and adding more unique vocabulary to the data set and expanding its size so that most expressions and words are covered to create any type of sentence.

References

[1] T. Starner, J. Weaver, and A. Pentland, "Real-time american sign language recognition using desk and wearable computer based video," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 20, pp. 1371 – 1375, 01 1999.

[2] C. N. Nyaga and R. D. Wario, "Sign language gesture recognition through computer vision," in *2018 IST-Africa Week Conference (IST-Africa)*, pp. Page 1 of 8–Page 8 of 8, 2018.

[3] M. Yeasin and S. Chaudhuri, "Visual understanding of dynamic hand gestures," *Pattern Recognition*, vol. 33, pp. 1805–1817, Jan. 2000.

[4] R. M. Gurav and P. K. Kadbe, "Real time finger tracking and contour detection for gesture recognition using opencv," in *2015 International Conference on Industrial Instrumentation and Control (ICIC)*, pp. 974–977, 2015.

[5] I. Culjak, D. Abram, T. Pribanic, H. Dzapo, and M. Cifrek, "A brief introduction to opencv," in *2012 proceedings of the 35th international convention MIPRO*, pp. 1725–1730, IEEE, 2012.

[6] J. J. Bird, A. Ekárt, and D. R. Faria, "British sign language recognition via late fusion of computer vision and leap motion with transfer learning to american sign language," *Sensors*, vol. 20, no. 18, 2020.

[7] M. Oudah, A. Al-Naji, and J. Chahl, "Hand gesture recognition based on computer vision: a review of techniques," *journal of Imaging*, vol. 6, no. 8, p. 73, 2020.

[8] R. P. Sharma and G. K. Verma, "Human computer interaction using hand gesture," *Procedia Computer Science*, vol. 54, pp. 721–727, 2015. Eleventh International Conference on Communication Networks, ICCN 2015, August 21-23, 2015, Bangalore, India Eleventh International Conference on Data Mining and Warehousing, ICDMW 2015, August 21-23, 2015, Bangalore, India Eleventh International Conference on Image and Signal Processing, ICISP 2015, August 21-23, 2015, Bangalore, India.

[9] G. A. Rao, K. Syamala, P. Kishore, and A. Sastry, "Deep convolutional neural networks for sign language recognition," in *2018 Conference on Signal Processing And Communication Engineering Systems (SPACES)*, pp. 194–197, IEEE, 2018.

[10] B. Garcia and S. A. Viesca, "Real-time american sign language recognition with convolutional neural networks," *Convolutional Neural Networks for Visual Recognition*, vol. 2, pp. 225–232, 2016.

[11] R. A. Kadhim and M. Khamees, "A real-time american sign language recognition system using convolutional neural network for real datasets," *Tem Journal*, vol. 9, no. 3, p. 937, 2020.

[12] L. Torrey and J. Shavlik, "Transfer learning," in *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, pp. 242–264, IGI global, 2010.

[13] L. Taylor and G. Nitschke, "Improving deep learning with generic data augmentation," in *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1542–1547, IEEE, 2018.

[14] S. Mitra and T. Acharya, "Gesture recognition: A survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 3, pp. 311–324, 2007.

[15] J. Shin, A. Matsuoka, M. A. M. Hasan, and A. Y. Srizon, "American sign language alphabet recognition by extracting feature from hand pose estimation," *Sensors*, vol. 21, no. 17, 2021.

[16] M. Jia, Y. Zhou, M. Shi, and B. Hariharan, "A deep-learning-based fashion attributes detection model," 2018.

[17] J. Egger, A. Pepe, C. Schwarz-Gsaxner, and J. Li, "Deep learning - a first meta-survey of selected reviews across scientific disciplines and their research impact," workingpaper, Cornell University Library, Nov. 2020.

[18] L. Mo, F. Li, Y. Zhu, and A. Huang, "Human physical activity recognition based on computer vision with deep learning model," in *2016 IEEE International Instrumentation and Measurement Technology Conference Proceedings*, pp. 1–6, 2016.

[19] M. Taskiran, M. Killioglu, and N. Kahraman, "A real-time system for recognition of american sign language by using deep learning," in *2018 41st International Conference on Telecommunications and Signal Processing (TSP)*, pp. 1–5, 2018.

[20] T. N. Abu-Jamie and S. S. Abu-Naser, "Classification of sign-language using mobilenet - deep learning," 2022.

[21] H. B. Nguyen and H. N. Do, "Deep learning for american sign language fingerspelling recognition system," 2019.

[22] S. Gattupalli, A. Ghaderi, and V. Athitsos, "Evaluation of deep learning based pose estimation for sign language recognition," 2016.

[23] O. Oyedotun and A. Khashman, "Deep learning in vision-based static hand gesture recognition," 12 2017.

[24] R. A. K. al Khalissi and M. Khamess, "A real-time american sign language recognition system using convolutional neural network for real datasets," 08 2020.

[25] S. Sharma and K. Kumar, "Asl-3dcnn: American sign language recognition technique using 3-d convolutional neural networks," 07 2021.

[26] J.-H. Kim, N. Kim, H. Park, and J. C. Park, "Enhanced sign language transcription system via hand tracking and pose estimation," *Journal of Computing Science and Engineering*, vol. 10, no. 3, pp. 95–101, 2016.

[27] Y. Chai, H. Liu, and J. Xu, "Glaucoma diagnosis based on both hidden features and domain knowledge through deep learning models," *Knowledge-Based Systems*, vol. 161, pp. 147–156, 2018.

[28] D. Lévy and A. Jain, "Breast mass classification from mammograms using deep convolutional neural networks," 2016.

[29] P. . K. L. A. A. A. K. B. B. E. T. Alam Noor CISTER Research Center, Porto, "A hybrid deep learning model for uavs detection in day and night dual visions," pp. 221–231, 2021.

[30] D. George, H. Shen, and E. Huerta, "Classification and unsupervised clustering of LIGO data with deep transfer learning," *Physical Review D*, vol. 97, may 2018.

[31] S. G. Radhika Malhotra, Barjinder Singh Saini, "Future visions for deep-learning-based approaches for ndds," p. 16, 2022.

[32] M. Zulqarnain, R. Ghazali, Y. M. M. Hassim, and M. Rehan, "A comparative review on deep learning models for text classification," *Indones. J. Electr. Eng. Comput. Sci*, vol. 19, no. 1, pp. 325–335, 2020.