© 2024 IJNRD | Volume 9, Issue 5 May 2024| ISSN: 2456-4184 | IJNRD.ORG

IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

REGIONAL SIGN LANGUAGE LEARNING AND TRANSLATION USING THE TRANSFER LEARNING MODEL

Ms.V. Swathilakshmi
Assistant professor
Computer Science and Engineering
Sri Manakula Vinayagar Engineering College,

Sudhaarshan R
Computer Science and Engineering
Sri Manakula Vinayagar Engineering

Sanjay R
Computer Science and Engineering
Sri Manakula Vinayagar Engineering College

Gururaman T N
Computer Science and Engineering
Sri Manakula Vinayagar Engineering

Abstract— This study discusses the use of handgestures in sign language for non-verbal communication, particularly among individuals who are deaf or have speech impairments.
Predicting sign language gestures in Tamil and Telugu using CNN algorithms may lack accuracy due to the struggle with capturing sequential and temporal dependencies. To address this, the researchers propose a system that combines GRU and LSTM architectures, designed for modeling sequences. These recurrent neural network models excel at understanding the intricate movements of sign language, picking up on patterns over time with the help of memory mechanisms. Switching to GRU and LSTM architectures boosts accuracy and promotes inclusivity, improving communication for the deaf and speech-impaired.

Keywords: CNN algorithm, GRU and LSTMarchitectures, Hand gesture.

I. INTRODUCTION

Language is a versatile faculty that has developed from touch and movement-based interactions among primates to predominantly aural-oral communication in humans. However, it is important to distinguish between speech andlanguage, as language encompasses various modalities including writing, sign language, and visual communication. Recognizing this distinction is crucial for fostering inclusivity for individuals who communicate through non-oral means.

Sign language is a rich and complex form ofcommunication that utilizes hand movements,facial expressions, and body language. It is not simply a manual representation of spoken language, but a distinct linguistic system in its own right. British Sign Language (BSL) and American Sign Language (ASL) are mutually unintelligible despite English being their spoken counterparts, highlighting the complexity and independence of sign language.

Indian Sign Language (ISL) is used by the Deaf community in India and neighboring countries. It possesses its own grammar, syntax, and cultural nuances, and is not simply a visual mimicry of spoken Hindi or other Indian languages. While ISL shares some similarities with other South Asian Sign Languages, each has significant variations. Learning ISL promotes inclusivity, breaks down communication barriers, and allows hearing individuals to engage with the Deaf community on their own terms.

Despite the diversity of dialects, ISL is incredibly important for empowering the Deaf community, promoting cultural understanding, empathy, and a more connected society.

II. LITERATURE SURVEY:

Motion Based Indian Sign Language Recognition Using Deep Learning

AUTHORS: Atharv Ganpatye, Sunil Mane Although there are many individuals who are hearing mpaired, there is a lack of awarenesssurrounding sign language. Sign language is usedby these individuals to interact with others,utilizing a range of hand gestures, facial expressions, and body movements. Different countries have their own unique variations of sign language. This study introduces a framework forrecognizing Motion-Based Indian Sign Language (ISL) through deep learning techniques. The gestures are recorded using OpenCV, key points are identified using MediaPipe, and a trained LSTM model is employed for sign prediction. The dataset utilized in this research was developed by the team members. An average accuracy of 92% was obtained. The proposed system can be used for real-time Indian Sign Language recognition and can be integrated with video-conferencing applications.

Real-Time Recognition of Indian SignLanguage

AUTHORS: H Muthu Mariappan; V Gomathi A system has been developed to recognize Indian Sign Language (ISL) gestures in real-time. Sign languages typically involve hand movements and facial expressions. The system identifies and tracks Regions of Interest (ROI) using OpenCV's skin segmentation feature. Hand gestures aretrained and predicted using the fuzzy c-meansclustering machine learning algorithm. Gesturerecognition has a variety of applications including controlling robots and home automation, game control, Human-Computer Interaction (HCI), and sign language interpretation. The new system canidentify signs in real-time, making it beneficial forindividuals who are deaf or have speechimpairments to easily communicate with others.

Prototype for Peruvian Sign Language translation based on an artificial neural network approach

AUTHOR: José Enrique Mejía amarra;

Martín Alonso Salazar Cubas; JuniorDavid Sosa Silupú; Carlos Enrique Córdova Chirinos

In this project, we strive to create a prototype that can recognize the Peruvian Sign Language alphabet and translate it into the Spanish alphabet. Our goal is to enhance communication and facilitate interactions for individuals who are not familiar with sign language. We achieved this using an advanced artificial neural network system that was optimized for efficiency. Bend sensors were utilized to capture the movement and characteristics of each sign. After connecting an Arduino board to an Excel spreadsheet, a database was created using a test user's information for training and validation. The neural network was then built using Matlab software to identify signals, such as letters. Real-time data from the test user was collected using a Matlab-Arduino connection, which, along with an online prediction algorithm and neural network, could recognizespecific letters. Finally, an artificial voice algorithm developed in the Matlab software was used to reproduce the recognized lyrics, and another, which allows the corresponding signal to be viewed online. The prototype developed has anaccuracy of 94.60% for the training set and 94.32% for the validation set, in turn, the cost was 120 USD. It is concluded that this prototype is capable of translating Peruvian Sign Language with a high percentage of accuracy, also is viable and economical, so it is recommended to test the effectiveness of this prototype in multiple people with speech disabilities.An artificial voice algorithm created using Matlab software was employed to replicate identified lyrics. Additionally, another algorithm was used toenable viewing of the corresponding signal online.The prototype achieved 94.60% accuracy on the training set and 94.32% on the validation set, with a cost of 120 USD. This study demonstrates that the prototype effectively translates Peruvian Sign Language with high accuracy, making it both feasible and cost-effective. Further testing on individuals with speech disabilities is recommended to assess its effectiveness.

Conversion of Indian Sign Language to Speech by Using Deep Neural Network
AUTHORS: Shaghayegh Shahiri Tabarestani; Ali Aghagolzadeh; Mehdi Ezoji
Computer-aided diagnosis systems are essential for assisting radiologists and speeding up the diagnostic process. In this study, Faster-RCNN was utilized with three different backbone structures to extract features for predicting fracture zones on bone X-rays from the MURA database. Only three subsets out of seven were used, which included X-rays of the humerus, elbow, and forearm. The findings revealed that Faster-RCNN with Inception-ResNet-Version-2 asthe feature extractor achieved the bestperformance. The average precision (AP) of this model on test samples under optimal parameters was 66.82% for IOU=50%.

ign Language Translation
AUTHORS: R.Harini; R.Janani; S.Keerthana; S.Madhubala; S.Venkatasub amanian
Sign language is used by individuals who are hearing impaired as a means of communication. This can pose a challenge for those who do not know sign language, making a system like this beneficial for assisting in communication. The goal of this project is to utilize computer vision to

recognize and translate signs in real-time. The system consists of four main modules: image capture, preprocessing, classification, and prediction. Through image processing, segmentation of the signs can be achieved. Sign gestures are captured and processed using the OpenCV Python library. The captured gesture is resized, converted to a grayscale image, and noise is filtered out to improve prediction accuracy.The captured gesture is resized, converted to grey scale image and the noise is filtered to achieve prediction with high accuracy. The classification and predication are done using convolution neural network.

Portable Hand Gesture Recognition System forGeneralized Sign Language
AUTHORS: Jyotsna Singh; Arpit Jaiswal; Akshay K Sood; Anuj Dhillon; Divyansh Manchanda
Gesture recognition is a cutting-edge technology with many practical uses and has been improving over time to be more precise and responsive. This article introduces a system for recognizing two- handed gestures that is quick and easy to carry around. Instead of using image processing like other designs, this system uses sensors in both

gloves that work together to identify specific handmovements for personalized sign language. The hardware setup includes a set of sensors in each glove. An advanced method has been developed for gathering data and making predictions using machine learning algorithms. An innovative technique has been used to record data, facilitate communication between devices, and synchronize to reduce delays. The anticipated gesture is then shown on a mobile app.

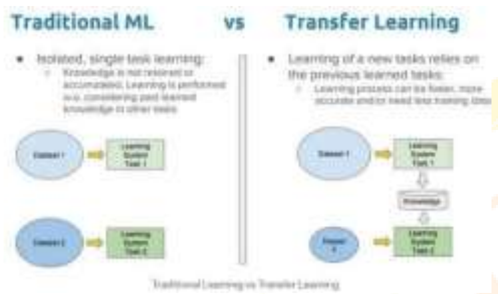
Recognition of hand configuration: A criticalfactor in automatic sign language translation
AUTHORS: Nuno Escudeiro; Paula Escudeiro;Fernando Soares; Orfeas Litos; Marcelo Norberto; Jorge Lopes
Recognizing hand shapes is essential for translating sign language effectively. Our method involves identifying hand shapes in real-time to ensure accurate predictions for automatic sign language translation. We use data gloves equipped with 14 sensors to track finger movements, sampling data at 100Hz. This information is then used to classify the current hand shape. The classification model is built using a set of annotated hand shapes collected in advance. We anticipate that this method will provide consistent and reliable results, ensuring that the classificationmodel performs consistently regardless of the userutilizing it. Our experiments demonstrate a high level of accuracy, indicating that data gloves are effective for capturing hand configuration details. However, the dependability of this method is not ideal, as the classifier's accuracy is reliant on the specific user who trained it, leading to fluctuations in performance in other scenarios.

SNO	Title of the Paper	Authors	year	Publisher
1.	Sign Language Recognition using Machine Learning	Manikandan J; Brahmadesam Viswanathan Krishna; Surya Narayan S; Surendar K	2022	IEEE
2.	Motion Based Indian Sign Language Recognition using Deep Learning.	Atharv Ganpatye, Sunil Mane	2022	IEEE
3.	Sign Language Recognition System.	Chirag Saini, Supreetha S M, Chetana Prakash	2022	IJERT
4.	Sign language identification and recognition: A comparative study.	Ahmed Sultan , Walied Makram , Mohammed Kayed and Abdelmaged Amin Ali	2022	Open Computer Science

5.	Sign Language Recognition System.	Chirag Saini, Supreetha S M, Chetana Prakash	2022	IJERT
6.	Sign language identification and recognition: A comparative study	Ahmed Sultan , Walied Makram , Mohammed Kayed and Abdelmaged Amin Ali	2022	Open Computer Science
7.	Sign language recognition.	Kanchon Kanti Podder Muhammad E. H. Chowdhury Anas M. Tahir, Zaid Bin Mahbub, Amith Khandakar	2022	sensors
8.	Sign language recognition system for communicating to people with disabilities	yulius Obia , Kent Samuel Claudioa , Vetri Marvel Budimana , Said Achmada,, Aditya Kurniawana	2023	Procedia Computer Science
9.	Continuous word level sign language recognition using an expert system based on machine learning	R Sreemathy, MP Turuk , S Chaudhary, K Lavate , A Ushire, S Khurana	2023	Internation Journal of Cognitive Computing in Engineering
10.	Sign language recognition using the fusion of image and hand landmarks through multi-headed convolutional neural network	Refat Khan Pathan, Munmun Biswas, Suraiya Yasmin, Mayeen Uddin Khandaker, Mohammad Salman & Ahmed A. F. Youssef	2023	Scientific Reports
11.	A Comprehensive Review of Sign Language Recognition: Different Types, Modalities, and Datasets	M. MADHIARASAN	2022	arxiv.org
12.	Isolated Word Sign Language Recognition Based on Improved SKResNet-TCN Network	Xuebin Xu,[1,2]Kan Meng,[1,2]**Chen Chen**,[1,2]and Longbin Lu	2023	Journal of sensors

Transfer learning:

Transfer learning is a method in machine learning where knowledge gained from solving one task is applied to improve performance on a different but related task. Instead of training models for specific tasks with separate datasets, transfer learning allows pre-trained models to be adjusted or fine-tuned for a new task with limited labeled data. The idea behind transfer learning is that the information learned from one task can be useful for solving a similar task, even if the new task has different data characteristics or a smaller dataset.



When labeled data for a specific task is hard to come by, transfer learning can be a valuable strategy. It has been successful in a range of fields such as computer vision, natural language processing, and speech recognition. By leveraging existing knowledge, transfer learning can speed up the creation of strong and efficient machine-learning models.

Workflow of the project:

A. Start by finding a pre-trained CNN that is commonly used for tasks such as recognizing images.

B. Decide on the specific task: predicting Tamil and Telugu sign language gestures with accuracy.

C. Gather or create a collection of images or videos showing Tamil and Telugu sign language.

D. Adjust the pre-trained model with the sign language dataset.

E. Use the pre-trained model layers as feature extractors while adjusting it.

F. Use the information learned from the pre-trained model for the new task at hand.

G. Adjust the pre-trained model's features to better fit the intricacies of Tamil and Telugu sign languages.

H. Utilize dropout and gradient descent methods for optimization.

I. Assess how well the fine-tuned model performs with validation data.

J. Incorporate the model into a Flask GUI app to make it more user-friendly.

Applications of the project:

A. In urgent situations, the system can help first responders or emergency services understand sign language gestures quickly and accurately. This is important for ensuring vital information is communicated effectively in high-pressure situations.

B. Social media platforms can use the ISL recognition system to increase accessibility on their platforms. By allowing users to use sign language in videos or live streams, the system can provide real-time translations or subtitles, making content more inclusive for a wider audience.

C. Learning Apps for Sign Languages: This system can be used in apps created for teaching sign languages. Users can get immediate feedback on how accurately they're signing, which helps them learn better with live interaction and guidance.

D. Communication Platforms for Inclusivity: This system can easily be incorporated into video-conferencing apps to translate ISL gestures into either text or speech in real-time. This integration allows deaf individuals to actively join virtual meetings, overcoming communication obstacles and promoting inclusivity.

E. Educational Tools: Implementing the model in educational software can help individuals with hearing impairments learn more effectively. This can improve interactive lessons, quizzes, and educational games that include sign language, making learning more engaging and accessible.

F. Public Services: Connecting with public service systems like customer support centers can improve accessibility for those with hearing impairments. The model can help interpret signs from users, leading to smoother communication and better service delivery.

Existing System:

In existing research, a new dataset and a state-of-the-art Convolutional Neural Network (CNN) were developed for interpreting the American Sign Language alphabet. The CNN has greatly enhanced predictive accuracy, showing improvements even when challenged with different datasets under various conditions. The inclusive dataset, which accounts for real-life variables, enhances the effectiveness of sign language recognition systems. This demonstrates the adaptability of the model in different situations and highlights its potential for enhancing technology accessibility.

Challenges with existing system:

A. A drawback of this study is that the CNN model may not be able to recognize signs outside of the American Sign Language alphabet, limiting its generalizability.

B. By focusing on a specific set of gestures, the model's usefulness in other sign language contexts may be restricted, possibly necessitating more training for broader sign language recognition.

C. Furthermore, the model's effectiveness in real-world situations could differ, as the dataset's controlled settings may not encompass the intricacies of the various environments where sign language is utilized.

III. PROBLEM IDENTIFICATION

Current research shows that current methods for interpreting sign language gestures, especially in languages like Tamil and Telugu, have limitations. Although CNN algorithms have been effective in some cases, they struggle to accurately capture the detailed and sequential aspects of sign language, especially in languages with intricate hand movements. This barrier hinders effective communication and inclusivity for individuals who use sign language as their primary form of communication.

A. Limited Generalizability: The current CNN model created to interpret the American Sign Language alphabet may not be applicable to other sign languages, like Tamil and Telugu. This limitation stems from the model's narrow focus on a specific group of gestures, which could limit its ability to recognize signs in different contexts.

B. Contextual restrictions: When the CNN model focuses solely on certain gestures, it may not be as effective in understanding a wide range of signs in sign language. This means that the model might need more training or adjustments to accurately interpret signs from various sign language systems. As a result, deploying the system in different linguistic environments could become more complicated and resource-intensive.

C. effectiveness in real world: In real-world situations, the CNN model may not perform as well as it does in controlled settings. The training dataset used may not fully represent the complexities of real-life sign language interactions, which could result in differences in model performance when used in practical scenarios.

D. limited ability to capture sequential and temporal dependencies: CNN algorithms work well for classifying images, but they can struggle with understanding the sequential and nuanced nature of sign language gestures. This challenge is especially noticeable in languages like Tamil and Telugu, which have intricate hand movements that rely on the temporal relationships between different gestures for proper interpretation.

E. suboptimal prediction accuracy: Subpar

Prediction Accuracy: CNN algorithms struggle to accurately predict sign language gestures due to their limitations in capturing sequential and temporal dependencies. This can lead to lower accuracy, particularly in languages with complex hand movements. This drawback creates obstacles for effective communication and inclusivity for those who depend on sign language.

To tackle these obstacles, we need to create a more versatile and adaptive method for recognizing sign language. This method should be able to work with various sign language patterns, understand the sequence and timing of gestures, and improve accuracy when predicting words in languages such as Tamil and Telugu. This involves incorporating sophisticated technologies like GRU and LSTM into the prediction model, compiling extensive datasets that cover a wider variety of sign language gestures, and testing the effectiveness of this approach in practical situation

IV. PROBLEM DEFINITION:

The issue with using the CNN algorithm for sign language is that sign gestures are complex and constantly changing. Sign language includes detailed hand movements and body positions, which are essential for accurate interpretation. The difficulty lies in creating a CNN structure that can accurately detect and comprehend these subtle details in visual data.

A. Complexity of Sign Gestures:
The complexity of sign gestures involves intricate hand movements, facial expressions, and body postures, all of which convey important information for interpretation.

B. Need for Effective CNN Architecture:
There is a need for an effective LSTM architecture that can accurately capture and comprehend the subtle features present in sign language gestures.

C. Diversity Across Languages and Individuals:
Different sign languages and individual signers show diverse characteristics, which makes it difficult to create a LSTM that can effectively adapt to these variations.

D. Maintaining Accuracy and Robustness:Creating a lstm that can uphold accuracy and resilience, while also adjusting to the diverse gestures of sign language, presents a complex computational and algorithmic challenge.Real-time Processing Requirement:
We need an algorithm that can quickly process video input to recognize signs instantly for real-time sign language communication.

E. Addressing Variations:
Our goal is to create an LSTM solution that can accurately recognize signs in sign language, even with all the complexities and variations present in real-world scenarios.

F. Enhanced Accessibility for hearing impaired:
Enhanced Accessibility for Hearing Impaired: Our ultimate goal is to help improve accessibility for people with hearing impairments by creating an LSTM solution specifically designed to tackle the obstacles of sign language interpretation.

Objective:

Sign language prediction is a groundbreaking technological advancement that aims to revolutionize communication between deaf and hearing individuals. It offers three key promises:

- Democratizing access
- Harnessing machine learning magic
- Embracing challenges for continuous progress.

The objective of sign language prediction is to develop a system that accurately recognizes and interprets sign language gestures, converting them into spoken or written language for individuals unfamiliar with sign language. This technology has the potential to transcend the limitations of spoken language and facilitate seamless communication.

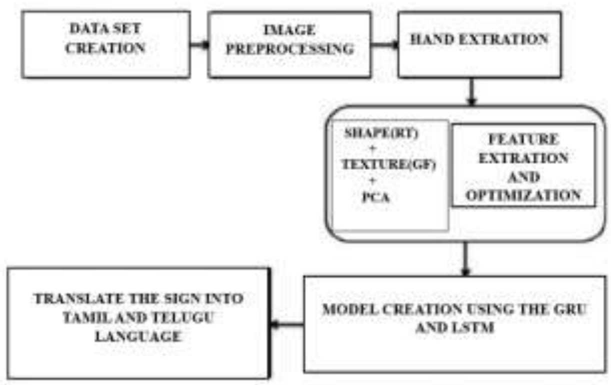
Sign language prediction has numerous transformative applications. In education, AI-powered systems can convert sign language into captions, ensuring equal access for deaf students. In healthcare, it can aid in understanding the needs of deaf patients, leading to improved diagnosisand treatment.

Machine learning, particularly algorithms like Convolutional Neural Networks, plays a crucial role in sign language prediction. These algorithms analyze hand movements, facial expressions, and body language, gradually learning the grammar and vocabulary of sign language. Transfer learning techniques further enhance accuracy by leveraging pre-trained models.

However, challenges persist, including variations in regional sign languages, lighting conditions, and individual signing styles. Interpreting emotion and context conveyed through facial expressions and body language remains an ongoing struggle for perfect accuracy.

Despite these challenges, sign language prediction is rapidly advancing. With continuous research and development, it has the potential to seamlessly integrate into daily life, enabling effortless communication and fostering inclusivity.

V. ANALYSIS AND DESIGN



A. Creating a Database: This is where we gather and arrange the data needed for the translation process.

B. Preparing Images: Various techniques are used in this stage to enhance the quality and characteristics of images, improving feature extraction.

C. Extracting Hand Gestures: Using OpenCV, a library of functions for real-time computer vision, to detect and extract hand gestures from images.

D. Analyzing Features and Optimization: Using Shape (RT), Texture (GF), and Principal Component Analysis (PCA) to extract important features from hand gestures and optimize them for improved performance.

E. Creating Models with GRU and LSTM: In this stage, we build a machine learning model that utilizes Gated Recurrent Unit (GRU) and Long Short-Term Memory (LSTM) networks. These are types of recurrent neural networks that can interpret and convert hand gestures. Translating Signs into Tamil and Telugu: Finally, the model deciphers the signs and translates them into the respective languages.

This framework offers a holistic approach to converting sign language into spoken language using advanced machine learning methods. It demonstrates how technology can improve communication and enhance accessibility to information.

Implementation:

Data Creation:

how the MediaPipe library can be used in Python, specifically its comprehensive model, to detect poses on images or frames from a video stream. The `mediapipe_detection` function processes the input image by converting it from BGR to RGB color format, as needed for the MediaPipe model. It then applies the holistic model to identify different landmarks representing facial features, hand positions, and body poses in the image. After the detection process is done, the function converts the altered image back to BGR format and returns both the modified image and the detection results. This simple implementation highlights the power of MediaPipe for pose detection, offering opportunities for integration into various applications such as gesture recognition, motion tracking, and augmented reality. Additionally, it mentions the creation of a custom dataset, implying the potential for further training or experimentation with pose detection algorithms.

A. Face Connections:
Connecting facial landmarks are drawn using the FACEMESH_CONTOURS parameter, specifying features like contours. The DrawingSpec function sets the color, thickness, and radius for these connections.

B. Pose Connections:

Body pose landmarks, such as joints and limbs, are connected using the POSE_CONNECTIONS parameter. The Drawing Spec function sets the color, thickness, and radius for these connections.

C. Left Hand Connections:
When the AI detects the left hand's position, it draws landmarks and connects them using specified parameters. The DrawingSpec function determines the color, thickness, and circle radius for these connections.

D. Right Hand Connections:
Likewise, when the AI detects the right hand's position, landmarks are drawn with connections between them using the specified parameters. The DrawingSpec function controls the color, thickness, and circle radius for these connections.

Pre-Processing:
To improve the quality and reliability of data before using it in machine learning and computer vision tasks, pre-processing is essential in image processing. This involves various operations to enhance the quality, clarity, and usability of images, such as resizing, normalization, noise reduction, and color conversion. Resizing is important to ensure all images have the same dimensions for compatibility with specific algorithms and models. When we talk about normalization, we are essentially adjusting the pixel values in a way that makes them all within a standard scale. This helps in making learning more effective by minimizing the differences in brightness and contrast. Techniques like noise reduction, through filtering, play a crucial role in getting rid of any unwanted artifacts or distortions that might be present in the image. Sometimes, converting colors may also be necessary to make sure that the colors are represented uniformly across various systems or models. All in all, pre-processing is like laying the foundation for accurate and efficient analysis, which then allows algorithms to draw valuable insights from the data.

Model Creation:
When creating models for sequence modeling tasks, it is important to utilize Gated Recurrent Unit (GRU) and Long Short-Term Memory (LSTM) architectures. These variations of Recurrent Neural Networks (RNNs) are designed to tackle the issue of vanishing gradients and are particularly skilled at recognizing and predicting patterns in natural language processing or time series analysis. By defining the number of layers, units, and connections, the structure of the architecture is established. The GRU and LSTM units effectively retain and update information through their gating mechanisms, allowing for precise predictions and classifications. Ultimately, the model produced can identify features and patterns to make accurate predictions.

Formulas used in the GRU:

The GRU, a component of recurrent neural networks, tackles the issue of handling sequentialdata by using update and reset gates. These gates regulate the flow of information by deciding which past hidden states to keep and how to incorporate new data. The gate values are calculated using sigmoid activation functions, which consider both previous hidden states and current inputs. Moreover, new information is added to the model through a candidate activation using a hyperbolic tangent function. The hidden state is then updated based on the update gate values, blending previous states and candidate activations. This approach allows for selective retention of information, making it easier to capture long-term dependencies in sequential data.

Update gate z:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z)$$

Reset gate r:

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r)$$

Candidate activation $\tilde{h}_t$:

$$\tilde{h}_t = \tanh(W_h \cdot [r_t \cdot h_{t-1}, x_t] + b_h)$$

Hidden state update:

$$h_t = (1 - z_t) \cdot h_{t-1} + z_t \cdot \tilde{h}_t$$

Long Short-Term Memory (LSTM) is a type of neural network that is great at understanding sequences of data. It has memory cells and gates for input, forgetting, and output, which help it remember and update information as needed. The input and forget gates control what information goes in and out of the memory cell, while the output gate manages what information is sent out. These gates use a special function to decide their values based on the input and previous data. With LSTM, the model can remember important details and forget unnecessary ones, making it really good at handling different tasks that involve sequences.

Prediction:

In order to predict sign language using GRU and LSTM algorithms, the first step is to create a dataset that includes sequences of hand gestures and their corresponding labels. These sequences can be in the form of images or video frames. Then, a neural network architecture needs to be

designed, which incorporates GRU or LSTM cells

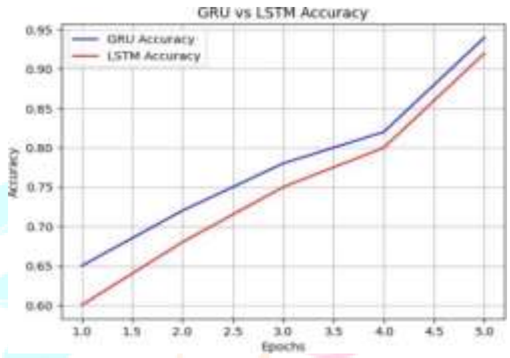
```
# Train the model
history = model.fit([X_train, X_train], y_train,
                    validation_data=([X_val, X_val], y_val),
                    epochs=300,
                    batch_size=128,
                    callbacks=[checkpoint])

# Evaluate the model
test_loss, test_accuracy = model.evaluate([X_val, X_val], y_val)
print(f'Test Loss: {test_loss}, Test Accuracy: {test_accuracy}')
```

for processing the sequential data. Additional layers such as convolutional layers for feature extraction from the images can also be included in this architecture. Once the architecture is prepared, the model is trained on the sign language dataset, allowing it to learn the relationships between hand gesture sequences and their labels through backpropagation and optimization algorithms. Once the model is trained, it can be used to make predictions.

Result and Discussion:

The proposed system represents a notable advancement in the domain of sign language learning, particularly for languages such as Tamil and Telugu. By integrating GRU (Gated Recurrent Unit) and LSTM (Long Short-Term Memory) architectures, specialized variants of recurrent neural networks (RNNs) designed for sequence modeling tasks, the system enhances its ability to capture the intricate temporal dependencies inherent in sign language gestures. Through the utilization of the memory mechanisms embedded within GRU and LSTM units, the system gains a nuanced understanding of the sequential nature of Tamil and Telugu sign language gestures, thereby improving prediction accuracy. Furthermore, the incorporation of these architectures enables the model to discern long-range dependencies and subtle patterns within the sign language data, essential for precise prediction. This transition from the CNN algorithm to GRU and LSTM architectures signifies a significant enhancement in the system's predictive capabilities, offering users a more effective tool for learning sign language in Tamil and Telugu. Beyond mere accuracy improvement, this advancement fosters inclusivity and communication across diverse linguistic communities, facilitating a deeper understanding and interpretation of sign language gestures.

Graph for accuracy:

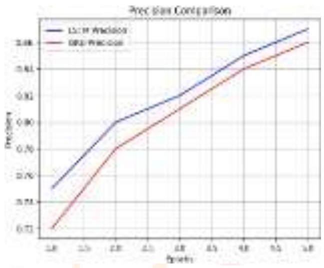


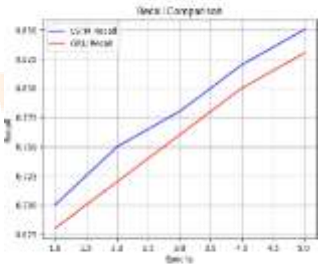
The graph shows how well the GRU and LSTM models perform as they are trained over time. Both models show an increase in accuracy as they are trained, indicating that they are getting better at predicting sign language symbols. At first, the GRU model has a slightly higher accuracy than the LSTM model. However, as training goes on, the LSTM model catches up and eventually reaches a similar or slightly higher accuracy. This means that both GRU and LSTM models are good at predicting sign language, with the LSTM model showing resilience and the ability to reach similar performance levels as the GRU model eventually. In general, the accuracy chart gives us a better understanding of how the models learn and how well they can understand the complexities of sign language sequences while they are being trained. Machine learning accuracy is usually determined through a specific formula.:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \times 100$$

When trying to classify sign language symbols, the "Number of Correct Predictions" measures how many predictions from the model align with the actual labels. The "Total Number of Predictions" is just the overall number of predictions the model makes.

Precision:





In evaluating the performance of LSTM and GRU models, precision, recall, and F1 score are crucial metrics. Precision measures the accuracy of positive predictions, while recall quantifies the ability to correctly identify all relevant instances. The F1 score provides a balance between precision and recall, as it is the harmonic mean of these two metrics.

Precision:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives}+\text{False Positives}}$$

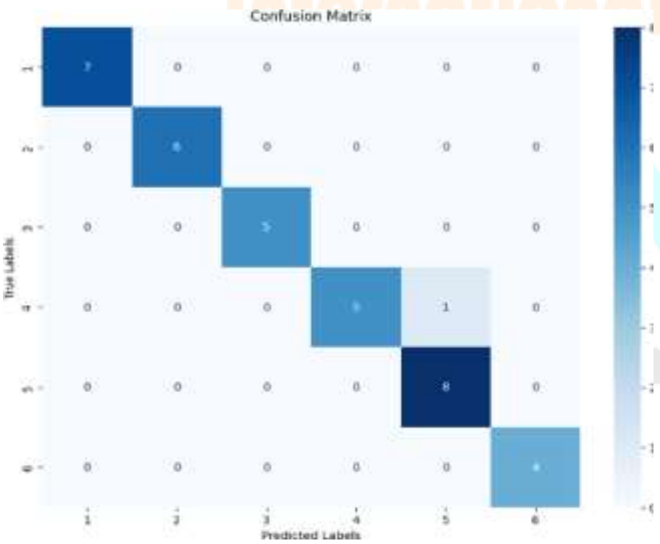
Recall:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives}+\text{False Negatives}}$$

F1 Score:

$$\text{F1 Score} = 2 \times \frac{\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}}$$

In the provided scenario, precision, recall, and F1 score metrics serve as essential measures to assess In this study, we examine the performance of LSTM and GRU models over different training epochs. These metrics offer valuable insights into how well the models can predict and categorize sequential data. By tracking precision, recall, and F1 score values across multiple epochs, the code tracks the changing performance of both LSTM and GRU structures. Using Matplotlib, distinct graphs are created for each metric, enabling a thorough comparison of how LSTM and GRU models perform throughout training.These graphs enable researchers and practitioners to evaluate the effectiveness of LSTM and GRU architectures in handling sequential data tasks and make informed decisions regarding model selection and optimization strategies.



Using the learned model, we can predict values for the validation set (X_{val}). By predicting the output of the model on the validation set, we get labels predicted on individual points. These are produced figures for every class. They are then processed with the use of the `np.argmax` function in order to determine the class with highest probability. This enables us to find a possible class for each data point.After that, a confusion matrix is created by comparing the predicted labels (`y_pred`) with the true labels from the validation set (`y_val`). This matrix gives us a detailed overview of the model's performance by showing the number of correct and incorrect predictions for each class.In simpler terms, the matrix shows how many times the predicted label matches the actual label for each category. A heatmap displays the confusion matrix showing figures of how often each prediction is right. The colors denote the frequency of the matches which enables us to judge accuracy or areas that require modification. This also helps us to know the general performance of the model with respect to different categories especially where errors occur. This visualization is helpful as it enables us to immediately understand how the effectiveness of the model differs from one category to another, observe any patterns in misclassification while at the same time coming up with suggestions on improving the model.

VI. CONCLUSION

In the given system for training sign languages, using advanced GRU and LSTM models is a huge step forward especially for languages like Tamil and Telugu where recurrent neural networks that have always been meant for this purpose are used. This helps in capturing complex time related issues that are associated with sign languages hence improved predictability is achieved by the system. Memory mechanisms inside the GRU and LSTM units help the system to understand well the oral features and contexts of the languages thus make accurate predictions. This shift from traditional CNN schemes to GRU and LSTM networks enables the model to perceive complex relationships as well as subtle signs in sign language data which leads to enhanced learning experience for users Going beyond the ordinary improvements in accuracy, using GRU and LSTM architectures could enhance communication and understanding for various linguistic diversities. It suggests that the intended system brings a more accurate interpretation of sign language movements that would make it possible for the deaf to have more meaningful interaction with other people without regards to their spoken language.This progression not only helps improve the availability of learning sign language, therefore, connecting the gap in communication between people who understand this language and those who have no idea but also leading to bridging the communication gap between those who can sign or not but also significantly moving us closer to having a comprehensive language that allows us to learn more than one dialect through Tamil or Telugu signs.

REFERENCE:

[1] Karpathy and F. Li, "Deep visual- semantic alignments for generating image descriptions," IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 4, pp. 664–676, Apr.2017.

[2] Park, B. Kim, and G. Kim, "Towards personalized image captioning via multimodal memory networks," IEEE Trans. Pattern Anal. Mach. Intell., vol. 41, no. 4, pp. 999–1012, Apr. 2019.

[3] S. Ravi et al., "Foil it! find one mismatch between image and language caption," in Proc. Annu. Meeting Assoc. Comput. Linguist., 2017, pp. 255–265.

[4] Y. Feng and M. Lapata, "Automatic caption generation for news images," IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 4, pp. 797–812, Apr. 2013.

[5] W. Lan, X. Wang, G. Yang, and X.-R. Li, "Improving Chinese image captioning by tag prediction," Chin. J. Comput., vol. 42, no. 1, pp. 136–147, Jan. 2019.

[6] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Boston, MA, USA, 2015, pp. 3156–3164.

[7] Zhang, Y. Dai, Y. Cheng, Z. Jia, and K. Hirota, "Recurrent attention LSTM model for image Chinese caption generation," in Proc. Joint 10th Int. Conf. Soft Comput. Intell. Syst. Int. Symp. Adv. Intell. Syst., Toyama, Japan, 2018, pp. 808–813.

[8] Z. Li, J. Liu, J. Tang, and H. Liu, "Robust structured subspace learning for data representation," IEEE Trans. Pattern Anal. Mach. Intell., vol. 37, no. 10, pp. 2085–2098, Oct. 2015.

[9] Tang et al., "Tri-clustered tensor completion for social-aware image tag refinement," IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 8, pp. 1662–1674, Aug. 2017.

[10] R.Mason and E.Charniak, "Nonparametric method for data-driven image captioning," in Proc. Annu. Meeting Assoc. Comput. Linguist., 2014, pp. 592–598.

[11] P. Jing, Y. Su, X. Jin, and C. Zhang, "High-order temporal correlation model learning for time-series prediction," IEEE Trans. Cybern., vol. 49, no. 6, pp. 2385–2397, Jun. 2019.

[12] X. Jin, Y. Su, L. Zou, Y. Wang, P. Jing, and Z. J. Wang, "Sparsitybased image inpainting detection via canonical correlation analysis with low-rank constraints," IEEE Access, vol. 6, pp. 49967–49978, 2018.

[13] Liu, W. Guan, J. Yan, and H. Hu, "Correlation identification in multimodal weibo via back propagation neural network with genetic algorithm," J. Vis. Commun. Image Represent., vol. 60, pp. 312–318, Apr. 2019.

[14] Liu, L. Zhang, Y. Liu, H. Hu, and W. Fang, "Recognizing semantic correlation in image-text weibo via feature space mapping," Comput. Vis. Image Understand., vol. 163, pp. 58–66, Oct. 2017.

[15] Kulkarni et al., "BabyTalk: Understanding and generating simple image descriptions," IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 12, pp. 2891–2903, Dec. 2013.