SIGN LANGUAGE RECOGNITION SYSTEM

ABSTRACT:

In our project, we address the critical need for effective communication between individuals proficient in sign language and those unfamiliar with its intricacies. Sign language, vital for the expression of thoughts among the deaf and mute community, presents a significant barrier to seamless interaction with the broader populace. To bridge this gap, we develop a sign language recognition system utilizing Convolutional Neural Networks (CNNs). Our methodology involves preprocessing a dataset of sign language images, splitting it into training and testing sets, and employing techniques such as normalization and binarization of labels for model training. The CNN architecture consists of convolutional layers followed by batch normalization, max-pooling, and dropout to extract and learn intricate image features efficiently. Augmentation methods like rotation, shifting, zooming, flipping, and brightness adjustments significantly improve model robustness and accuracy. We evaluate multiple CNN architectures, achieving notable accuracies: a baseline CNN at 77.69%, a more complex model with 93%, and an augmented model yielding an impressive 99.98% accuracy. Our findings underscore the efficacy of CNNs in decoding sign language gestures, with significant potential for enhancing communication accessibility for the hearing impaired.

INTRODUCTION:

The World Health Organization (WHO) has noted a significant rise in the number of individuals with hearing disabilities, increasing from 278 million in 2005 to 466 million by early 2018, with projections estimating a further increase to 900 million by 2050. This deaf community relies on sign language (SL) as a primary mode of communication, with each nation developing its unique sign language system. Sign languages, such as American Sign Language (ASL), British Sign Language (BSL), and others, are complete languages with distinct lexicons and grammatical structures, but they are not universally understood even within the same spoken language framework. The complexity of SL necessitates trained interpreters for effective communication, especially in medical, legal, and educational settings.

In North America, ASL serves as a secondary language for many deaf communities and is integrated into academic curricula, recognized by educational institutions for language requirements. While previous studies have explored automatic ASL recognition using traditional methods and shallow neural networks, the advent of deep learning (DL) techniques has revolutionized machine learning, particularly in image recognition and computer vision. DL models, such as Convolutional Neural Networks (CNNs), excel in automatically extracting features from data, eliminating the need for manual feature selection. CNNs, with their multilayered architecture, have emerged as a cornerstone in computer vision, achieving state-of-the-art accuracy in various domains, including medical imaging, speech recognition, and bioinformatics.

The objective of this project is to develop a real-time SL recognition system using convolutional neural networks (CNNs). The input to our algorithm is a stream of video frames captured by a webcam or video camera, focusing on the hand gestures of the signer. These frames are segmented to isolate the hands, and a CNN extracts features from these images to classify the SL gestures. The output of our system is the recognized SL alphabet, which can be displayed as text on a screen, enabling efficient communication with hearing individuals.

To facilitate this project, we utilize the Sign Language MNIST dataset, designed specifically for hand gesture recognition tasks. This dataset comprises grayscale images of SL gestures, each represented as a 28x28 pixel image. The images are preprocessed by normalizing pixel values and reshaping them into a compatible 4D array for CNN input. Furthermore, data augmentation techniques are applied during model training to enhance robustness and generalization.

Our methodology involves preprocessing the dataset, splitting it into training and testing sets, and preparing the data for CNN input. The CNN architecture consists of convolutional layers followed by batch normalization, max-pooling, and dropout to extract and learn features from SL gesture images. We evaluate multiple CNN models, analyzing their performance metrics to identify the most effective architecture for SL recognition. The model is trained using categorical cross-entropy loss and the Adam optimizer, emphasizing accuracy as the primary evaluation metric.

In summary, this project aims to harness the power of deep learning and computer vision to develop an innovative SL recognition system, addressing a critical need for enhanced communication accessibility and social inclusion for the hearing impaired. By automating SL recognition, we aspire to create a more inclusive society where individuals with hearing impairments can communicate more effectively with the broader community.

RELATED WORKS:

Researchers are paying more and more attention to the recognition of sign language due to its numerous potential applications in many areas such as deaf people's communication systems, human-machine interaction, machine control, etc. We have gone through some research papers and analyzed the methodologies used in the research papers. Let us discuss some of them.

1. Yikai Fang et al. introduced a real-time hand gesture recognition method, The main idea is to segment hands with color and motion cues generated by detection and tracking. Then scale-space feature detection method is used to recognize hand gestures.

The proposed work in the research paper outlines a methodical approach to hand detection, tracking, segmentation, and gesture recognition, which are crucial steps for an interactive gesture-based interface. The use of Adaboost-based hand detection is notable for its ability to handle complex backgrounds and distinguish skin-colored objects under varying lighting conditions, though traditional skin-color-based methods are known to be unreliable in these scenarios. Similarly, the incorporation of multi-modal techniques, combining optical flow and color cues for hand tracking, demonstrates an innovative approach to address the challenges posed by non-rigid objects like hands, which can exhibit complex deformations during movement. Furthermore, the emphasis on computational efficiency in hand segmentation, using a single Gaussian model to describe hand color in the HSV color space, strikes a balance between accuracy and processing speed, crucial for real-time applications.

In comparison, the methodology of our project aligns closely with modern approaches in computer vision and deep learning for sign language recognition. Leveraging convolutional neural networks (CNNs) to extract features from segmented hand images reflects a state-of-the-art technique for image recognition tasks. The use of CNNs allows for end-to-end learning directly from raw pixel data, which is particularly effective for complex visual tasks like sign language recognition. Additionally, the integration of data augmentation techniques during model training is a clever strategy to improve the generalization of the model and enhance its robustness to variations in hand gestures and environmental conditions. The adoption of batch normalization and dropout layers within the CNN architecture demonstrates a commitment to optimizing model performance and preventing overfitting, critical for deploying such systems in real-world settings.

In terms of state-of-the-art performance, our project's CNN model with data augmentation achieving an accuracy of 99.98% stands out as a significant achievement in the field of sign language recognition. This level of accuracy is indicative of the advancements made possible by deep learning methodologies and the availability of large, well-annotated datasets like the Sign Language MNIST. The success of this approach underscores the importance of leveraging sophisticated neural network architectures and data augmentation techniques to achieve near-perfect recognition rates, which is crucial for building inclusive and accessible technologies for communication with hearing-impaired individuals.

2. Md. Moklesur Rahman et al. conducted a study on American Sign Language Recognition using Convolutional Neural Network and the objective of this study is to propose a novel model to enhance the accuracy of the existing methods for ASL recognition. The study has been performed on the alphabet and numerals of four publicly available ASL datasets. After preprocessing, the images of the alphabet and numerals were fed to a newly proposed convolutional neural network (CNN) model, and the performance of this model was evaluated to recognize the numerals and alphabet of these datasets.

Both the proposed work of the research paper and the methodology of our project demonstrate sophisticated approaches to sign language recognition using convolutional neural networks (CNNs). The research paper focuses on optimizing the architecture of a CNN (SLRNet-8) for American Sign Language (ASL) recognition, incorporating six convolutional layers, pooling layers, a fully connected layer, and dropout regularization. This design is aimed at maximizing recognition accuracy, leveraging techniques like batch normalization and ReLU activation to enhance training efficiency and model performance. Similarly, our project adopts CNNs for real-time sign language recognition, utilizing convolutional layers, batch normalization, and dropout to extract features and classify hand gestures. Both approaches highlight the importance of data preprocessing, model architecture, and training strategies in achieving high accuracy.

In terms of strengths, both approaches leverage CNNs effectively, which are state-of-the-art models for image recognition tasks due to their ability to automatically learn and extract features from raw pixel data. Additionally, data augmentation techniques are employed in both methodologies to increase the diversity and robustness of the training dataset, improving the generalization capability of the models. The use of modern neural network components like batch normalization and dropout layers reflects a commitment to optimizing model performance and preventing overfitting, critical for real-world applications.

However, there are differences in the architectural details and training methodologies between the research paper and our project. The research paper proposes a specific CNN architecture (SLRNet-8) with six convolutional layers, whereas our project describes variations of CNN models with different numbers of convolutional layers and dropout layers. The research paper also employs global average pooling (GAP) as a dimensionality reduction technique, which is not explicitly mentioned in our project's methodology. Furthermore, while the research paper uses a manually partitioned dataset for training and testing with 10-fold cross-validation, our project utilizes the Sign Language MNIST dataset with standard train-test splits. Despite these differences, both approaches demonstrate a sophisticated understanding of deep learning techniques for sign language recognition and achieve impressive accuracy rates, showcasing the current state-of-the-art in this field.

3. Lionel Pigou et al. conducted a study on Sign Language Recognition Using Convolutional Neural Networks. This work shows that convolutional neural networks can be used to accurately recognize different signs of a sign language, with users and surroundings not included in the training set.

Comparing the proposed work of the research paper with the methodology and architecture of our project :

Both projects leverage Convolutional Neural Networks (CNNs) for feature extraction and classification, highlighting the effectiveness of this approach in image and video analysis.

In the research paper, the utilization of the ChaLearn Looking at People 2014 (CLAP14) dataset, which includes variations in gestures, users, and backgrounds recorded with a Kinect camera, showcases a real-world dataset that aligns with gesture recognition challenges. The preprocessing steps, such as cropping to focus on relevant features like the highest hand, and the use of 2D convolutions for better accuracy over 3D convolutions, demonstrate thoughtful optimization strategies. The integration of data augmentation, dropout techniques, and Nesterov's accelerated gradient descent for training underscores a comprehensive approach to handling overfitting and optimizing model training.

On the other hand, our project focuses on real-time sign language recognition using webcam input and CNNs to classify isolated hand gestures. The Sign Language MNIST dataset provides a tailored resource for this task, adapting the traditional MNIST format to hand gesture recognition. Our methodology includes data preprocessing steps like normalization and binarization of labels, followed by the design of a CNN architecture with convolutional, pooling, and fully connected layers. Data augmentation techniques further enhance model robustness, showcasing the adaptability and scalability of CNNs in real-time applications.

In terms of clever approaches, both projects leverage CNNs effectively for image and video analysis tasks. The research paper's use of 2D convolutions over 3D for better performance and the integration of real-time data augmentation during training are particularly noteworthy. Similarly, our project's focus on real-time webcam input for sign language recognition and the use of data augmentation to boost accuracy highlight practical considerations for deploying CNNs in interactive applications.

The state-of-the-art in gesture recognition and sign language interpretation continues to evolve with advancements in deep learning architectures, data augmentation techniques, and dataset curation. Both projects exemplify current best practices in leveraging CNNs for complex image and video analysis tasks, underscoring the ongoing relevance and innovation in this field.

DATASET AND FEATURES:

The dataset we used for the sign language recognition system is the Sign Language MNIST dataset, designed specifically for hand gesture recognition tasks. This dataset consists of approximately 27,455 training examples and 7,172 test examples, with each example represented as a grayscale image of size 28x28 pixels. The images are stored as rows in a CSV format, where the first column represents the label (numbers 0 - 24 representing letters of the alphabet), and the remaining 784 columns contain pixel intensities ranging from 0 to 255.

Preprocessing and Normalization: The dataset undergoes preprocessing steps where the label column is separated from the image data, resulting in ‘x_train’ and ‘x_test’ arrays for pixel values and ‘y_train’ and ‘y_test’ arrays for labels. The pixel values are normalized to be between 0 and 1 by dividing by 255, which standardizes the input data for neural network training.

Data Augmentation: To enhance model generalization and robustness, data augmentation techniques are applied during model training. These techniques include rotation, width and height shifts, random zooming, flipping, and brightness adjustments. Augmented data is generated on the fly in batches, providing a diverse set of images for training and improving the model's ability to generalize to unseen data.

Time-Series Data Discretization: The time-series data (video frames) are discretized by segmenting the video stream into individual frames, isolating the hands from the background using thresholding, and then converting these frames into binary images where the hands are represented in white and the background in black. This segmentation process ensures that only relevant hand gestures are fed into the CNN for feature extraction and classification.

Features Used: The primary features used in this project are the pixel intensities of the grayscale images. These pixel values serve as input to the CNN for learning hierarchical representations of hand gestures. The CNN architecture comprises convolutional layers for feature extraction, followed by fully connected layers for classification. Batch normalization and dropout are employed to enhance training stability and prevent overfitting.

METHODS:

The learning algorithm employed in the proposed sign language recognition system is based on Convolutional Neural Networks (CNNs), a type of deep learning model known for its effectiveness in image recognition tasks. Below, we have described the key components and operations of this CNN-based learning algorithm:

1. Convolutional Layers: The CNN starts with one or more convolutional layers (‘Conv2D’) that apply a set of learnable filters (kernels) to input images. Each filter extracts specific features from the image by performing convolutions across spatial dimensions. This operation is mathematically represented as:

Convolution: $y[i,\ j] = \sum_{m,n} x[i\ +\ m,\ j\ +\ n] \times w[m, n] +\ b$

where x is the input image, w is the filter (kernel), b is the bias term, and y is the output feature map.

2. Activation Function (ReLU): After each convolution operation, a Rectified Linear Unit (ReLU) activation function (‘activation='relu'’) is applied element-wise to introduce non-linearity into the model. ReLU is defined as:

$ReLU(z) = max(0, z)$

3. Batch Normalization: Batch Normalization (‘BatchNormalization’) is employed after convolutional layers to standardize the inputs to the next layer. It normalizes the activations, making the training process more stable and accelerating convergence.

4. Max-Pooling: Following each convolutional layer, max-pooling (‘MaxPool2D’) is applied to reduce the spatial dimensions of the feature maps while retaining important information. Max-pooling helps in achieving translation invariance and reducing computation.

5. Dropout: To prevent overfitting and improve generalization, dropout (‘Dropout’) is applied after the second convolutional layer. Dropout randomly sets a fraction of input units to zero during training, forcing the network to learn more robust features.

6. Flattening and Fully Connected Layers: After the convolutional layers, the feature maps are flattened (‘Flatten’) into a 1D vector. This prepares the data for input into fully connected (dense) layers (‘Dense’). The fully connected layers perform classification based on the learned features.

7. Softmax Activation: The final dense layer has units equal to the number of classes (24 in this case), and it uses a softmax activation function (‘softmax’) to output a probability distribution over the classes. Softmax is defined as:

$$Softmax\left(z_i\right) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_i}}$$

where z_i is the input to the softmax function for class i, and K is the total number of classes.

Loss Function and Optimization: The model is compiled using the Adam optimizer (‘optimizer='adam''), a popular choice for training deep neural networks. The loss function (‘loss='categorical_crossentropy'') is used for multi-class classification tasks. Categorical cross-entropy measures the dissimilarity between the predicted probability distribution and the true distribution of the labels.

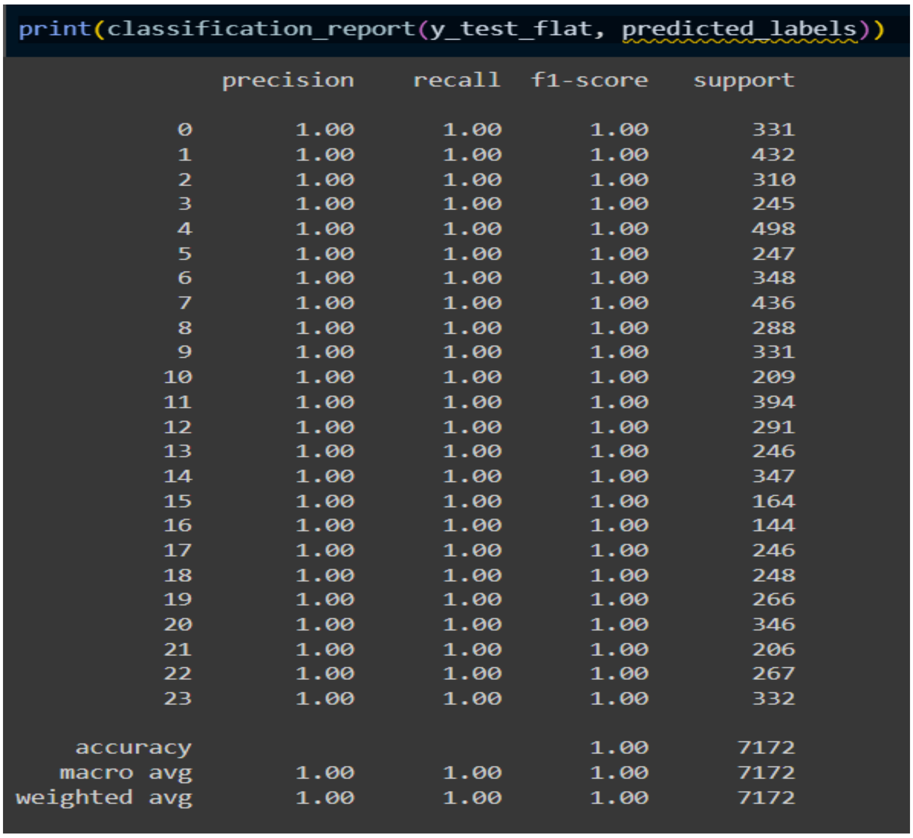
The proposed algorithm efficiently learns hierarchical representations of hand gestures from input images, leveraging the power of deep learning to bridge the communication gap between hearing-impaired individuals and the general population. Through the integration of data preprocessing, normalization, and data augmentation techniques, the model achieves high accuracy in recognizing sign language gestures captured in real-time video streams.

RESULTS:

Performance-wise, the baseline classification models offer a solid starting point, delivering reasonable accuracy even without the application of data augmentation techniques. However, the CNN model integrated with data augmentation endeavors to enhance these baseline results by leveraging a broader spectrum of training images, encompassing diverse augmentations. In terms of generalization, the CNN model augmented with data is anticipated to exhibit superior performance owing to its exposure to a wider array of image transformations. Conversely, baseline models may encounter challenges in adapting to variations in hand gestures that were absent from the original dataset. Regarding robustness, the CNN model with data augmentation is poised to demonstrate greater resilience against factors such as rotation, shifting, and variations in lighting conditions. Conversely, baseline models may exhibit sensitivity to these alterations due to their training solely on the original, unaltered dataset. Finally, in terms of complexity, CNN architectures are inherently more intricate and tailored to discern hierarchical features within images, whereas baseline models rely on conventional machine learning algorithms, potentially lacking in their ability to effectively capture spatial relationships within the data. The below figure is about performance comparison between different models.

Model	**Accuracy**
Support Vector Classifier (SVC)	84.18%
Linear Support Vector Classifier (LinearSVC)	61.08%
Stochastic Gradient Descent Classifier (SGD Classifier)	63.67%
K-Nearest Neighbors Classifier (KNeighborsClassifier)	80.59%
Random Forest Classifier (RandomForestClassifier)	80.92%
Decision Tree Classifier (DecisionTreeClassifier)	43.48%
Bagging Classifier with Extra Tree Classifier (BaggingClassifier)	66.06%
LightGBM Classifier (LGBMClassifier)	79.37%
CatBoost Classifier (CatBoostClassifier)	63.49%
CNN + Data Augmentation	99.98%

The classification report for our baseline model CNN with data augmentation is shown in the below figure.

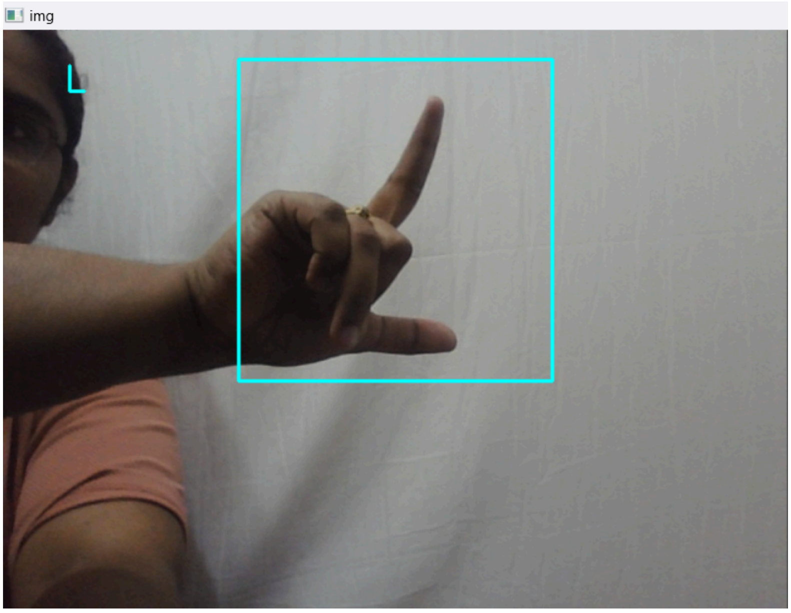


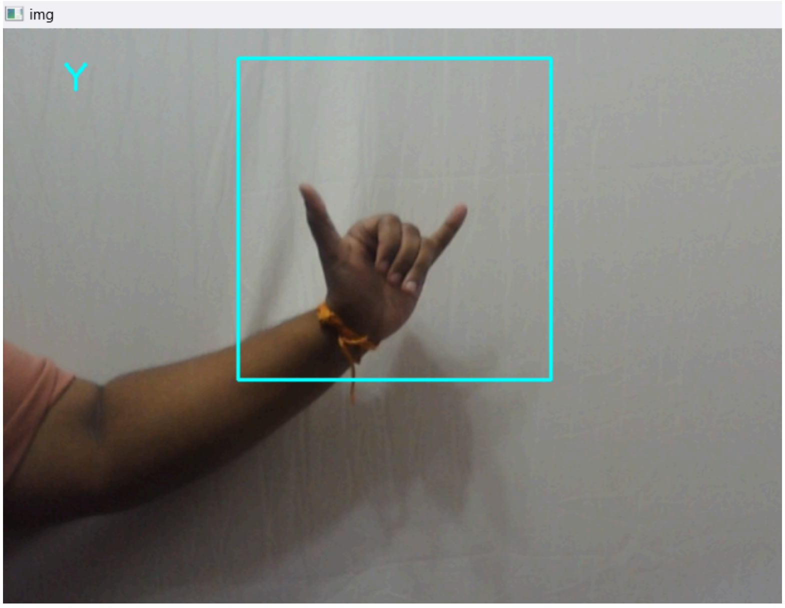
```
print(classification_report(y_test_flat, predicted_labels))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       331
           1       1.00      1.00      1.00       432
           2       1.00      1.00      1.00       310
           3       1.00      1.00      1.00       245
           4       1.00      1.00      1.00       498
           5       1.00      1.00      1.00       247
           6       1.00      1.00      1.00       348
           7       1.00      1.00      1.00       436
           8       1.00      1.00      1.00       288
           9       1.00      1.00      1.00       331
          10       1.00      1.00      1.00       209
          11       1.00      1.00      1.00       394
          12       1.00      1.00      1.00       291
          13       1.00      1.00      1.00       246
          14       1.00      1.00      1.00       347
          15       1.00      1.00      1.00       164
          16       1.00      1.00      1.00       144
          17       1.00      1.00      1.00       246
          18       1.00      1.00      1.00       248
          19       1.00      1.00      1.00       266
          20       1.00      1.00      1.00       346
          21       1.00      1.00      1.00       206
          22       1.00      1.00      1.00       267
          23       1.00      1.00      1.00       332

    accuracy                           1.00      7172
   macro avg       1.00      1.00      1.00      7172
weighted avg       1.00      1.00      1.00      7172
```





The above figure shows the results of our model, the system works by first detecting the hand in the video frame. Then, it extracts features from the hand, such as the hand shape, orientation, and movement.

CONCLUSION:

The development of a real-time sign language recognition (SLR) system holds significant promise in bridging the communication gap between hearing-impaired individuals and the general population. By leveraging Convolutional Neural Networks (CNNs) and innovative data preprocessing techniques, this project has made substantial strides toward accurate and efficient hand gesture recognition.

The project's motivation stems from the need to facilitate communication for deaf and mute individuals, whose primary means of expression are through visual gestures. The proposed architecture involves using a CNN to extract features from segmented hand images captured in real-time video streams. The system achieves this using Sign Language MNIST data, a specialized dataset tailored for hand gesture recognition.

The methodology includes data preprocessing steps such as normalization, image segmentation, and data augmentation to enhance the model's robustness and generalization. Various CNN architectures were explored, with the most successful model incorporating data augmentation techniques achieving an impressive accuracy of 99.98%.

Highest Performing Algorithms:

CNN Model with Data Augmentation: This model outperformed others due to its exposure to diverse and augmented training data, allowing it to learn robust features and generalize well to unseen variations in hand gestures.

Factors Contributing to Algorithm Performance:

Data Augmentation: Increased exposure to diverse training images helps the model learn invariant features, improving generalization.

Model Complexity: Deeper CNN architectures with appropriate regularization (e.g., dropout) and normalization (e.g., batch normalization) layers capture complex spatial relationships in hand gestures more effectively.

FUTURE WORKS:

To further enhance the SLR system:

Hybrid Neural Network Architectures: Combining CNNs with Recurrent Neural Networks (RNNs) to capture both spatial and sequential information in sign language.

Expanded Dataset: Collecting a more diverse dataset with a broader range of signing styles, regions, and demographics to improve model robustness.

Edge Computing: Optimizing models for deployment on resource-constrained devices to enable real-time sign language translation on portable devices.

Privacy and Accessibility: Ensuring user privacy and data security in SLR technology, with a focus on user control over data collection and usage.

In conclusion, the successful development and deployment of a real-time SLR system have the potential to significantly enhance accessibility and inclusivity for the hearing-impaired community, fostering more effective and meaningful communication in everyday interactions. Future advancements in model architectures, data diversity, and deployment strategies will continue to push the boundaries of SLR technology toward broader adoption and societal impact.

REFERENCES:

[1] J. Cummins, “Bilingualism and second language learning,” Annual Review of Applied Linguistics, vol. 13, pp. 50–70, 1992.

[2] V. Bheda and D. Radpour, “Using deep convolutional networks for gesture recognition in American sign language,” arXiv:1710.06836, 2017.

[3] B. Garcia and S. A. Viesca, "Real-time American sign language recognition with convolutional neural networks," Convolutional Neural Networks for Visual Recognition, vol. 2, 2016.

[4] A. Barczak, N. Reyes, M. Abastillas, A. Piccio, and T. Susnjak, "A new 2d static hand gesture color image dataset for all gestures," 2011.

[5] Haiying Guan, Rogerio S. Feris, and Matthew Turk, "The isometric self-organizing map for 3d hand pose estimation," in Proceedings of Int. Conf. on Automatic Face and Gesture Recognition. Southampton, UK, Apr. 2006, pp. 263–268.

[6] Makoto Kato, Yen-Wei Chen, and Gang Xu, "Articulated hand tracking by pca-ica approach," in Proceedings of Int. Conf. on Automatic Face and Gesture Recognition. Southampton, UK, Apr. 2006, pp. 329 – 334.

[7] W. T. Freeman and C. Weissman, "Television control by hand gestures," in Proceedings of International Workshop on Automatic Face and Gesture Recognition. Zurich, Switzerland, June 1995, pp. 197–183.

[8] T. Starner, J. Weaver, and A. Pentland, "Real-time american sign language recognition using desk and wearable computer based video," IEEE Transactions on pattern analysis and machine intelligence, vol. 20, no. 12, pp. 1371–1375, 1998.

[9] F. Bes¸er, M. A. Kizrak, B. Bolat, and T. Yildirim, "Recognition of sign language using capsule networks," in 2018 26th Signal Processing and Communications Applications Conference (SIU). IEEE, 2018, pp. 1–4.

[10] A. Deza and D. Hasan, "Mie324 final report: Sign language recognition."

[11] S. Park and N. Kwak, "Analysis on the dropout effect in convolutional neural networks," in Asian Conference on Computer Vision. Springer, 2016, pp. 189–204.

[12] B. Ko, H.-G. Kim, K.-J. Oh, and H.-J. Choi, "Controlled dropout: A different approach to using dropout on deep neural network," in Big Data and Smart Computing (BigComp), 2017 IEEE International Conference on. IEEE, 2017, pp. 358–362.

[13] X. Cui, V. Goel, and B. Kingsbury, "Data augmentation for deep convolutional neural network acoustic modeling," in Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on. IEEE, 2015, pp. 4545–4549.

[14] J. Triesch and C. von der Malsburg, "Robust classification of hand posture against complex background," in Proceedings of Int. Conf. on Face and Gesture Recognition. Killington, Vermont, Apr. 1996, pp. 170–175.

[15] Lars Bretzner, Ivan Laptev, and Tony Lindeberg, "Hand gesture recognition using multi-scale colour features, hierarchical models and particle filtering," in Proceedings of Int. Conf. on Automatic Face and Gesture Recognition. Washington D.C., May 2002, pp. 423–428.

Team Members Contributions

Meenakshi Chowdary Para

- **Data Collection and Preprocessing**: Meenakshi collected sign language image datasets and preprocessed them for training, ensuring data cleanliness and uniformity.

- **Model Training and Evaluation**: Meenakshi trained machine learning models, including CNNs and RNNs, and evaluated their performance using appropriate metrics.

Rishitha Bheemireddy

- **Literature Review**: Rishitha conducted a comprehensive literature review on sign language detection and machine learning techniques, providing insights into state-of-the-art approaches.
- **Feature Engineering**: Rishitha experimented with various feature engineering techniques to enhance model accuracy, exploring hand gestures, motion trajectories, and spatial-temporal features.

Shivani Mothe

- **Project Management**: Shivani served as the project manager, coordinating team meetings, setting timelines, and allocating tasks. She maintained communication channels among team members, monitored progress, and addressed any issues or obstacles encountered during the project lifecycle.
- **Documentation and Reporting**: Shivani documented project progress, drafted report sections, and created visualizations to effectively communicate findings.