National College of Ireland

A Novel Combination Of 3D CNNs And Recurrent Neural Networks for Sign Language to Text Conversion

MSc Research Project

Data Analytics

Pritish Mehta

Student ID: x20184409

School of Computing

National College of Ireland

Supervisor: Dr. Giovani Estrada

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	British Mehta
Student ID:	x20184409
Programme:	MSc in Data Analytics **Year:** 2022
Module:	Research Project
Supervisor:	Dr. Giovani Estrada
Submission Due Date:	15/08/2022
Project Title:	A Novel Combination Of 3D CNN's And Recurrent Neural Networks for Sign Language to Text Conversion
Word Count:	6300 **Page Count: 21**

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: British Mehta

Date: 15/08/2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

A Novel Combination Of 3D CNN's And Recurrent Neural Networks for Sign Language to Text Conversion

Pritish Mehta

X20184409

Abstract

Sign Language Translation has recently achieved significant accomplishments, raising hopes for improved communication with the Deaf. The primary language of the Deaf population is now American Sign Language (ASL), which is conveyed via body language and understood through eye contact. The main objective of this study is to construct a deep learning-based automatic translation system that can translate ASL to English text. The WLASL dataset is used for the experiment. For sign-to-text translation, this study uses the CNN, GRU, CNN+LSTM, and planned 3D-CNN+LSTM networks. PCA is often employed as a pre-processing step to facilitate the identification of hands. The accuracy of the findings from the suggested model, which is better than the accuracy of the other models utilized in this study, was 83.33% at the end.

1 Introduction

One of the most crucial forms of non-verbal communication is hand movements. The process of interpreting hand gestures with the use of wearable sensors or cameras, also known as hand gesture recognition, tries to translate the movement of the hand into clear instructions. There are 7,151 spoken languages in the globe, including vocal and sign languages, according to Ethnologue, one of the best databases for all language-related information[1]. People with hearing and speech impairments mostly communicate through sign language. According to data from the World Health Organization, 5% of people worldwide—or 432 million adults and 34 million children—have hearing impairment. This is almost double the population of New York. By the year 2050, it is predicted that 700 million individuals will have some form of hearing impairment [2] . People who have hearing loss frequently feel isolated from the rest of the population because of which they find it extremely challenging to go about their daily activities or know how to react in any emergency. People who have hearing loss frequently rely on visual cues around them and frequently experience depression since they are unable to communicate with others. Due to it is importance for those with disabilities, sign language (SL) serves as a bridge for them to develop communication. It's one of the languages where

[1] https://www.ethnologue.com/guides/how-many-languages

[2] https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss

expressing the meaning of a message takes both hand and facial gestures. Grammar and vocabulary are also parts of sign language, just like in other languages. The number of persons who can translate and comprehend sign language is also quite low. With the advancement of algorithms to recognize sign language, machine learning has gone a long way, greatly easing the communication challenge faced by those with impairments. Based on how the motions are conveyed, such as hand detection, form recognition, and sequence classification, Sign Language is categorized into a few categories. Based on the motions of the hand gestures, Sign Language's meaning can simply alter. The problem outlined above can be solved using a variety of methods. The majority of methods only translate signs simply into English alphabets. The first difficulty to be considered is the ability to recognize hand motions because each person's hands are unique in look and form. Second, using different strategies to construct lengthy sequences of phrases is a problem in sentence construction. The goal of this project is to develop a translation system that can quickly convert ASL motions into English text and vice versa by displaying the video frame of each letter or word. The initial stage in translating sign language to text is to segment images using a 3D-CNN technique. Second, to identify hand signals, YOLO and Faster R-CNN algorithms rapidly and accurately will be utilized. Finally, after acquiring the translated videos and images, 3D - CNN may be employed once more for summarizing lengthy words. On the other hand, to translate text into signs, we first utilize neural networks to tokenize each video clip and extract frames from the video dataset. These frames may then be translated using the text.

Table 1 from Section 2 contains a comprehensive comparison of our study with past review research publications. The following section is organized as follows: Section 3 outlines the approach as well as the first procedure. Section 4 describes the design requirements, architecture, and an overview of the proposed Machine Learning and Deep Learning algorithms. Sections 5 and 6 discuss the research's execution and assessment. Section 7 of the study article discusses findings and limitations.

2 Related Work

The literature is divided into two sections: one is a Recognition using external devices, and the other webcams. Since American Sign Language is one of the most widely used datasets that is readily accessible to the public, it will be the dataset that is used. For improved recognition, a variety of pre-processing techniques are applied, such as picture compression and transforming the available images to greyscale or a variety of colour scales. This article reviews translation that has utilized several machine learning and neural network modules based on the data acquired from various sensors to recognize the movement of the hand.

2.1 Recognition using external devices

In an effort to improve communication between the deaf and hard of hearing and the hearing world, experts have been working on translating American Sign Language (ASL) into English over the past several decades. Many of these devices employ cumbersome, unpleasant gloves to record distinctive motions. The use of sign language may be done in a

number of different ways. Flex sensors and accelerometers are used in the examples from (Punsara, et al., 2020), (Rizwan, Khan, & Imran, 2019), and (Yeasin, et al., 2019) to record the data stream from the gloves where the sensors are positioned. Flex sensors are employed to determine each finger's degree of deflection, which results in a more accurate recording of the gesture. The accelerometer is utilized to capture hand movement together with the flex sensors. The use of both sensors together allows for a more precise detection of the sign being done by giving an exact depiction of how the hand is moving. (Yeasin, et al., 2019) recorded gestures and assisted with sign translation using data from the sensors that were supplied to the Arduino Nano. The audio amplifier is then used to transform the translated signs into speech. The limitation of this study is the manual configuration of each sign language word or letter. Furthermore, the strategy adopted by (Rizwan, Khan, & Imran, 2019) was comparable to the one previously discussed. The main distinction is that the gesture or sign is represented visually and acoustically on the android device using an Arduino Leonardo and machine learning to determine the alphabet or word. The model works quite well, with a 94.23% accuracy rate. The sensors' mistakes during altering gestures throughout this experiment were to blame. Comparing the approach employed by (Punsara, et al., 2020) to the methods mentioned above, it is a more cost-effective model. Instead of using Arduino in this research, the same sensors are combined with Printed Circuit Board (PCB) to capture the data and provide a mean dataset. After the gesture has been recognized, it is once more input into LSTM model for phrase synthesis. Finally, after evaluation, the model's accuracy is 87%, which is lower than it was with the prior model. The customized PCB speeds up processing and saves money because all models use the same glove-based technology. The cost of sign language translation can be significantly reduced. The research portrayed by (Zhao, et al., 2021) and (Park, et al., 2020) both provide examples of two such methods that emphasize the use of easily accessible gadgets. The tools employed include smartphones and smartwatches, which are widely accessible in the modern world. The solution suggested by Park et al. (Park, et al., 2020) makes use of depth video samples acquired at 10 frames per second from a mobile device's front camera (frames per second). To help users with the motions and video labelling, a customized app was created. The movies were then run through MobileNet v2 CNN with weights taken from the ImageNet training phase. Additionally, the output is used for classification in the LSTM model with ReLU and SoftMax layers. When assessed with 26 participants and 17 words in Korean Sign Language, the outcome showed a 92% accuracy. This study demonstrates how hand gestures in sign language may be accurately detected and converted to text using a depth image from a mobile device. This method's main flaw is the short dataset utilized to accomplish the translation, which means that even if the dataset's size is expanded, the accuracy won't remain the same. The PPG sensor, which is used in smart wearable devices and used to monitor pulse, heart rate, and blood pressure, was used by (Zhao, Liu, Wang, Liu, & Chen, 2021). Each intricate muscle movement required for hand- and finger-gestures presses the artery at a specific angle. Blood absorbs most of the green light coming from the sensors. The waveform of the light reflecting to the sensors is altered by variations in the angles. Performance is enhanced by using the motion sensor in conjunction with the PPG sensors. Therefore, finger level gesture recognition uses the PPG data that was acquired together with the motion and cardiac data. Before employing the Gradient Boosting classifier, several strategies were used

to retrieve the PPG data. Additionally, gesture detection makes use of Deep Residual Network. In the end, 98% accuracy was achieved using only the PPG sensor by combining the detection and extraction of PPG data with the use of Gradient Boost Classifier and Deep ResNet. The research's benefit was the development of a highly accurate, reasonably priced recognition system for sign language. The Kinect motion detection sensor from Microsoft is utilized with the Xbox 360. Through facial and speech recognition, the Kinect detects the person. Its depth camera builds a 3D skeleton representation of the player, and it is motion sensor picks up their physical motions. The system can interpret spoken orders according to it is voice recognition capabilities, and it can follow player motions thanks to gesture recognition. On the other hand, Orientation Based Hash code, Gabor Filter, and Artificial Neural Network have all been employed by (Arif-Ul-Islam & Shamim, 2018). To limit the amount of information input to the ANN model, principal component analysis is utilized in conjunction with ANN. The result of which demonstrated 93.85% accuracy with 9-fold validation and 95.8% accuracy versus a random dataset. An interesting approach by (Wu, Sun, & Jafari, 2016) suggested using a customized IMU sensor with three gyroscopes, three accelerometers, and three magnetometers to record real-time data that can be analysed afterwards. A sEMG is further employed to identify and assess the electrical activity of the muscle. Data from the 3D accelerometer, 3D gyroscope, and the 4 channel sEMG—used for pre-processing and noise rejection—are collected throughout the training stages. Additionally, classification is accomplished using techniques like LibSVM, NN, DT, and Naive Bayes. The result shows that 40 characteristics had an average accuracy of 96.16%. All 80 features cannot be taken into consideration due to wearable system limitations.

2.2 Recognition using webcams

In order to improve communication between the hearing world and the deaf and hard of hearing community, a variety of software programs that translate American Sign Language (ASL) into English have been employed in research during the past ten years. Convolutional neural networks are one of the models used in the translation process in most of the today's research on sign language translation. The researh done by (Liao, Xiong, Min, Min, & Lu, 2019) makes use of the DEVISIGN D dataset, a collection of films in Chinese Sign Language. The study uses CovNet and Bidirectional LSTM to demonstrate the utilization of 3D-Residual networks (B3DResnet). The position of the hand is captured by Faster R-CNN, which performs the object detection. Conv Layers' trained video frames are utilized to extract features. Using the ROI Pooling layer, a fixed size feature map is produced. The segmented movies are completely trained using the B3D ResNet model during the video extraction stage to produce the feature vectors. For the DEVISIGN D and SLR datasets, respectively, the results of using this approach yielded sign detection accuracy of 89.8% and 86.9%. Future implications imply that there is room for phrase construction and the addition of new traits to the article. On the other hand, CNN and CABM were combined for image and video recognition by (Khan, et al., 2021). The responsibilities of CBAM include channel and spatial detection. ResNet-18's convolution layers are positioned within blocks and prior to the classifier. Using two convolution layers, the results demonstrate that the prior classifier outperforms the CABM-2D ResNet inside blocks. Prior Classifier excels in picture recognition. This research describes a revolutionary method for recognizing images and

videos using the CNN network, which produced a completely accurate result. In comparison to the model employing CABM, the method using 3D-ResNets, and Faster R-CNN performs better overall. Tokens are employed in a novel way as a pre-processing stage for SL translation by (Orbay & Akarun, 2020). To help them learn from the sign movies, they have used the tokens. It makes use of the RWTH-PHOENIX-Weather-2014T (Koller, et al., 2015). To identify body and hand movements that are present in the frame, OpenPose which has served as the very first real-time multi-person system to detect keypoints on a single picture, including feature points on the human body, hands, face, and feet. is used. They ran upon two major problems: first, the dataset's backgrounds varied, and the data were poorly labelled. They employed a hybrid CNN-HMM model to address the first problem, using CNN for picture identification and HMM for accurate labelling. The study shown by (Yuan, et al., 2019) used a creative approach when they employed S2VT, OpenPose, and other similar tools to interpret sign language. They chose CSLD, a sizable Chinese dataset with 24 million colour and depth photos and 49,708 movies. First, the movies' captions are generated using the S2VT (sequence to sequence model). The body postures in this model, which has two layers of LSTM. The RGB camera is used to extract elements like facial expression. OpenPose is further employed for body detection. Additionally, the Jieba tools used for segmentation were employed to perform the tokenization. The team's future goals include growing the ad dataset and further enhancing the recordings. The outcomes of this paper totally depend on the student taking the exam's comprehension because it is based on control and experimental groups. Different lines of study use various techniques to recognize sign language. A unique method for sign language identification is used in the study by (Saleh & Walaa, 2020). As a semantic segmentation model, they employed DeepLab v3+. Encoding and decoding make up it is two stages. With the use of pre-trained CNN models, the relevant information from the picture is extracted during the encoding process. The extraction of hand shape features is done next using Convolutional Self-Organizing Maps (CSOM), which consists of three processing layers: contrast normalization, CSOM, and local histogram. Finally, the videos are categorized using Bi-LSTM. By delivering an output of 87.5%, which was well beyond the present state-of-the-art systems, using the models and mapping them produced astonishing outcomes. A unique approach is shown by (Thrimahavithana, Yasodha, Kannangara, Welgama, & Weerasinghe, 2019) which provide examples of a method for constructing 3D-Avatars that portray what the text says (Duarte, 2019). The video frames could not be as clear when displayed as the result of a text translation, which is a major problem. However, this problem is resolved through the usage of 3D-avatar-based translation. With their method, (Thrimahavithana, Yasodha, Kannangara, Welgama, & Weerasinghe, 2019) converted the sign language of Sri Lanka to a 3D avatar in the Sinhala language. Using Java 3D, they developed a database and application to record the dynamic meaning of the words that users contribute. A local NGO then evaluates the application for two groups. Eight students are divided into experimental and control groups to create the groupings. Additionally, it was evident that the experimental group's students outperformed those in the control group based on studies of their text-based understandability. The results of the test totally depend on the exam-takers' comprehension because the paper is based on control and experimental groups. However, (Duarte, 2019) took a somewhat different strategy since they made a brand-new public dataset. Their strategy focuses on breaking the process down into

three components, the first of which is the conversion of text or voice to Gloss utilizing an encoder-decoder network design based on LSTM. The Gloss is next transformed into a skeleton, which is an important stage since it creates the human stances that match to the text or phrase. Finally, they have produced the sign language for the users using two methods. The avatar was first animated using the skeleton's main points. Additionally, the dataset developed that includes all the information from films, glossaries, and translators to speech may be utilized to quickly recognize and interpret sign language.

Table 1: Summary of the related work on Hardware and Software based Recognition

Name of the Researcher	Main Model/Code	Hardware used	Accuracy
(Arif-Ul-Islam & Shamim, 2018)	OBH, OpenCV and PCA	Kinect Sensors	96%
(Duarte, 2019)	Encoder-Decoder, LSTM	Webcam	91.60%
(Punsara, et al., 2020)	LSTM	Accelerometer & Flex Sensor	87%
(Khan, et al., 2021)	CN	CABM	99.17%
(Liao, et al., 2019)	3D-ResNet, CovNet	Faster R-CNN	89.80%
(Orbay & Akarun, 2020)	OpenPose & CNN- HMM	3D-CNN	92.88%
(Rizwan, et al., 2019)	Arduino Leonardo	Flex Sensors	65.70%
(Saleh & Walaa, 2020)	CNN	CSOM	87.50%
(Thrimahavithana, et al., 2019)	3D Avatar-based Model	Avatar Based	65%
(Wu, et al., 2016)	LibSVM, Neural Networks	EMG	96.10%
(Yeasin, et al., 2019)	Arduino Nano	Flex Sensors	N/A
(Yuan, et al., 2019)	S2VT, LSTM	OpenPose, Jieba Tool	N/A
(Zhao, et al., 2021)	Gradient Boosting and Deep ResNet	Photoplethysmography Sensor	89%
(Park, et al., 2020)	LSTM with ReLU	iPhone Depth Sensing Camera	92%
(Suri & Gupta, 2019)	CNN	IMU	95%
(Ko, et al., 2019)	GRU	Webcam	93.28%
(Koller, et al., 2015)	CNN into HMM	Webcam	73.40%
(Kopuklu, et al., 2019)	3D Convolutional Neural Networks	Webcam	94%
(Papastratis, et al., 2021)	Multiple Deep Learning Models	Webcam	98.09%
(Ahmed, et al., 2016)	N/A	Kinect Sensors and Unity 3d	84-87%

3 Research Question

What is the most effective way to combine CNN and LSTM layers for increasing accuracy of sign language applications? What impact have LSTM layers on convolutional networks for sign language recognition?

4 Research Methodology

This section discusses the steps used in the research technique. As illustrated in fig. 1, the primary phases in this research include data collection, pre-processing, data modelling and evaluation, and knowledge. We have utilized the KDD methodology for this report. There are nine phases in the iterative, interactive knowledge discovery process. Each stage of the procedure is iterative, which suggests that going back to earlier steps can be necessary. The process requires a great deal of imagination because it is impossible to present a single formula or classify all possible applications in full scientific terms. Therefore, it is important to comprehend the process as well as the various needs and opportunities at each stage.

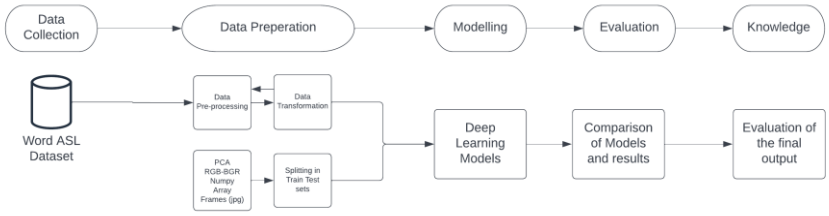

Figure 1: KDD Methodology

4.1 Data Collection

Word-level ASL datasets have been made widely available. The WLASL, which contains 2,000 typical ASL terms, is the biggest video dataset for word-level American Sign Language (ASL) recognition. WLASL can help with sign language comprehension study and ultimately improve deaf and hearing cultures' ability to communicate (Li, et al., 2020). A GitHub repository hosting the dataset has python scripts that may be used to download about 28,000 movies for every word.

4.2 Data Pre-processing and Transformation

The videos may be seen in Jupyter notebook using the OpenCV library and are in the MP4 format. Additionally, the films are first transformed to Grayscale so that we may use the unique PCA approach that has been provided to efficiently detect hands utilizing media pipe. Additionally, we must convert the RGB video to BGR for the Mediapipe detection to accurately identify hand and face locations. The next step is to write down the frames and the places where the hands were placed. The hand and facial coordinates are entered into a numpy array using OpenCV and Mediapipe, and the frames are concurrently taken as JPEG files. The numpy array will then be used by the LSTM and GRU model while the JPEG files will be used by CNN, CNN+LSTM and 3D-CNN+LSTM

4.3 Modelling

On the pre-processed data, five distinct models are used in this stage. On the pre-processed video dataset, we applied deep learning algorithms such Convolution neural networks, long short-term memory, Gated recurrent units, CNN - LSTM, and 3D CNN-LSTM as baseline techniques. The innovation of this study for identifying hand movements is the application of PCA on 3D-CNN+LSTM model.

4.4 Evaluation

Finally, there are four possible outcomes when making categorization predictions.

- A true positive occurs if we accurately predict how a data fits to a category and it does.
- A true negative if we predict how a data may not correspond to a class and it really does not.
- A false positive when you predict how a finding did belong to a category when it doesn't.
- A false negative if we predict how a data does not fit to a category when it really does.

The structure and behaviour of a learning curve may be used to diagnose the behaviour of a machine learning algorithms, which could then advise configuration changes to improve learning and/or efficiency. The example would be based on the idea if we are using a decreasing measurement, in which case higher training is demonstrated by reduced relative score on the y-axis.

5 Design Specification

This section discusses the suggested various model architecture. Convolution neural networks, LSTM, GRU, CNN - LSTM, and 3D CNN-LSTM with additional layers are used in this research, along with several pre-processing techniques such picture augmentation, image size reduction, and colour conversion.

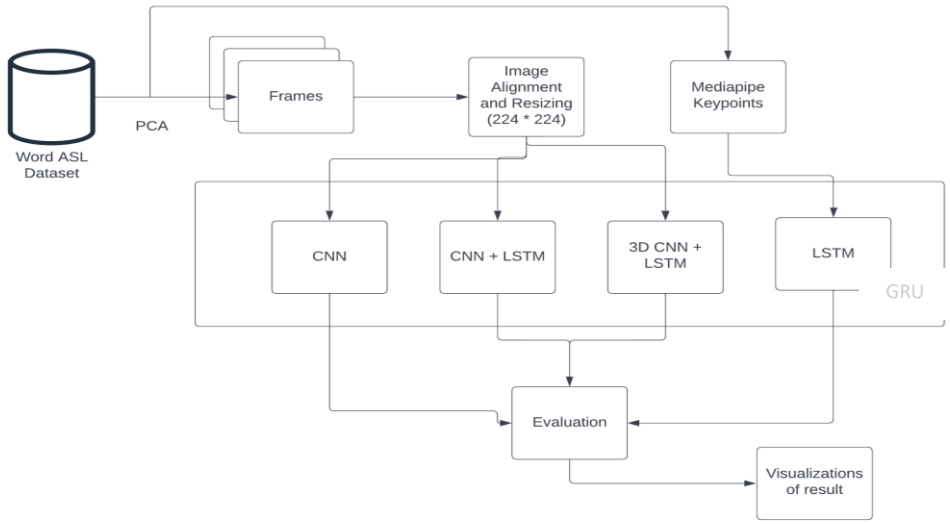

Figure 2: Process Flow Model

The system that is being developed is for translation and recognition of sign language. That is, it will display the shown hand gesture or ASL word in text after recognizing it. ASL hand gestures are used as the input image, while ASL words are used to train the models. We've also included a wait queue that will serve as a pause between words.

The modules for ASL Translation and Recognition are listed below.

- Pre-processing

  The 28,000 videos that make up the collection, which was gathered from a public portal, total 2000 words. We apply PCA to the movies after collecting the dataset, using the media pipe and OpenCV functions to extract the frames and frame points. While the frame points are kept in ".npy" files, the frames are kept in ".JPEG" files. Splitting the dataset into train and test sets is the last step before feeding the data to the models.

- Models

  Following are the models used for this experiment
  - CNN
  - GRU
  - CNN+LSTM
  - 3D-CNN + LSTM

  The brief description of all the layers used is explained in Section 5.

- Evaluation

  After all the models are trained, we further use OpenCV to perform the prediction on the hand gestures accurately and all the metrics are visualized.

5.1 Tool Used

5.1.1 Implementation

The Jupyter Notebook has been used to complete this project in it is entirety. A server-client program called the Jupyter Notebook enables editing and executing notebook papers from a web browser. The Jupyter Notebook can be run locally on a desktop without the need for an internet connection. It was created specifically to carry out deep learning tasks. The AMD Ryzen 5 CPU and 16GB of RAM are powering the processing system.

5.1.2 Model Creation

An open-source Python library called TensorFlow aids in quick numerical computation. It is an artificial intelligence library that primarily does tasks like categorization, prediction generation, etc. by using data flow graphs to create a model. TensorFlow has been heavily leveraged for several tasks in this project. To work on pre-processing of the image data, such as resizing and rescaling of the picture, ImageDataGenerator is loaded from the TensorFlow package. This assists in tracking model execution so that, in the event of a disruption like a runtime disconnect, the model may be recovered from it is most recent execution and subsequent execution can resume.

6 Implementation

This section described how multiple Deep Learning models were used to translate ASL. The three stages of the study were data preparation, cross-validation, and model refining utilizing hyperparameters. The goal of using hyperparameters is to improve the models' prediction accuracy. The findings of multiple detection models are evaluated in order to select the optimum predictive analytic strategy.

6.1 Deep Learning

A subtype of machine learning is deep learning, which is just a neural network having 3 or even more layer. These neural networks attempt to imitate how well the human mind works, but they are unable to replicate it, allowing it to "learn" from huge amounts of information. Transfer learning, a technology that powers most artificial intelligence (AI) products or services, improves robotics by performing mental and physical tasks while requiring human intervention. All current and emerging technology, like vocal style Television remote control, smart speakers, credit card fraud detection, and self-driving automobiles, are powered by deep learning.

6.1.1 Convolutional Neural Network

Convolutional neural networks (CNN/ConvNet) are a kind of deep learning models that are frequently employed in deep learning to assess visual pictures. Matrix multiplications come to mind when we think of neural networks, however ConvNet does not operate in this way. It uses a technique called convolution. The raw pixel data of an image is processed via a series of filters by CNNs to extract and learn higher-level characteristics that the model may utilize for categorization.

CNNs are composed of three parts:

- Convolutional filters are added to an image using convolutional layers. Each subregion is subjected to a series of arithmetic computations by the layers in order to produce a specific value for the predicted output. By adding a ReLU activation function to the outputs, convolution layer then are frequently employed to introduce nonlinearities into the model.
- To reduce the dimensions of the feature maps and thereby shorten processing time, convolution layers that downsample the photo data generated by the convolution operation are used. A well-liked pooling method called max pooling takes feature map subregions, keeps their maximum benefit, and throws away all other values.
- Dense layers that classify the features recovered by convolutional layers and downsampled by pooling layers. Every node in a dense layer is linked to every node in the previous layer.

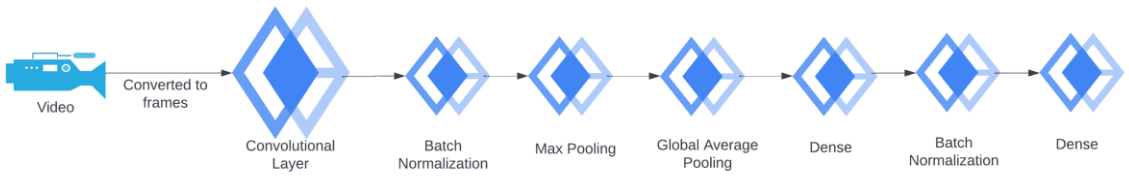

Figure 3: CNN Process Flow

The Figure 3 shows the Architecture of the CNN network used in the research. The video is first taken as input and then converted into different frames of images. Then the frames are passed through the convolutional layer to extract the relevant features from the images. For the purpose of this experiment, we have utilized one convolutional layer with two batch normalizations one before the max pooling layer and the other between the two dense layers.

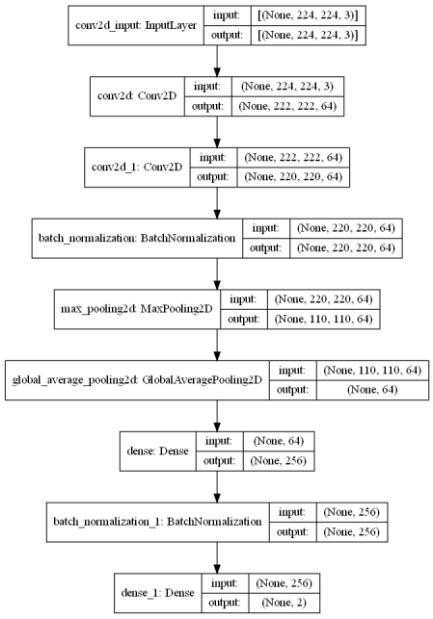

Figure 4: CNN Model Architecture

6.1.2 GRU

The GRU is a member of the more recent generation of recurrent neural networks and resembles an LSTM in many ways. The concealed state was employed by GRUs instead of the cell state for information transmission. A reset gate and an update gate are the only gates it possesses. Another intriguing feature of GRU is that, in contrast to LSTM, it lacks a distinct cell state (Ct). There is just a concealed state (Ht). GRUs are quicker to train because of the architecture's simplicity.

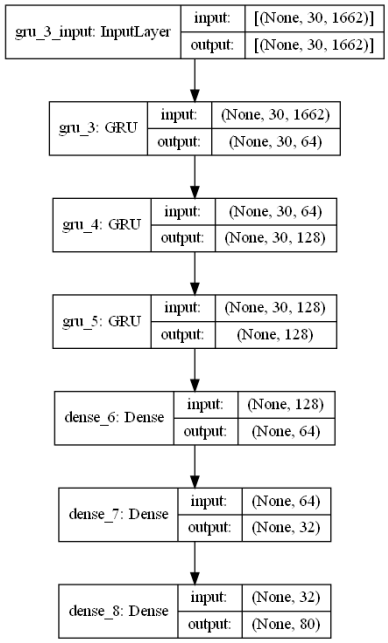

Figure 5: GRU Model Architecture

To the experiment, we first utilise the numpy arrays that were created from the media pipe detection. The numpy array contains the co-ordinates of all the facial and hand points to detect the movements. The numpy array is then fed into the LSTM model. The model that is shown above depicts that there are 2 LSTM layers and 3 dense layers.

6.1.3 CNN – LSTM

In the CNN LSTM architecture, Convolutional Neural Network (CNN) layers for feature extraction on input data are coupled with Long short - term memory to improve sequential predictions. Despite the fact that during this session we will refer to Long short - term memory that use a CNN as a front end as "CNN LSTM," this design was first referred to as a Long-term Recurrent Convolutional Network or LRCN model.
In this experiment we have used the Time distributed layer. With the help of this specialized layer, we may apply the same layer to several inputs and obtain output for each one separately so that we can aggregate them and transmit the results to a different layer to generate predictions. Apart from that we have utilised 3 Conv2D layers along with 1 GRU layers. We have also used BatchNormalisation layers with each Conv2D layer. The layer of batch normalization enables the network's layers to learn more independently. The output of the earlier layers is normalized using it. The activations scale the input layer during normalization. Batch normalization improves learning efficiency and can also be used as generalization to avoid model fitting problem. The layer has been added to the sequential model to regulate the input or output. It may be used at many locations between both the layers of the model. After describing the sequential model and just after the convolution and pooling layers, it is typically placed.

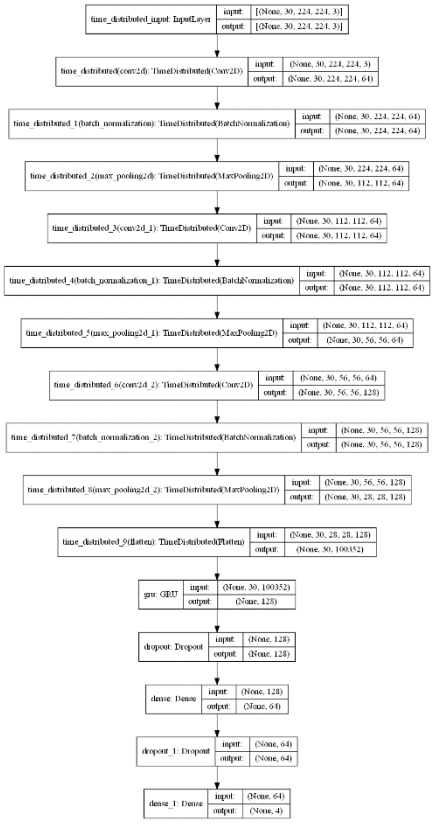

Figure 6: CNN-LSTM Model Architecture

6.1.4 The Proposed 3D-Convolutional Neural Network – LSTM Network

Regardless of what we say, a 3D CNN still closely resembles a 2D CNN. However, it differs in the following ways (non-exhaustive listing). An initial 2D convolution layer is a multiplication of the input by each of the many filters, where the filters and inputs are 2D matrices. The same procedures are employed in a 3D convolution layer. These procedures are performed on several pairs of 2D matrices. Slides step choices and padding options function similarly.

In this experiment we have utilized Conv3D layers along with ConvLSTM2D layer. Most often, 3D picture data is utilized using Conv3D. like the results of magnetic resonance imaging. We have utilized 2 Conv3D layers. Conv3D may be used to categorize or extract characteristics from the dataset. Kernel flows in three directions in 3D CNN. The 4 dimensions of the 3D CNN's input and output data. Additionally, we have utilized 2 MaxPooling3D layers. The goal of the 2D Maxpool Layers (2x2 filter) is to extract the most significant element from a 2x2 square that we delimit from the input. We now search for the most elements in a width 2 cube in a 3D Maxpool (2x2x2 kernel). The area that the 2x2x2 input zone defines is represented by this cube. Finally, we have utilized the ConvLSTM2D layers which is a Convolutional-LSTM model is an alternative to a ConvLSTM in which the picture is processed via convolutional layers and the resulting set is flattened into a 1D array containing the resulting features. Repeating this procedure on all the photos in the time set yields a collection of features across time, which serves as the input for the LSTM layer.

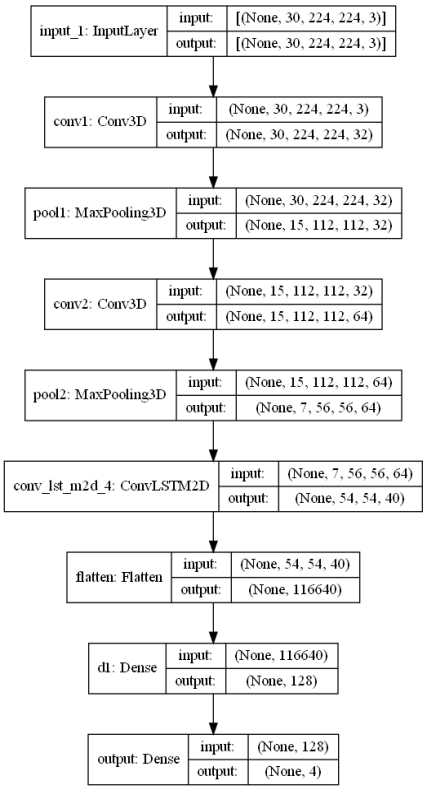

Figure 7: 3D-CNN+LSTM Model Architecture

7 Evaluation

This section focuses on the assessment of the used technique. The primary criteria used to assess the model's implementation are accuracy and loss. For each model, the model was run through 200 epochs. The metrics were set to accuracy when the models were created, allowing for fine-tuning. In this way, the accuracy and loss are recorded when each epoch is run. In order to separate the data into train and test sets, 95% of the data is used for training and 5% for testing.

7.1 Experiment with CNN Model

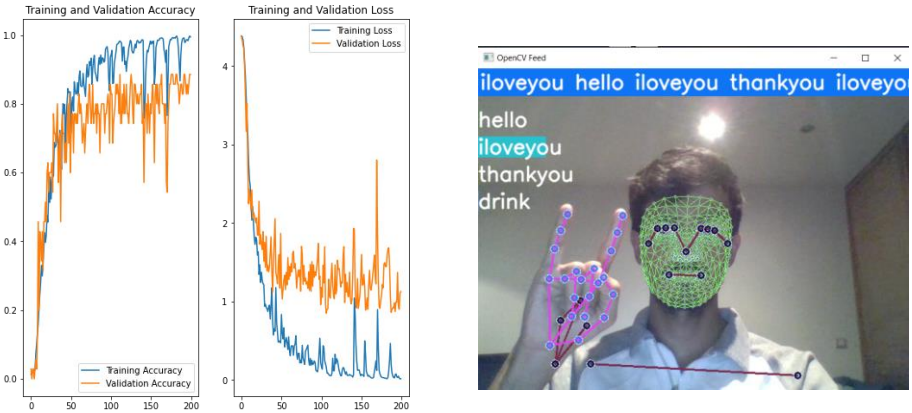

Figure 8: CNN Model Evaluation and Accuracy/Loss graphs

A training loss that is dropping and keeps decreasing towards the conclusion of the plot can also be used to spot an underfit model. This shows that the training process was stopped too soon and that the model is capable of more learning and potential enhancements.

7.2 Experiment with GRU Model

Figure 9: GRU Model Evaluation and Accuracy/Loss graphs

This typically means the model is overfitting and unable to generalize to fresh data. The model performs very well on training data but poorly on the fresh data in the validation set. The validation loss first lowers but then gradually increases again. The model may be too sophisticated for the data, or it may have been trained for a lengthy time, as a noteworthy cause of this occurrence. When the loss is small and stable, training can be stopped—a practice known as early stopping—in this situation. One of the various strategies used to avoid overfitting is early quitting.[3]

7.3 Experiment with CNN – LSTM Model

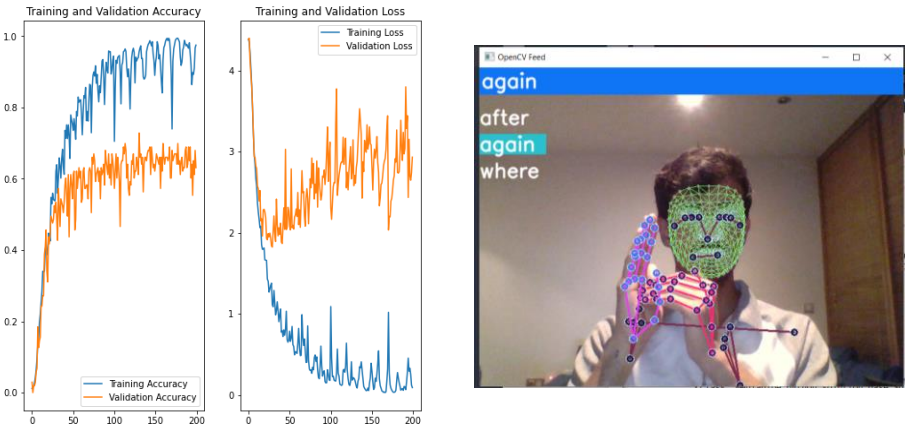

Figure 10: CNN-LSTM Model Evaluation and Accuracy/Loss graphs

[3] https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance/

When a model has learnt the training dataset—including the statistical noise or random fluctuations in the training dataset—too well, it is said to be overfit. Overfitting has the drawback of making a model less effective at generalizing to new data as it becomes more focused on training data, which raises generalization error. The model's performance on the validation dataset may be used to gauge this rise in generalization error.

7.4 Experiment with the proposed 3D CNN – LSTM Model

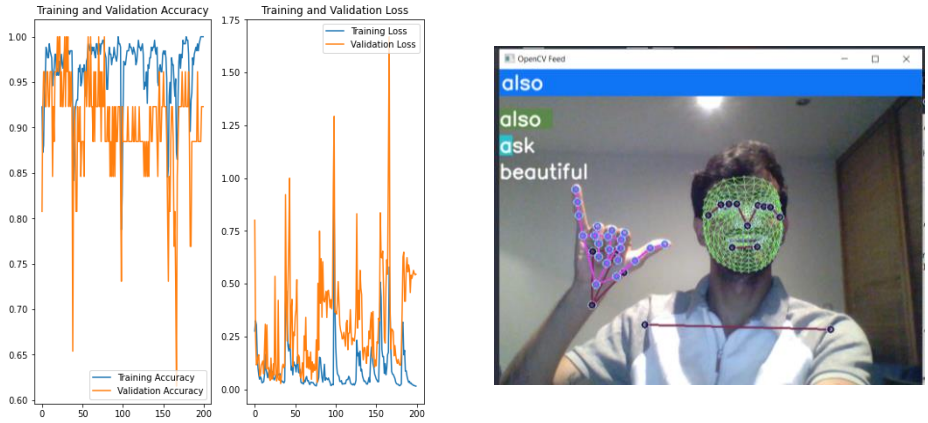

Figure 11: 3D-CNN-LSTM Model Evaluation and Accuracy/Loss graphs

The learning algorithm aims for a suitable fit, which may be found between an overfit and an underfit model. A training and validation loss that lowers to a point of stability with a small difference between the two final loss values indicates a good match. Almost invariably, the training dataset will have a lower model loss than the validation dataset. Accordingly, there will likely be a discrepancy between the train and validation loss learning curves. The "generalization gap" is the name given to this discrepancy.

Model	Epochs	Training Time (s)	Accuracy
CNN	200	2,350	73.33%
GRU	200	1,500	55%
CNN + LSTM	200	18,000	63.33%
3D CNN + LSTM	200	28,800	83.333%

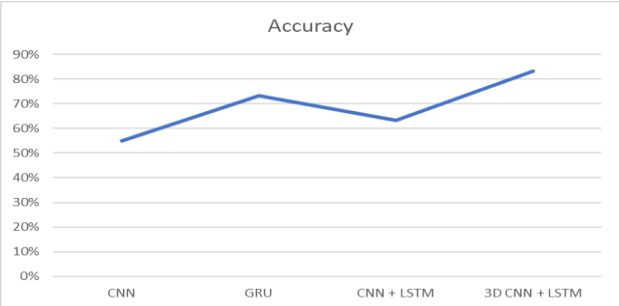

Figure 12: Accuracy Graph

7.5 Discussion

Four deep learning techniques—CNN, GRU, CNN+LSTM, and 3DCNN+LSTM—are used in this study. The major goal of this study is to determine how a person who has trouble speaking and hearing might wish to explain something to someone who does not know sign language. As a result, communication becomes extremely challenging for someone who is deaf and dump. The suggested method would make things simpler; all that is required is for the user to display an ASL word on the webcam, and the system will display the words that go with it on top of the screen, making sentences.

The proposed system that contrasts the proposed model of 3DCNN+LSTM with the detection of sign language. The accuracy of 83% attained higher than that of the other 3 models, as was demonstrated in the preceding section. Which support the research question that 3D-CNN+LSTM is the effective way of combining the CNN and LSTM model to get better accuracy. Additionally, after analysing the Accuracy and Loss graph, we can observe that the 3D-CNN + LSTM model, which was the proposed model, showed the best match out of the 4 models.

The 3DCNN and LSTM model combination's ability to extract the spatio-temporal aspects of the gesture sequence, particularly when it comes to dynamic gesture detection, is the suggested model's main strength. In particular with the dynamic gesture recognition, the 3DCNN and LSTM model combination could extract the spatio-temporal aspects of the gesture sequence. This work's objective was to develop a network that can recognize short-term temporal elements in lengthy video sequences. To understand the long-term dependencies in lengthy video sequences, an LSTM unit was added to the final 3D-CNN.

8 Conclusion and Future Work

The main objective of this research is to find an effective way to combine CNN and LSTM networks to further increase the accuracy of sign language recognition. The idea is to build a suitable model combining the Conv layers and LSTM layers to translate the Sign Language to words. Hence, we tried various approaches and finally combined the 3DCNN model with LSTM model. This model utilizes Conv3D layers along with ConvLSTM2D layers to further improve the recognition of sign language. The models were built for WLASL dataset with extra layers and PCA as a pre-processing technique to improve the hand and pose recognition. The proposed model has achieved an accuracy of 83.33% which is better when compared to the other models like CNN, GRU and CNN+LSTM in this research. The effective spatiotemporal characteristics are extracted in the form of feature maps via three-dimensional convolutional components. We also create a linear weighted fusion technique to successfully combine spatial and temporal feature maps. In order to get video level representations for video classification, we finally use an LSTM encoder/decoder. Finally, this model is fully software and vision based hence it is not expensive.

Limitations of this research: The limitation of this model was the processor intensive tasks like pre-processing the videos which took more than 40 hours. Additionally, since the system used did not have enough processing power and RAM only 100 words were trained out of the 2000 words in the dataset.

Future Scope: As part of our ongoing effort, we will incorporate sentence creation and leverage models like HMM and other NLP models to enhance word recognition and create sentences that are grammatically accurate. Finally, since the model was only able to run on 100 words due to limited processing power running the same model using the full dataset on a machine with good processing power will help with better visualisations and evaluations.

9 Acknowledgements

I am really appreciative to my supervisor Giovani Estrada for his outstanding guidance and insightful criticism during my research project. I genuinely appreciate all the inspiration and drive he has given me to advance my study.

10 References

Arif-Ul-Islam & Shamim, A., 2018. Orientation hashcode and articial neural network based combined approach to recognize sign language. 2018 21st International Conference of Computer and Information Technology (ICCIT).

Duarte, A. C., 2019. Proceedings of the 27th ACM International Conference on Multimedia. Association for Computing Machinery, p. 1650–1654.

Punsara. et al., 2020. IOT based Sign Language Recognition System. 2020 2nd International Conference on Advancements in Computing (ICAC).

Khan, R. U. et al., 2021. Intelligent Malaysian sign language translation system using convolutional-based attention module with residual network. Computational Intelligence and Neuroscience, pp. 1-12.

Koller, O., Forster, J. & Ney, H., 2015. Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers. Computer Vision and Image Understanding, Volume 141, pp. 108-125.

Kopuklu, O., Kose, N., Gunduz, A. & Rigoll, G., 2019. Resource efficient 3D convolutional Neural Networks. 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW).

Ko, S.-K., Kim, C. J., jung, H. & Cho, C., 2019. Neural sign language translation based on human keypoint estimation. Applied Sciences, 9(13), p. 2683.

Liao, Y. et al., 2019. Dynamic sign language recognition based on video sequence with BLSTM-3D residual networks. IEEE Access, pp. 38044 - 38054.

Li, D., Opazo, C. R., Yu, X. & Li, H., 2020. Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison. 2020 IEEE Winter Conference on Applications of Computer Vision (WACV).

Orbay, A. & Akarun, L., 2020. Neural Sign Language Translation by Learning Tokenization. CoRR.

Papastratis, I. et al., 2021. Artificial Intelligence Technologies for sign language. Sensors, p. 5843.

Park, H., Lee, J.-S. & Ko, J., 2020. Achieving real-time sign language translation using a smartphone's true depth images. 2020 International Conference on COMmunication Systems & Networks (COMSNETS).

Rizwan, S. B., Khan, M. S. & Imran, M., 2019. American Sign Language Translation via smart wearable glove technology. 2019 International Symposium on Recent Advances in Electrical Engineering (RAEE).

Saleh, A. & Walaa, A., 2020. Deeparslr: A novel signer-independent deep learning framework for isolated Arabic sign language gestures recognition. IEEE Access, pp. 83199-83212.

Suri, K. & Gupta, R., 2019. Convolutional Neural Network Array for Sign Language Recognition Using Wearable imus. 2019 6th International Conference on Signal Processing and Integrated Networks (SPIN).

Thrimahavithana, S. et al., 2019. Empowering the text-based understandability of students with hearing impairments. 2019 19th International Conference on Advances in ICT for Emerging Regions (ICTer).

Wu, J., Sun, L. & Jafari, R., 2016. A wearable system for recognizing American sign language in real-time using IMU and surface EMG Sensors. IEEE Journal of Biomedical and Health Informatics, pp. 1281 - 1290.

Yeasin, M. et al., 2019. Design and implementation of bangla sign language translator. 2019 5th International Conference on Advances in Electrical Engineering (ICAEE).

Yuan, T. et al., 2019. Large scale sign language interpretation. 2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019).

Zhao, T. et al., 2021. Towards low-cost sign language gesture recognition leveraging wearables. IEEE Transactions on Mobile Computing, pp. 1685 - 1701.