Sign Language Translator

Abdelrahman Mohamed Hanafi   
Scientific computing  
Faculty of Computer & Information SciencesCairo, Egypt  
abdelrahman20191700346@cis.asu.edu.eg

Eslam Mohamed Mohamed  
 *Scientific Computing*  
Faculty of Computer & Information SciencesCairo, Egypt  
eslam20191700112@cis.asu.edu.eg Abdelrahman Ali Mohamed   
Scientific computing  
Faculty of Computer & Information SciencesCairo, Egypt  
abdelrahman20191700342@cis.asu.edu.eg

*Abstract*: *The project aims to develop a cost-effective and easy-to-use computer-based interpreter system to bridge the communication gap between signers and non-signers of Argentina Sign Language (ASL). The system will convert ASL signs into text and allow users to learn ASL interactively. A new ASL dataset consisting of 64 words, each word has 10 videos, and a Deep learning model was chosen for the task. The system achieved high accuracy in sign recognition, with the word recognition model achieving 97% accuracy and the word recognition model achieving 94.5% accuracy on test data.*

Keywords: Sign language, GRU, CNN, Deep learning

# Introduction

Sign language has a long history dating back to ancient times, but it wasn't until the 18th century that it began to be recognized as a legitimate form of communication. American Sign Language (ASL) emerged in the United States in the early 19th century and has since evolved into a complete and complex language with its own unique grammar and syntax. Today, there are many sign languages used around the world, and sign language interpretation approaches try to automatically translate sign languages using vision techniques. Two key tasks involved in sign language interpretation are sentence-level sign language recognition and word-level sign language recognition. The word-level sign recognition task is extremely difficult, and the small-scale datasets currently available do not adequately capture the distinctions between signs. The authors of this paper gathered a large-scale dataset of word-level signs and their related annotations to train sign recognition models that can be used in typical situations without specialized tools. They only gathered videos of signers in near-frontal views to create a high-quality, sizable dataset. The paper discusses the problem of action recognition in videos and the goal of implementing a video action recognition system. A CNN-GRU architecture is used for this project, where the features of the video are extracted using a convolutional neural network and the video is categorized using an GRU neural network. The decision to use GRU was made based on its ability to handle long-term dependencies in sequential data, effectively learn from past experiences, handle input sequences of varying lengths, and handle multi-class classification problems. Overall, the utilization of GRU was the optimal choice for achieving the desired outcomes in this project. The motivation behind creating a sign language application is to help the deaf communicate better and the impact it can have on their lives by promoting inclusivity and independence. It also highlights the potential of deep learning to improve the accuracy and impact of the project by analyzing vast amounts of complex data and identifying patterns. The authors are eager to explore how they can integrate deep learning into their project to drive positive change in their respective industries. The characteristics that could be helpful in creating a successful sign language translator, including innovation, attention to detail, patience, compassion, collaboration, determination, adaptability, empathy, persistence, and a drive to make a positive impact on the lives of others. It also discusses the development of a deep learning-based video extraction application that can autonomously process video data, identify key events, extract critical information, and present it in an easily interpretable format. The block explains the normal format of video data, the challenges of combining multiple videos into a mini-batch, and how convolution must span both temporal and spatial dimensions to account for multiple frames within the input.

# Literature Review

In this section, we will be exploring different aspects of the topic at hand through a literary review, followed by a survey covering the latest research trends.

## Related Work

**2D Convolution with Recurrent Neural Networks Model:** CNN (VGG 16) are used to extract spatial features of input images while RNN (GRU) are employed to capture the long-term temporal dependencies among inputs. The proposed a reliable ensemble model in this paper. proposed model produced results of 63.95% on the WLASL100 dataset.

**3D Convolutional Networks:**

Establish not only the holistic representation of each frame but also the temporal relationship between frames using Inception network trained on ImageNet. The results show that our model achieves 89.92% on the WLASL100 dataset.

**Pose Based Temporal Graph Neural Networks:**

models the spatial and temporal dependencies of the pose sequence. We used a holistic representation of the trajectories of body Key Points. The results show that our model achieves 87.60% on the WLASL100 dataset.

## Theoretical background

Sign language translation is a rapidly growing field driven by advancements in computer vision and machine learning. The goal is to create systems that interpret gestures and movements of sign language users and translate them into spoken or written language. Researchers are working on computer vision algorithms, deep learning models, and natural language processing (NLP) techniques to accurately detect and recognize user signs. Techniques include hand tracking, gesture recognition, pose estimation, and deep learning models like CNNs and RNNs. Additionally, researchers are exploring the use of sign language corpora and databases to improve translation accuracy.

## Mathematical Background

Sign language translation requires expertise in computer vision, machine learning, NLP techniques, linear algebra, calculus, probability theory, and statistics. These mathematical concepts are crucial for deep learning models, optimization algorithms, and evaluation metrics development.

## Algorithms and Techniques

1. Hand tracking: This detects the user’s hand position and movement.
2. Gesture recognition: This uses machine learning algorithms to recognize specific gestures.
3. Pose estimation: This estimates 3D body and limb positions.
4. Machine translation: This converts signs into written language.

## Conclusion

Sign language translation is a promising field for improving communication between deaf and hearing individuals. Computer vision has made significant progress in accurately interpreting sign language and translating it into written language. However, challenges remain, and further research is needed to enhance accuracy, efficiency, and usability.

# Sysem Architecture

A picture containing text, screenshot, diagram, line

Description automatically generated

Figure 1: System Architecture

## System Architecture

1. User records a video.
2. Take this video as an input to the system.
3. Extract Key points or Joints by using Media pipeline.
4. Take those key points to GRU model.
5. Probabilities for each word.
6. Thresholding to extract the exact word.
7. Output is a text.

Dataset contains 64 words; each word has 50 videos; each video consists of 35 frames. Media pipeline is used to extract key points from the current frame. Then, the GRU model extracts the features from those key points and predicts the word based on the information it gathered.

A picture containing text, diagram, screenshot, font

Description automatically generated

Figure 2: Activity Diagram

* The user enters the introduction website.
* Then, the user can login if has an account.
* If not, he will sign up for the home page.
* After that, he browses it or can click the button called translate to put his video.
* Drop his video on the website and the website passes it to Flask.
* Then Flask makes preprocessing on video and extracts key points from it, pass key points to the model.
* Thresholding the result to extract the exact word.
* Give a text to website to display it for user.

## What is Flask and what can Flask do?

Flask is a lightweight Python web framework designed for developers to build web applications quickly and efficiently. It offers a minimalistic design, highly customizable tools, and libraries, making it easy to learn and use. Flask's simplicity, flexibility, and ease of use make it a popular choice for developers seeking a lightweight, user-friendly solution.

Flask is a lightweight Python web framework for building quick, easy web applications and APIs with minimal setup and configuration. It offers flexibility and a simple, flexible approach to Python development. Here are some of the things that Flask can do:

* Routing: Flask enables developers to define URL routes for web applications, mapping incoming requests to appropriate functions or views.
* Templating: Flask supports Jinja2, a flexible templating engine for dynamic HTML content generation.
* Request and Response Handling: Flask simplifies handling incoming requests and generating responses, supporting various content types and response codes.
* Session Management: Flask supports managing user sessions, enabling developers to store and retrieve data across multiple requests.
* Authentication and Authorization: Flask offers tools and extensions for user authentication and authorization, supporting popular protocols like OAuth and OpenID.
* Database Integration: Flask integrates seamlessly with various databases like SQLite, PostgreSQL, and MySQL.
* Extension Ecosystem: Flask's active extension ecosystem offers diverse third-party extensions for database integration, authentication, and authorization, fostering a vibrant and active community.

A diagram of a person

Description automatically generated with low confidence

Figure 3: Website Architecture

User life cycle to enter Website:

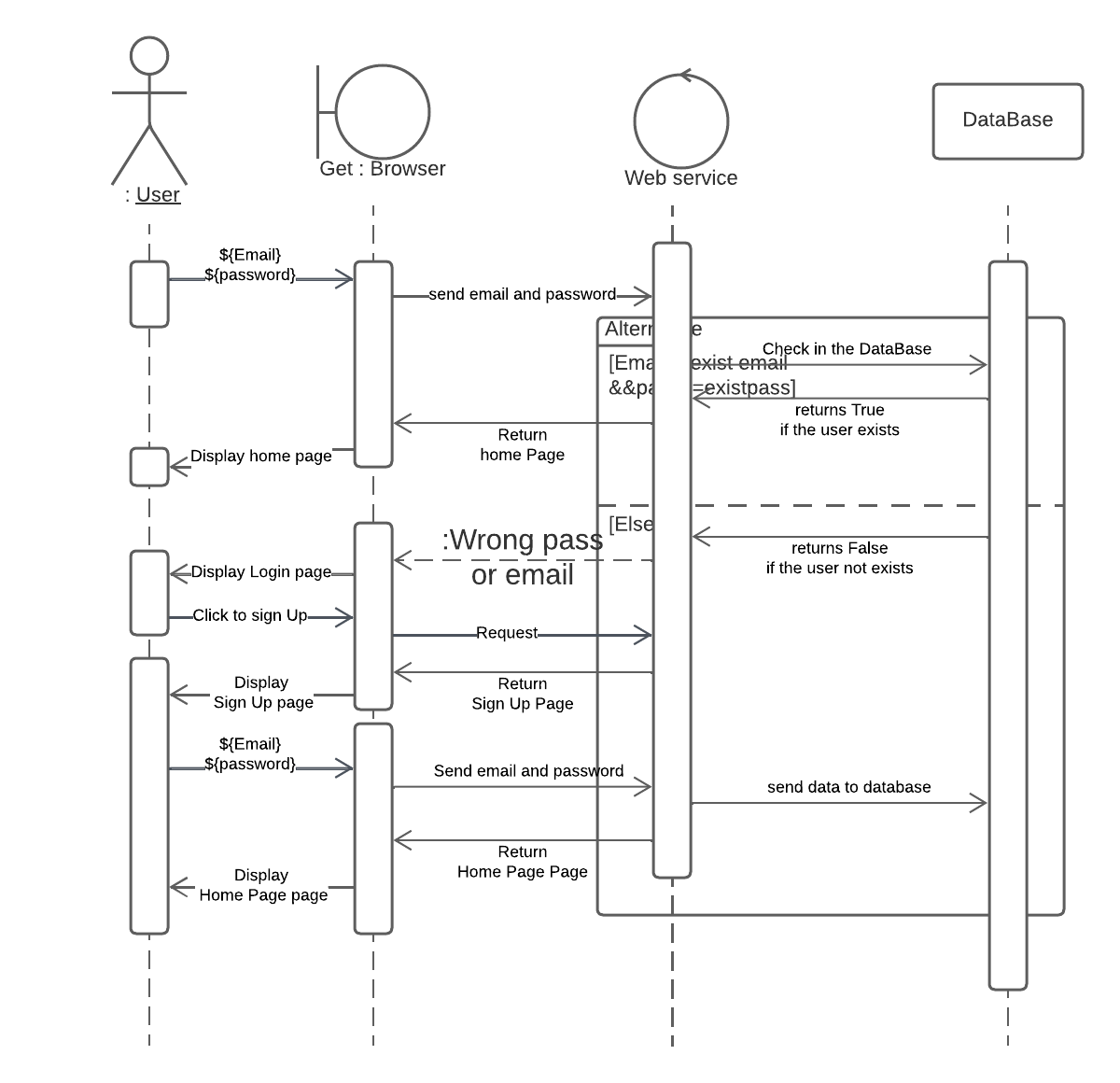
****

Figure 4: Sequence Diagram

# Sign Language Datasets

LSA64 is a dataset for Argentinian Sign Language (LSA) [21], he signs database for the Argentinian Sign Language, created with the goal of producing a dictionary for LSA and training an automatic sign recognizer, includes 3200 videos where 10 non-expert subjects executed 5 repetitions of 64 different types of signs. Signs were selected among the most used ones in the LSA lexicon, including both verbs and nouns.

## Dataset Collection

The database was recorded in two sets. In the first one, 23 one-handed signs were recorded. The second added 41 signs, 22 two-handed and 19 one-handed as well. The first recording was done in an outdoors environment, with natural lighting, while the second in an indoors environment, with artificial lighting, to provide differences in illumination between signs. Subject 10 of the first recordings was unavailable for the second set of recordings and was replaced by another subject. This change in no way diminishes the utility of the dataset, since the set of classes recorded in the first session is disjoint from the one recorded in the second session. In both sets of recordings, subjects wore black clothes and performed the signs standing or sitting, with a white wall as a background. To simplify the problem of hand segmentation within an image, subjects wore fluorescent-colored gloves. These substantially simplify the problem of recognizing the position of the hand and performing its segmentation, and remove all issues associated to skin color variations, while fully retaining the difficulty of recognizing the handshape. Table 4.1: Argentina Signs 38 Each sign was executed imposing few constraints on the subjects to increase diversity and realism in the database. All subjects were non-signers and righthanded, were taught how to perform the signs during the recording session by showing them a video of the signs as performed by one of the authors and practiced each sign a few times before recording. The camera employed was the same in both sets of recording (Sony HDR-CX240). The tripod was placed 2m away from the wall at a height of 1.5m. Marks on the floor were used to indicate subjects where to position themselves. The resolution of the videos is 1920 by 1080, at 60 frames per second.

## Samples

Samples Sample snapshots of the dataset. The images on the left (first column) are from the first set of recordings.

A collage of people wearing gloves

Description automatically generated with medium confidence

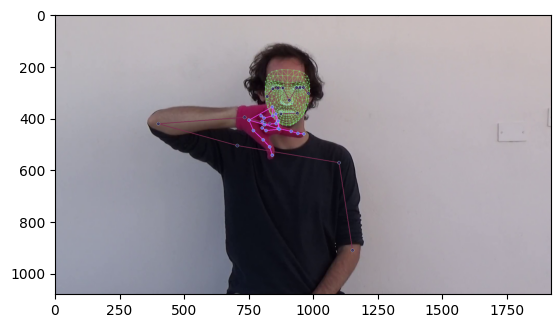
# Approaches

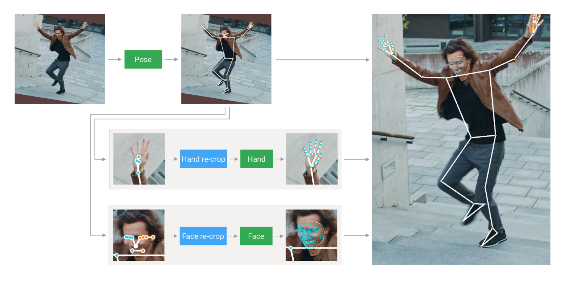
## Preprocessing

1. Resampling:

Not all cameras have the same fps “frames per seconds”, and not all words take the same amount of time to be done. so we need to choose a fixed number of frames for each video.

1. MediaPipe

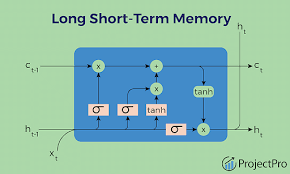




We used Media-Pipe holistic model to get pose information for each frame in video, Media-Pipe holistic model is Real-time, simultaneous perception of human pose, face landmarks and hand tracking on mobile devices can enable a variety of impactful applications, such as fitness and sport analysis, gesture control and sign language recognition, augmented reality effects and more. Media-Pipe, an open-source framework designed specifically for complex perception pipelines leveraging accelerated inference (e.g., GPU or CPU), already offers fast and accurate, yet separate, solutions for these tasks. Combining them all in real-time into a semantically consistent end-to-end solution is a uniquely difficult problem requiring simultaneous inference of multiple, dependent neural networks.

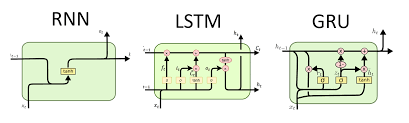
## Feature extraction

1. LSTM



LSTM is a type of recurrent neural network (RNN) that can handle sequential data and long-term dependencies. It has a special structure with four layers: a cell state, an input gate, an output gate, and a forget gate. These layers allow LSTM to control the flow of information and to remember or forget the relevant parts of the previous states. LSTM can overcome the problem of vanishing or exploding gradients that affect simple RNNs. LSTM can be used for various tasks such as time series, speech, and text processing. LSTM can learn long-term dependencies and capture complex patterns in sequential data, while simple RNN struggles to retain information from previous time steps.

1. GRU VS LSTM and why we choose GRU Model



LSTM and GRU are two types of recurrent neural networks (RNNs) that can handle sequential data and long-term dependencies. They both use gates to control the flow of information and gradients inside the network. However, they have some differences in their architecture and performance.

The main differences between LSTM and GRU are LSTM can store information for longer periods of time than GRU, because it has more control over what to keep or discard in the cell state. GRU is simpler and faster than LSTM, because it has fewer parameters and computations. The performance of LSTM and GRU depends on the task, the data, and the hyperparameters. Generally, LSTM is more powerful and flexible than GRU, but it is also more complex and prone to overfitting. GRU is faster and more efficient than LSTM, this is why in this project we preferred GRU over LSTM. Media-Pipe holistic poses are the preprocessing method we use in the latest model because poses are the closest features to the domain of the sign language recognition problem. For our project we don’t use the z component for all landmark, and we also don’t use all face landmarks, but we use just ten because we are more interested in the head position does not face details like mouse and nose, this also reduce overfitting, speed up training process and use less memory.

After extracting poses from each frame in a video we normalize the result using Z-score normalization, this also reduces overfitting and speeds up training time.

## Classification

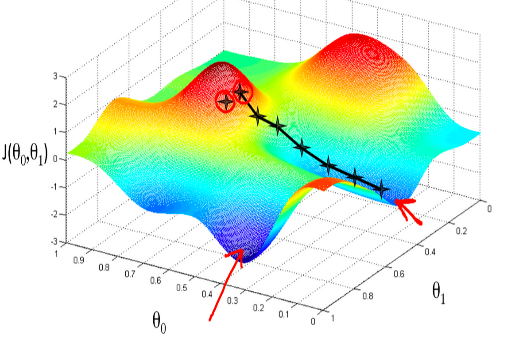
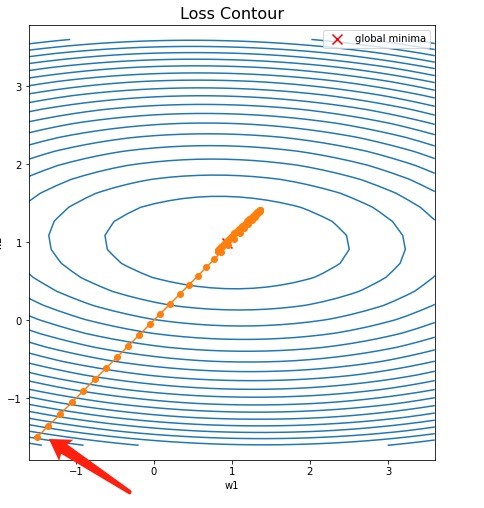
SoftMax is a function that converts a vector of real numbers into a vector of probabilities that sum to one. It is often used in machine learning models to produce output probabilities for multi-class classification problems. SoftMax can be defined as follows:

SoftMax(x) = exp(x) / sum(exp(x))

where x is the input vector, exp is the exponential function, and sum is the summation operator. The SoftMax function has some desirable properties, such as being differentiable, invariant to scaling, and maximizing the entropy of the output distribution. For classification we take the output from GRU as the input for the SoftMax layer which output a probability distribution for all the words in the data set then we take the word with the highest probability as the final output

## Training process

1. ADAM optimizer



ADAM optimizer is an optimization algorithm that can be used instead of the classical stochastic gradient descent procedure to update network weights iterative based on training data. ADAM optimizer was presented by Diederik Kingma from OpenAI and Jimmy Ba from the University of Toronto in their 2015 ICLR paper titled “Adam: A Method for Stochastic Optimization “. ADAM optimizer combines the best properties of the AdaGrad and RMSProp algorithms to provide an optimization algorithm that can handle sparse gradients on noisy problems. ADAM optimizer is computationally efficient, has little memory requirement, invariant to diagonal rescaling of gradients, and is well suited for problems that are large in terms of data or parameters.

1. Training

We split the data into train, test and validation (80,10,10) using SciKit-learn built-in method, we train our model on the training set. We implemented the model using TensorFlow Keras and use the built-in fit “train or optimize” method, we used mini batch optimization “128 sample per batch” with ADAM optimizer Training and testing was done on Google Collab

## Result

Model scored 97% on train set 95% on validation set and 94.5% on test set We also created our own small test data set and achieved 95% accuracy.

# Conclusion

In this paper, we develop a sign language translator is a significant achievement in bridging the communication gap between the hearing and non-hearing communities. The approach utilized an Argentina dataset to train an GRU model, which extracted key points from sign language videos to accurately predict and translate sign language gestures into words or sentences. The approach streamlines the translation process and improves system efficiency by reducing the amount of data that needs to be processed and computational resources required. Additionally, the approach has the potential to be adapted for different sign languages, allowing for customization in different regions and cultures. In our future work, we also aim to add a live translation feature, and increase our dataset to cover all signs.

##### Acknowledgment

All praise and thanks to ALLAH, who provided us with the ability to complete this work. I hope to accept this work from us. We are grateful to *our parents* and *our family* who are always providing help and support throughout the whole years of study. We hope we can give that back to them.

We also offer my sincerest gratitude to my supervisors, *Dr. Doaa Ezzat and T.A Heba Gamal* who have supported me throughout my thesiswith their patience, knowledge and experience.Finally, I would like to thank my friends and all the people who gave me support and encouragement.

# References

1. Dongxu Li, C. R. (2020). *Word-level Deep Sign Language Recognition from Video:.* The Australian National University, Australian Centre for Robotic Vision (ACRV): Dongxu Li.
2. *Deep Learning on Video (Part One): The Early Days…*. (2021, Dec 21). Retrieved from Medium: https://towardsdatascience.com/deep-learning-on-video-part-one-the-early-days-8a3632ed47d4
3. Carlos Ismael Orozco, M. E. (2012). *CNN–LSTM Architecture for Action.* Argentina: Simposio Argentino de Imágenes y Visión.
4. Li Wang, M. I. (2015). Deep Learning Algorithms with Applications to Video Analytics for A Smart City: A Survey. arxiv.
5. Carlos Ismael Orozco, E. X. (2020). *Human Action Recognition in Videos using a Robust CNN LSTM .* Ciencia y Tecnología.
6. *Deep Learning Tutorial to Calculate the Screen Time of Actors in any Video (with Python codes)*. (2018, Sep 11). Retrieved from Analytics Vidhya:https://www.analyticsvidhya.com/blog/2018/09/deep-learning-video-classification-python/
7. Necati Cihan Camgoz, Simon Hadfield, Oscar Koller, Hermann Ney, and Richard Bowden. Neural sign language translation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018.