

West Chester University

Digital Commons @ West Chester University

Computer Science Faculty Publications

Computer Science

10-2023

An AI-Based Framework for Translating American Sign Language to English and Vice Versa

Vijayendra D. Avina

Md Amiruzzaman

Stefanie Amiruzzaman

Linh B. Ngo

M. Ali Akber Dewan




Follow this and additional works at: https://digitalcommons.wcupa.edu/compsci_facpub



Part of the [Artificial Intelligence and Robotics Commons](#)

Article

An AI-Based Framework for Translating American Sign Language to English and Vice Versa

Vijayendra D. Avina¹, Md Amiruzzaman^{1,*}, Stefanie Amiruzzaman^{2,*}, Linh B. Ngo¹
and M. Ali Akber Dewan^{3,*}

¹ Department of Computer Science, West Chester University, West Chester, PA 19383, USA; va985361@wcupa.edu (V.D.A.); lngo@wcupa.edu (L.B.N.)

² Department of Languages and Cultures, West Chester University, West Chester, PA 19383, USA

³ School of Computing and Information Systems, Faculty of Science and Technology, Athabasca University, Athabasca, AB T9S 3A3, Canada

* Correspondence: mamiruzzaman@wcupa.edu (M.A.); samiruzzaman@wcupa.edu (S.A.); adewan@athabascau.ca (M.A.A.D.); Tel.: +1-610-436-3230 (M.A.)

Abstract: In this paper, we propose a framework to convert American Sign Language (ASL) to English and English to ASL. Within this framework, we use a deep learning model along with the rolling average prediction that captures image frames from videos and classifies the signs from the image frames. The classified frames are then used to construct ASL words and sentences to support people with hearing impairments. We also use the same deep learning model to capture signs from the people with deaf symptoms and convert them into ASL words and English sentences. Based on this framework, we developed a web-based tool to use in real-life application and we also present the tool as a proof of concept. With the evaluation, we found that the deep learning model converts the image signs into ASL words and sentences with high accuracy. The tool was also found to be very useful for people with hearing impairment and deaf symptoms. The main contribution of this work is the design of a system to convert ASL to English and vice versa.

Keywords: ASL; deep learning; image; translation; video



Citation: Avina, V.D.; Amiruzzaman, M.; Amiruzzaman, S.; Ngo, L.B.; Dewan, M.A.A. An AI-Based Framework for Translating American Sign Language to English and Vice Versa. *Information* **2023**, *14*, 569. <https://doi.org/10.3390/info14100569>

Academic Editors: Danilo Avola and Katsuhide Fujita

Received: 23 August 2023

Revised: 25 September 2023

Accepted: 12 October 2023

Published: 15 October 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Communication is an essential part of life, without which life would be very difficult. Each living being in the world communicates in their own way. We, as human beings, usually communicate by speaking a language. However, there is an exception; for example, people with deaf symptoms and hearing impairment. They use signs to communicate among themselves (i.e., deaf to deaf or deaf to impaired hearing). Over a period of time, these signs became a language. Just like all other languages, American Sign Language (ASL) also has its own syntax and semantics [1,2]. One must follow its syntax and semantics to communicate correctly and efficiently. Also, for communication to be successful, it is important to understand what is being communicated. Most people who do not have these disabilities are not aware of these signs, how to use them, or the meaning of the different signs. This could be because of lack of knowledge. As a result, they struggle to communicate with deaf and hearing-impaired people.

ASL has its own grammar and culture, which differ from place to place. Hence, there are many versions of sign language available in the world. French Sign Language (LSF), British Sign Language (BSL), and ASL are a few of the well-known sign languages. Based on the location, different signs are used to say different words. Therefore, it is also important to understand which signs to use in which area. For this research, we are focusing on ASL. ASL is a sign language used by deaf and hearing impaired people in the United States and Canada, devised in part by Thomas Hopkins Gallaudet and Laurent Clerc based on sign language in France [2,3]. It is a visual-gestural language used by approximately

500,000 deaf or hearing-impaired people in North America. For each letter of English grammar, ASL has a specific sign, as well as for different words. If we get to know this word sign mapping, it is easy to understand ASL.

There are some existing studies that focus on identifying this mapping. People in the past have developed wearable devices that help to identify ASL signs [4,5]. Also, using the Convolution Neural Network (CNN) model and deep learning methodologies, there are a few research studies exploring ways to identify ASL. Most of these studies mainly focused on identifying fingerspelling, which is nothing but identifying signs for the English alphabet [6–8]. However, ASL has a vast variety of signs for different words, and little work has been conducted on the identification of these signs.

We propose a method to identify word-level ASL using deep learning, the CNN model, and the rolling average prediction method. Rolling average or moving average is often used in CNN models to improve the prediction accuracy [9]. The CNN model is well suited to identify image data. An ASL video for a particular word is nothing but a series of image frames with different hand gestures [10]. Hence, we train the CNN model with images containing temporal and spatial changes of hand gestures. We use the trained model to predict the correct ASL word. The ResNet50 model is used as the base model to achieve this goal. The ASL videos are converted into image frames and are pre-processed before using them to train the model. It is equally important to enable hearing people with a way to communicate with the deaf people. To facilitate this, we also concentrated on producing an ASL fingerspelling video for speech content from them. We hosted a model in FastAPI. A user-friendly application is developed using the ReactJS framework which takes the user input either to produce the ASL fingerspelling video or to translate the ASL video into the English language. The user has to upload an audio or video file to the application dashboard. This file is transferred via Rest API to the model present in the FastAPI back-end. The API interprets the request and performs the appropriate translation using the trained model. The result of this translation is sent back to the application for the user's assistance.

We summarize the contributions of this research study in the following:

- We used the ResNet50 model and transfer learning concept to train our model to recognize and classify these words. We made use of rolling average prediction to recognize the temporal changes present in the video and recognize the word without any jitters in prediction.
- The proposed framework translates both ASL to English and English to ASL.
- To showcase the framework, we developed a web application, which makes use of the trained CNN model to translate ASL to English and vice versa.
- We generated a dataset consisting of images showing the hand gestures and facial expressions used by people to convey 2000 different words.

The rest of the paper is organized as follows. In Section 2, we discuss some existing works related to the classification of ASL. The details of the proposed approach are presented in Section 3. Information on the ASL-to-English/English-to-ASL translation application developed is provided in Section 4. In Section 5, the results and evaluation of the proposed approach is discussed. Finally, in Sections 6 and 7, we provide a discussion and conclusion, respectively.

2. Literature Review

Over the past several years, a good number of research studies has been conducted on interpreting ASL. Thad Starner et al. proposed sign language recognition based on Hidden Markov Models (HMM) [6]. This study used a camera to track hand movement to identify hand gestures. They extracted the features from the hand movements and fed them into four-state HMM to identify the ASL words in sentences. They evaluated their work by using a webcam or desk-mounted camera (second-person view) and a wearable camera (first-person view) for a 40-word lexicon. Similarly, Gaus and Wong [11] used two real-time hidden Markov model-based systems that were used to recognize ASL sentences by using

a camera to track the user's hands. The authors used word lexicon, and in their system they used a desk-mounted camera to observe the user's hands.

In [7], Qutaishat et al. proposed a method that does not require any wearable gloves or virtual markings to identify ASL. Their process is divided into two phases—feature extraction and classification. At the feature-extraction phase, features are extracted using Hough transformation from the the input images. These features are then passed as input to the neural network classification model. Their work was mainly focused on recognizing static signs. Several studies, such as [8,12–14] used the CNN model to classify ASL alphabets. In a separate study, Garcia et al. [8] used the transfer learning concept and developed the model using the Berkeley version of GoogLeNet. Most of these works concentrated on recognizing the ASL fingerspelling corresponding to the English alphabet and numbers [6,7,13]. Furthermore, Rahman et al. [12] used a CNN model to recognize ASL alphabets and numerals. Using a publicly available dataset, their study mainly focused on improving the performance of the CNN model. The study did not involve any human interaction to assess the accuracy of the approach. A similar work was found in [15], where the authors used an ensemble classification technique to show performance improvement. In a separate study, Kasapbasi et al. [16] used a CNN model to predict American Sign Language Alphabets (ASLA), and Bellen et al. [17] focused on recognizing ASL-based gestures during video conferencing.

In a study, Ye et al. [18] used a 3D recurrent convolutional neural network (3DRCNN) to recognize ASL signs from continuous videos. Moreover, they used a fully connected recurrent neural network (FC-RNN), which captured the temporal information. The authors were able to recognise ASL alphabets and several ASL words. In [13,18], the authors used 3D-CNN models to classify ASL. In [13], authors developed a 3D-CNN architecture which consists of eight layers. They used multiple feature maps as inputs for better performance. The five features which they considered are color-R, color-G, color-B, depth, and body skeleton. They were able to achieve better prediction percentages compared to the GMM-HMM model. In [7], Munib et al. used images of signers' bare hands (in a natural way). Their goal was to develop an automatic translation system for ASL alphabets and signs. This study used Hough transform and neural network to recognize the ASL signs.

In [18], authors proposed a hybrid model, and it consisted of the 3D-CNN model and the Fully Connected Recurrent Neural Network (FC-RNN). The 3D-CNN model learns RGB, motion, and depth channel whereas FC-RNN captures the temporal features in the video. They collected their own dataset consisting of sequence videos and sentence videos. They achieved 69.2% accuracy. However, the use of 3D-CNN is a resource-intensive approach. Jeroen et al. [19] proposed a hybrid approach to recognize sign language using statistical dynamic time wrapping for time wrapping and wrapped features are classified by separate classifiers. This approach relied mainly on 3D hand motion features. Mahesh et al. [20] tried to improve the performance of traditional approaches by minimizing the CPU processing time.

These existing previous works focus on building applications that enable communication between deaf people and hearing people [20]. However, creating an app requires a more precise design. One has to think of memory usage and other operations to enable a smooth user experience. Dongxu Li et al. [21] worked on gathering the word-level ASL Dataset and an approach to recognize them. In their work, they concluded that more advanced learning algorithms are needed to recognize the large dataset created by them. In [14,22], authors developed a means to convert from ASL to text. They used the CNN model to identify the ASL and then they converted the predicted label to text. They mainly concentrated on generating the text for fingerspelling instead of word-level signs. Garcia and Viesca [8], focused on classifying alphabet handshape correctly for letters a–k instead of all types of ASL alphabets. Another work presented in [23] detected ASL signs and converted to audio, and authors of [24] focused on constructing a corpus using the Mexican Sign Language (MSL).

After studying and understanding what has been achieved in existing studies, we first determined the goal for this study, which was to develop a framework to translate English to ASL and vice versa. We understood that not all deaf people know English, and several existing works focused on using CNN models, and improving computational performance. CNN models are a good choice for classifying ASL signs from image and video data. So, we experimented with a few CNN models before selecting the one that provided better results. For example, VGG16 [14], 3D-CNN [13], I3D [21], 3DRCNN [18], and ResNet50 (see Table 1 for more details). Among all, the ResNet50 model provided the best performance. Hence, the ResNet50 model was used for training and testing our research data. The model uses $224 \times 224 \times 3$ input images in four-stage processing. The next section presents the methodology used in this study, i.e., dataset collection, pre-processing, model training, model evaluation, and application development.

Table 1. Comparison of our work with others.

Models	VGG16 [14]	3D-CNN [13]	I3D [21]	3DRCNN [18]	ResNet50
Accuracy	98.7%	90.8%	89.92%	69.2%	95.31%
Application Developed	Yes	No	No	No	Yes

3. Methodology

ASL gesture identification is a challenging task because of the complexity of ASL’s combination of hand movements and facial expressions to represent words and sentences. We need to concentrate on both aspects to identify the signs properly. In one approach to ASL identification, people use wearable devices to capture hand movements and expressions. This approach is highly dependent on hardware components. With the advancement in machine learning techniques, deep learning in particular, there are many alternate ways to identify ASL gestures. Deep learning models like CNN, CNN-RNN, light-weight CNN, or 3D-CNN became the point of interest in ASL identification and classification. We chose to use the CNN model along with the rolling average prediction methodology. The detailed steps involved are shown in Figure 1. As described in the Figure, the whole process was divided into five main steps:

1. Dataset collection
2. Data pre-processing
3. Model training
4. Model evaluation
5. Application development

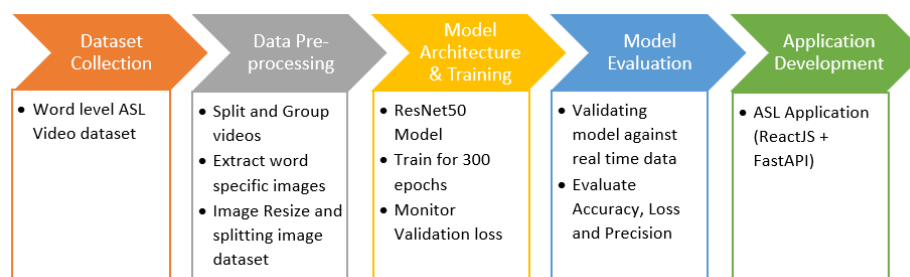


Figure 1. Overview of proposed methodology.

3.1. Dataset Collection

Obtaining a dataset to train the model is an important step in the deep learning approach. How well the model is trained also depends on the quality and quantity of the data available. For this work, we used one of the largest public ASL word level datasets prepared by [21]. This dataset was made available by its authors via Kaggle [25]. It consists

of videos showing the signs used in ASL for 2000 different words. After downloading the dataset, we grouped the videos on a per-word basis. Each word consists of a minimum of four videos and a maximum of seven videos. Each video spans for a few seconds, showing the hand movements and facial expressions signed by the user. Using these videos, we extracted frame-wise images by the OpenCV Python library showing the different hand positions and expressions of the user. We used these extracted images as data to train our model.

3.2. Data Pre-Processing

The extracted images include those which are common across all words. For example, images at the beginning and ending of the video showed the user's posture before the sign and after the sign. These images are removed from the collection. The neural network takes input the images of the same size. Therefore, all remaining images need to be resized to the same size before inputting them to the neural network [26]. The image resolution plays a very important role in classification accuracy. The higher the image resolution is, the lesser the shrinkage is going to be, and hence better accuracy can be achieved. However, keeping image resolution high comes with the cost of memory consumption and the speed at which the model is trained. Hence, choosing the right input size is very critical in model training. The dataset we used in this study consists of $400 \times 400 \times 3$ size images, so we could not use the input images to train and test the model as is. We chose to resize the images to $224 \times 224 \times 3$ pixels to balance both the image shrinkage issue and the memory consumption problem. The dataset was then split into training and testing datasets. We chose a 75–25 ratio to split the dataset, where 75% is the training set ratio and 25% is the testing set ratio.

3.3. Model Architecture and Training

Transfer learning is a method in machine learning in which a model developed for a particular task is reused as a basis for a similar task. We used this transfer learning method to develop our model. The purpose of this study was to develop a way to allow deaf and hearing people to communicate and reduce the barrier. Rather than proposing a new or improved CNN model, this study focused on translating spoken English to ASL instead of generating English texts. Using a transfer learning model is better than constructing a new model from scratch as the pre-trained model will already contain information from a similar dataset. We decided to use the ResNet50 model, which was pre-trained on image data. We chose the ResNet50 model because it has better accuracy and we can increase the number of network layers without trading off the accuracy (see Table 1 for details). We use images obtained from the ASL videos to train the model. Figure 2 shows the details of the developed model.

As shown in Figure 2, ResNet50 is used as the base model and the first layer of our model. To this ResNet50 model, we added the following layers:

- AveragePooling2D layer with pool size (7, 7);
- Flatten layer;
- Dense layer with 512 units which has a Rectified Linear Unit (ReLU) activation function;
- Dropout layer;
- Dense layer with 2000 units which has a softmax activation function.



Figure 2. Model details: showing the steps used to train the model.

Model Training

Although the dataset contained signs for 2000 different words, we trained the model with multiple numbers of words grouped together. We formed 15-, 50-, 100-, 500-, 1000-,

and 2000-word batches and trained the model separately for each group. For every group, the images were split into training and testing sets as mentioned in the data pre-processing step. We used the batch size of 32 for our model training. Batch size plays an important role in overall training of the model, per-epoch training time, and model quality. Also, the batch size is dependent on the amount of data available. We chose a batch size in mini batch mode because of a quicker learning rate.

The learning rate of a model corresponds to the rate at which the model meets the solution. It is important to set the optimal learning rate as a setting since a smaller value may take more time to reach the required output. On the other hand, setting a bigger value may cause the gradient descent to diverge and the model may not reach the optimal solution. Upon carefully inspecting the model's performance (i.e., using the ROC curve (receiver operating characteristic curve) based on true positive rate (TPR) against the false positive rate (FPR)), we chose 0.0001 as the learning rate for our model. We chose SGD optimizer to make our model training quicker.

After analyzing our model performance using different batch sizes, I noticed that a batch size of 1260 was optimal. There were a total of 122,895 trainable parameters. We trained the model for 150 epochs. It is important to pass the input images through different steps before running the model on the whole dataset. The step size can be easily determined by using the equation below (see Equation (1)):

$$\text{number of steps} = \lceil \frac{\text{number of images}}{\text{batch size}} \rceil \quad (1)$$

These steps help to ensure that the model will not have errors after completion of each epoch as it helps to make sure that the input image dimensions are the same as the model input layer. We rounded the number to the nearest integer as steps for each epoch. In this study, each epoch consists of 97 steps. The training each epoch lasted for a couple of hours. After each epoch, the model was then saved, so that we can use it in our ASL translation application. We had saved the labels of each word using a "lb.pickle" file which we would be using as a reference at the time of predication.

We conducted the initial training on a computer with 16 GB RAM, and 1.8 GHz CPU (Intel (R) Core (TM) i710610U). The process took almost a week to complete. The computer uses a Windows 10 Operating system. The initial training helped us to understand the prospect of the data and its outcomes for our study. Then, the final training process took place on an AMD EPYC 7452 (a total of 64 cores) CPUs and with a 143 GB memory server computer which was using a Linux (Ubuntu 18.04.6 LTS) operating system.

3.4. Application Development

We developed a two-way application for ASL translation. The application is developed as a web-based application, so that users can upload their signed (i.e., ASL) video to the application and see what was signed. If an ASL video is uploaded, the application will translate from ASL to English words. On the other hand, if an audio file containing a speech is uploaded, the application will generate a corresponding ASL fingerspelling video. We have made use of the model previously trained to do the first translation, whereas the second translation is basically mapping the ASL fingerspelled image to the alphabet. The detailed explanation of the application is given in Section 4. The application is developed as a web-based application using ReactJS and FastAPI. The ReactJS is a JavaScript-based open-sourced library—good for routing applications. The FastAPI is a Python library that supports web applications, and it is considered to be high-performing library compared to other traditional Python libraries for web support.

4. ASL to English and English to ASL Conversion Application

Using ReactJS and FastAPI, we developed an application to translate ASL to English and English to ASL based on the type of request received. Figure 3 shows the screenshot of the application dashboard while converting ASL to English/English to ASL.

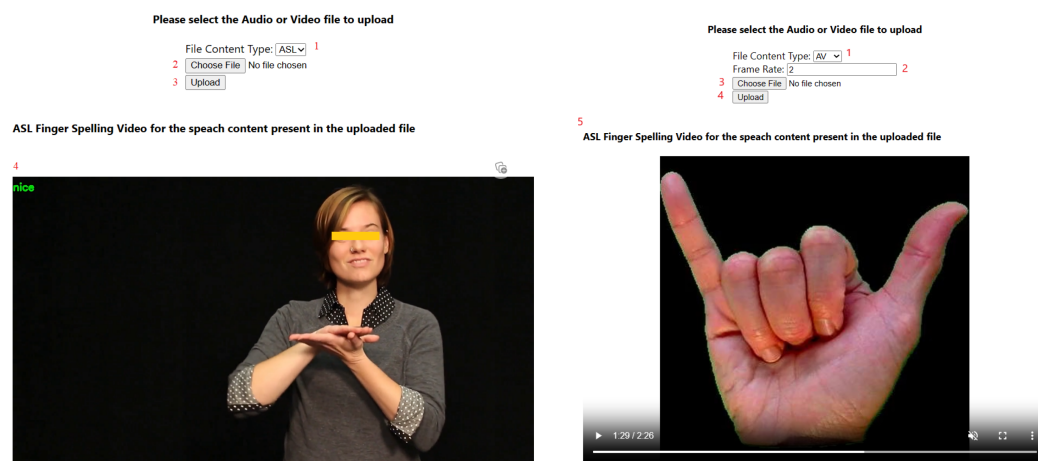


Figure 3. Screenshot of ASL–English–ASL application dashboard: (left) ASL-to-English Translation, and (right) English-to-ASL Translation.

The client-side interactive web application was developed using the ReactJs framework. The user is provided with options to choose the type of translation needed, i.e., a user can select either ASL-to-English translation or English-to-ASL translation (Annotation 1 in Figure 3(left,right)). Based on the selection of additional options, a frame rate is displayed for the user to select (Annotation 2 in Figure 3(right)). There is a file upload link provided for the user to upload the video or audio to be translated (Annotation 2 in Figure 3(left) and Annotation 3 in Figure 3(right)). After selecting the file, users can submit it to the server side application for translation using an upload button (Annotation 3 in Figure 3(left) and Annotation 4 in Figure 3(right)). There is a placeholder for the translated video to play (Annotation 4 in Figure 3(left) and Annotation 5 in Figure 3(right)).

When the user selects ASL to English translation, the translated video played will be the same video uploaded along with the English word added on the top left corner of the video, as shown in Figure 3(left). In the case of English to ASL translation, a video containing the fingerspelling of the words is played, as shown in Figure 3(right). This video also contains the original audio file embedded with it. Also, by using the frame rate option, the user can control the speed of the fingerspelling transition in the video.

Using the Rest API calls, users select information and the file details which need to be translated are transferred to the server side application. We used the FastAPI framework to host the back-end application, which uses the trained model to predict the signs and translate them to corresponding English words. The code is written in Python and the application is hosted using a Uvicorn server. Once the back-end application receives the request from the front-end dashboard, it reads the type of translation needed and executes the appropriate workflow.

When it comes to the translation from ASL to English, first it obtains the frame-level images of the video and then resizes to $224 \times 224 \times 3$ pixels. Using the model and `lb.pickle` file, it identified the word corresponding to that image. Using a rolling averaging method, it predicts the word corresponding to the whole image sequence and then it appends the corresponding word to the top left corner of the video and sends the updated video back to the application dashboard using Rest API response. If the translation requested is English to ASL, the text transcript from the audio file is fetched and the corresponding ASL fingerspelling for each spelling is fetched and a video is constructed using these images and the original audio file is attached to the video for users' reference, and this constructed video is sent back to the user using Rest API response.

5. Results and Evaluation

We trained all the dataset sub-groups with the same set of parameters. Figure 4(left, right) gives the accuracy and loss of training and validation sets. We used cross-validation

techniques to check if our model was over-fitted. A Receiver Operating Characteristic (ROC) [27] curve can be used to analyze the performance of a neural network model. It helps to stop the training epoch as the model accuracy tends to slow down and chances of over-fitting increases after a certain number of epochs (see Figure 4(left, right) for more details). In this study, we used ROC instead of network disturbance analysis.

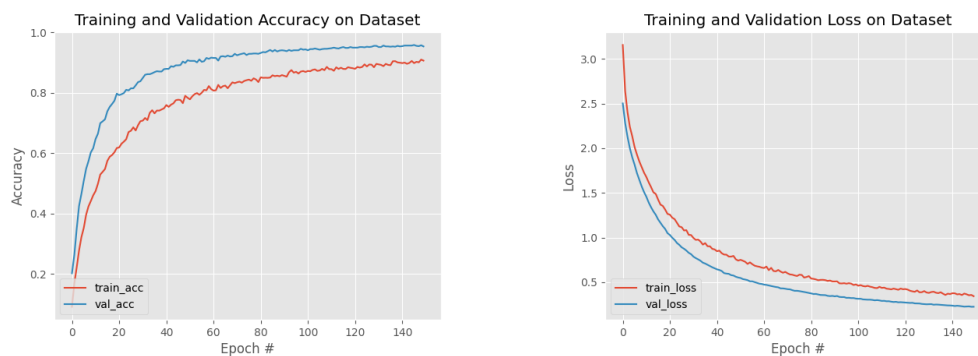


Figure 4. Accuracy and loss plot for our model trained with 15-word dataset: (left) Accuracy for 15 words, and (right) Loss for 15 words.

Figure 4(left) gives the model accuracy on the training and validation datasets. As we can see, the accuracy increased quickly in the beginning and increased steadily after 40 epochs. We were able to achieve 95.31% accuracy with the validation dataset, whereas with the training dataset we achieved 90.61% accuracy. Again, Figure 4(right) gives information on model loss on training and validation datasets. We used the categorical cross-entropy loss function. This means we trained the model to provide probability over all classes for each image that is going to be predicted. As seen from the graph, the loss dropped quickly in the initial epochs and with each epoch the loss reduced. The model had 0.37% training loss and 0.25% validation loss.

We also compared our method with other related works. Table 1 provides a high level overview of work carried out in this regard by other researches compared to ours.

We can see that very few concentrated on providing a means to convert classified signs to text. As shown in the table, [14] built an application, however they used the dataset consisting of signs for English alphabets and three other signs for space, delete, and nothing. We used the video dataset generated by [21] to generate our dataset. As shown in the table they were able to achieve an accuracy of 89.92% for 100 words. Refs. [13,18] used 3D models for the classification. Their accuracy is lower compared to our model's accuracy.

6. Discussion

In the beginning, we thought of developing a 3D-CNN model to classify the ASL video. However, upon investigating in depth, we found that it requires more computational power to implement. We thought of exploiting the basic nature of videos, which is "Video is made of n - number of frames(images)" and used the CNN model to classify ASL videos.

Initially, we trained the model with all the images obtained from the videos. This means all images that were common across words were included. Because of this, the model was becoming confused with such images and was not predicting correctly. We then removed all the common and unwanted images, so that the model was trained on unique images corresponding to the word. The prediction became much more accurate this time. We also tried training the model by removing the background of the image to make all the images uniform; however, removing the background did not increase the model's accuracy considerably.

There was a fluctuation in prediction when we passed an ASL video to predict. This was because we developed a CNN model to predict images. When we passed the video to predict, the frames of the video were obtained and the prediction was actually happening

on the images, which do not hold any information about the previous or the next frame. Hence, we made use of the rolling averaging concept in prediction to hold the details of previous frames so that the prediction is accurate across all the frames. It is important to note that we randomly selected the training and testing datasets, but used the same training and testing datasets for all models. This was done to make sure the obtained performance results are comparable.

Ablation study is a process in machine learning and deep learning models to find out the best possible selection of layers and/or training data inputs to obtain the better outputs [28,29]. This plays a major role in determining the types of layers and training weights to keep. However, this process is very important if the research is about improving the performance of models or comparing the output of different models. The focus of our study was to obtain the translation of ASL using pre-trained model ResNet. The model used is already proven to work well with the ImageNet dataset, which is like our dataset. As our focus was mainly on translating ASL with the proven best working model, we found that an ablation study was not necessary at this point for our study.

7. Conclusions and Future Work

We trained and implemented a CNN classifier for ASL translation using a transfer learning concept. We used a rolling averaging prediction technique to remove the fluctuations in the prediction. Using ReactJS and FastAPI, we developed a web-based application to translate American Sign Language, and to showcase our work and users to see the results or our framework. Evidence obtained from the study indicated 95% validation accuracy. Accommodating deaf people and help them to communicate with hearing people was the focus of the proposed framework. Therefore, this framework contributed in translating English to ASL, as we assume that most deaf people in the USA know ASL, and could not assume that they know English.

Currently, we are translating the English words to ASL fingerspelling videos. We would like to explore how we can translate the words to corresponding signs used instead of fingerspelling. Additionally, so far, we have considered only the ResNet50 model, so we would like to explore other models like InceptionV3, and VGG, which have proven effective in Image classification. During the training process, we did not consider disturbance analysis. In the future, a disturbance analysis should be considered for better performance.

So far, we have a dataset for only 2000 words. We would like to expand our dataset to include as many signs as possible to accommodate all possible words used in ASL. We are planning to provide an option in the application dashboard for the user to contribute to our dataset by uploading ASL videos.

Author Contributions: Conceptualization: M.A. and S.A.; methodology: M.A. and S.A.; software: V.D.A.; validation: V.D.A., M.A., S.A., L.B.N. and M.A.A.D.; formal analysis: V.D.A., M.A., S.A. and M.A.A.D.; investigation: V.D.A., M.A. and S.A.; resources: V.D.A., M.A. and S.A.; data curation: V.D.A.; writing—original draft preparation: V.D.A., M.A. and S.A.; writing—review and editing: V.D.A., M.A., S.A. and M.A.A.D.; visualization: V.D.A.; supervision: M.A. and S.A.; project administration: M.A. and S.A.; funding acquisition: M.A., S.A. and M.A.A.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partly funded by the Pennsylvania State System of Higher Education (PASSHE) Faculty Professional Development Council (FPDC) grant.

Data Availability Statement: All data publicly available, we downloaded the data from Kaggle.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kuhn, J.; Aristodemo, V. Pluractionality, iconicity, and scope in French Sign Language. *Semant. Pragmat.* **2017**, *10*, 1–49. [CrossRef]
2. Liddell, S.K. *American Sign Language Syntax*; Walter de Gruyter GmbH & Co KG: Berlin, Germany, 2021; Volume 52.
3. Vicars, W.G. ASL—American Sign Language. Available online: <https://www.lifefprint.com/asl101/pages-layout/lesson1.htm> (accessed on 1 March 2023).

4. Kudrinko, K.; Flavin, E.; Zhu, X.; Li, Q. Wearable sensor-based sign language recognition: A comprehensive review. *IEEE Rev. Biomed. Eng.* **2020**, *14*, 82–97. [[CrossRef](#)]
5. Lee, B.; Lee, S.M. Smart wearable hand device for sign language interpretation system with sensors fusion. *IEEE Sens. J.* **2017**, *18*, 1224–1232. [[CrossRef](#)]
6. Starner, T.; Weaver, J.; Pentl, A. Real-time american sign language recognition using desk and wearable computer based video. *IEEE Trans. Pattern Anal. Mach. Intell.* **1998**, *20*, 1371–1375. [[CrossRef](#)]
7. Munib, Q.; Habeeb, M.; Takruri, B.; Al-Malik, H.A. American sign language (ASL) recognition based on Hough transform and neural networks. *Expert Syst. Appl.* **2007**, *32*, 24–37. [[CrossRef](#)]
8. Garcia, B.; Viesca, S.A. Real-time American sign language recognition with convolutional neural networks. *Convolutional Neural Netw. Vis. Recognit.* **2016**, *2*, 8.
9. Kurian, E.; Kizhakethottam, J.J.; Mathew, J. Deep learning based surgical workflow recognition from laparoscopic videos. In Proceedings of the 2020 5th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 10–12 June 2020; pp. 928–931.
10. Dabre, K.; Dholay, S. Machine learning model for sign language interpretation using webcam images. In Proceedings of the 2014 International Conference on Circuits, Systems, Communication and Information Technology Applications (CSCITA), Mumbai, India, 4–5 April 2014; pp. 317–321.
11. Gaus, Y.F.A.; Wong, F. Hidden Markov Model-Based gesture recognition with overlapping hand-head/hand-hand estimated using Kalman Filter. In Proceedings of the 2012 Third International Conference on Intelligent Systems Modelling and Simulation, Kota Kinabalu, Malaysia, 8–10 February 2012; pp. 262–267.
12. Rahman, M.M.; Islam, M.S.; Rahman, M.H.; Sassi, R.; Rivolta, M.W.; Aktaruzzaman, M. A new benchmark on american sign language recognition using convolutional neural network. In Proceedings of the 2019 International Conference on Sustainable Technologies for Industry 4.0 (STI), Dhaka, Bangladesh, 24–25 December 2019; pp. 1–6.
13. Huang, J.; Zhou, W.; Li, H.; Li, W. Sign language recognition using 3d convolutional neural networks. In Proceedings of the 2015 IEEE international conference on multimedia and expo (ICME), Turin, Italy, 29 June–3 July 2015; pp. 1–6.
14. Thakar, S.; Shah, S.; Shah, B.; Nimkar, A.V. Sign Language to Text Conversion in Real Time using Transfer Learning. In Proceedings of the 2022 IEEE 3rd Global Conference for Advancement in Technology (GCAT), Bangalore, India, 7–9 October 2022; pp. 1–5.
15. Chung, H.X.; Hameed, N.; Clos, J.; Hasan, M.M. A Framework of Ensemble CNN Models for Real-Time Sign Language Translation. In Proceedings of the 2022 14th International Conference on Software, Knowledge, Information Management and Applications (SKIMA), Phnom Penh, Cambodia, 2–4 December 2022; pp. 27–32.
16. Kasapbaşı, A.; Elbushra, A.E.A.; Omar, A.H.; Yilmaz, A. DeepASLR: A CNN based human computer interface for American Sign Language recognition for hearing-impaired individuals. *Comput. Methods Progr. Biomed. Update* **2022**, *2*, 100048. [[CrossRef](#)]
17. Enrique, M.B., III; Mendoza, J.R.M.; Seroy, D.G.T.; Ong, D.; de Guzman, J.A. Integrated Visual-Based ASL Captioning in Videoconferencing Using CNN. In Proceedings of the TENCON 2022-2022 IEEE Region 10 Conference (TENCON), Hong Kong, 1–4 November 2022; pp. 1–6.
18. Ye, Y.; Tian, Y.; Huenerfauth, M.; Liu, J. Recognizing american sign language gestures from within continuous videos. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Salt Lake City, UT, USA, 18–23 June 2018; pp. 2064–2073.
19. Lichtenauer, J.F.; Hendriks, E.A.; Reinders, M.J. Sign language recognition by combining statistical DTW and independent classification. *IEEE Trans. Pattern Anal. Mach. Intell.* **2008**, *30*, 2040–2046. [[CrossRef](#)]
20. Mahesh, M.; Jayaprakash, A.; Geetha, M. Sign language translator for mobile platforms. In Proceedings of the 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Udupi, India, 13–16 September 2017; pp. 1176–1181.
21. Li, D.; Rodriguez, C.; Yu, X.; Li, H. Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Snowmass Village, CO, USA, 1–5 March 2020; pp. 1459–1469.
22. Patil, P.; Prajapat, J. Implementation of a real time communication system for deaf people using Internet of Things. In Proceedings of the 2017 International Conference on Trends in Electronics and Informatics (ICEI), Tirunelveli, India, 11–12 May 2017; pp. 313–316.
23. Santon, A.L.; Margono, F.C.; Kurniawan, R.; Lucky, H.; Chow, A. Model for Detect Hand Sign Language Using Deep Convolutional Neural Network for the Speech/Hearing Impaired. In Proceedings of the 2022 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS), Jakarta, Indonesia, 16–17 November 2022; pp. 118–123.
24. Trujillo-Romero, F.; Garcia-Bautista, G. Mexican Sign Language Corpus: Towards an automatic translator. *ACM Trans. Asian-Low-Resour. Lang. Inf. Process.* **2023**, *22*, 1–24. [[CrossRef](#)]
25. Kaggle. Available online: <https://www.kaggle.com/> (accessed on 13 June 2023).
26. Hashemi, M. Web page classification: A survey of perspectives, gaps, and future directions. *Multimed. Tools Appl.* **2020**, *79*, 11921–11945. [[CrossRef](#)]
27. Lee, C.S.; Baughman, D.M.; Lee, A.Y. Deep learning is effective for classifying normal versus age-related macular degeneration OCT images. *Ophthalmol. Retin.* **2017**, *1*, 322–327. [[CrossRef](#)] [[PubMed](#)]

28. Du, L. How much deep learning does neural style transfer really need? An ablation study. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Snowmass, CO, USA, 1–5 March 2020; pp. 3150–3159.
29. Mehta, T.I.; Heiberger, C.; Kazi, S.; Brown, M.; Weissman, S.; Hong, K.; Mehta, M.; Yim, D. Effectiveness of radiofrequency ablation in the treatment of painful osseous metastases: A correlation meta-analysis with machine learning cluster identification. *J. Vasc. Interv. Radiol.* **2020**, *31*, 1753–1762. [[CrossRef](#)] [[PubMed](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.