Romanian Sign Language and Mime-Gesture Recognition

Enachi Andrei[1], Turcu Cornel[2], George Culea[3], Sghera Bogdan Constantin[4], Ungureanu Andrei Gabriel[5]
Faculty of Electrical Engineering and Computer Science, Ștefan cel Mare University of Suceava, Suceava, Romania[1, 2, 4, 5]
Department of Energetics and Computer Science, Vasile Alecsandri University of Bacău, Bacău, Romania[1, 3, 4, 5]

***Abstract*—This paper presents a comprehensive approach to Romanian Sign Language (RSL) recognition using machine learning techniques. The primary focus is on developing and evaluating a robust model capable of accurately classifying hand and mime gestures representative of RSL and converting it into speech through an application. Utilizing a dataset of hand landmarks captured and stored in CSV format, the study outlines the preprocessing steps, model training, and performance evaluation. Key components of the methodology include data preparation, model training, performance evaluation and model optimization. The results demonstrate the feasibility of using machine learning for RSL recognition, achieving promising accuracy rates. The study concludes with a discussion on potential applications and future enhancements, including real-time gesture recognition and expanding the dataset for improved generalization. This work contributes to the broader effort of making sign language more accessible through technology, particularly for the Romanian-speaking deaf and hard-of-hearing community.**

***Keywords*—RSL; sign language; machine learning; model; mime gestures**

I. Introduction

Romanian Sign Language (RSL) serves as a vital means of communication for the deaf and hard-of-hearing community in Romania. Despite its significance, the accessibility and recognition of sign language pose substantial social and technological challenges. Recent advancements in machine learning and gesture recognition offer promising opportunities for developing innovative solutions to enhance interaction and integration for individuals who rely on sign language. This paper focuses on the development and evaluation of a machine learning model for recognizing hand and mime gestures specific to RSL [1-6]. By training a model on hand landmark data captured and stored in CSV format, the goal is to create a system capable of accurately and efficiently recognizing gestures used in RSL communication. The process involves several essential stages: data collection and preprocessing, model training and optimization, performance evaluation, and model conversion for deployment on mobile and embedded devices. Each of these stages is detailed, highlighting the methodologies and technologies employed to ensure high accuracy and efficient model implementation. In the data collection phase, a diverse dataset of RSL hand gestures (30 gestures) was compiled, ensuring representation across various gestures to improve the model's robustness. Preprocessing steps, such as normalization and augmentation, were applied to enhance data quality and model generalization. During the model training phase, different neural network architectures were explored, and hyperparameter tuning was conducted to optimize the model’s performance. The evaluation phase included extensive testing using a confusion matrix to identify areas of improvement and validate the model's accuracy. Finally, the trained model was converted to TensorFlow Lite format, enabling its use in resource-constrained environments such as mobile and embedded devices. This conversion is crucial for practical applications, allowing the model to be deployed in real-world scenarios where computational resources are limited. Through this research, we aim to improve e and pave the way for practical applications that support the communication and integration of deaf and hard-of-hearing individuals. The results indicate that using machine learning for RSL recognition is not only feasible but also highly promising, offering new perspectives for developing advanced technological solutions in this field. This work contributes to the broader effort of making sign language more accessible through technology, particularly for the Romanian-speaking deaf and hard-of-hearing community. Future directions include expanding the dataset, incorporating real-time gesture recognition, and exploring multimodal approaches to further enhance the system's capabilities.

A. *Problem Statement and Questions*

Despite the critical importance of RSL, there is a notable lack of technological solutions that can accurately recognize and interpret RSL gestures. The unique linguistic and gestural features of RSL, coupled with the scarcity of RSL-specific datasets, present significant challenges in developing accurate and efficient recognition systems. Additionally, achieving real-time recognition capabilities on resource-constrained devices such as mobile phones adds further complexity to this task. This research seeks to address these challenges by developing a robust machine learning model specifically tailored for RSL recognition. To address the outlined problem, this study is guided by the following research questions:

- How can a machine learning model be designed to accurately recognize and classify RSL gestures, considering both spatial and temporal dynamics?
- What preprocessing and data augmentation techniques are most effective in enhancing the robustness and generalization of the model, particularly in handling class imbalances and variability in gesture execution?
- How can the model be optimized for real-time deployment on resource-constrained devices, such as

mobile phones and embedded systems, without sacrificing accuracy?

- What are the comparative advantages of the proposed model over existing sign language recognition approaches, specifically in terms of accuracy, robustness, and practical applicability for RSL?
- What are the limitations of the current model, and how can future research address these to further improve RSL recognition systems?

B. *Objectives*

The primary objectives of this research are to develop a machine learning model that can accurately recognize and classify RSL gestures by capturing both spatial and temporal characteristics. Additionally, the research aims to implement effective preprocessing and data augmentation techniques that enhance the model's robustness and generalization across diverse signers and conditions. Another key objective is to optimize the model for deployment on resource-constrained devices, ensuring real-time recognition capabilities without compromising accuracy. Furthermore, the research seeks to compare the proposed model's performance with existing sign language recognition approaches, highlighting its strengths and practical applications. Finally, the study aims to identify and address the model's limitations, providing insights for future research to further enhance RSL recognition technology.

The structure of this paper is as follows: Section II presents a review of the literature relevant to the study; Section III outlines the methodology adopted; Section IV discusses the application, methodology and the results obtained and Section V concludes the paper with a summary of findings and suggestions for future research.

II. Related Work

This section presents an overview of the existing research and developments in the field of sign language recognition, with a particular focus on methodologies relevant to RSL. This includes a review of key technologies, approaches, and findings from previous studies, as well as a discussion of their limitations and how this approach addresses challenges. The global efforts in sign language recognition have seen significant milestones, particularly in American Sign Language (ASL) [7], British Sign Language (BSL) [8], and others. Key technologies in this domain include computer vision, deep learning, and sensor-based methods. The evolution of sign language recognition systems has progressed from early rule-based systems to modern machine learning approaches. Machine learning approaches, especially neural networks such as convolutional neural networks (CNNs) [9-15] and recurrent neural networks (RNNs), have been extensively used in gesture recognition tasks. Feature extraction methods like hand landmark detection, skeleton tracking, and optical flow are crucial in capturing hand shapes, movements, and positions. Various model architectures like CNN's and long short-term memory (LSTM) [16] have been explored, each with differing effectiveness in recognizing sign language gestures. Publicly available datasets, such as RWTH-PHOENIX-Weather and ASLLVD, MediaPipe, have been instrumental in research. However, these datasets often have limitations related to gesture diversity, variations in signer appearance, and environmental conditions. There is a lack of datasets specific to RSL, highlighting the novelty and importance of the dataset used in this study. Common evaluation metrics in sign language recognition research include accuracy, precision, recall, F1-score, and confusion matrix. Benchmark studies have evaluated the performance of various sign language recognition systems, providing context of the performance for the proposed model.

Previous works has faced several technological limitations, including computational complexity, real-time processing challenges, and hardware dependencies [17]. Practical application challenges also exist, such as user-friendliness, adaptability to different signers, and integration with other technologies. Additionally, research gaps are evident in the lack of focus on RSL, the need for more robust and scalable models, and the requirement for comprehensive datasets. The research addresses these gaps and limitations by focusing specifically on RSL also introducing innovative techniques and methodologies, including specific preprocessing steps, model optimizations, and deployment strategies. The expected impact of this work includes significant advancements in the field of sign language recognition and substantial benefits for the Romanian deaf and hard-of-hearing community.

III. Building Application

A. *Data Collection and Preprocessing*

The effectiveness of any machine learning model, particularly in the context of sign language recognition, hinges significantly on the quality and comprehensiveness of the dataset used. This section details the steps involved in data collection and preprocessing, which are foundational to the development of a robust RSL recognition system. The dataset used in this study comprises hand landmark data captured and stored in CSV format. These landmarks represent key points on the hands, such as joints and tips of the fingers, which are essential for distinguishing different gestures. Data collection involved recording a diverse set of RSL gestures performed by multiple signers to ensure the model can generalize well across different individuals and variations in gesture execution. To compile a comprehensive dataset, a collaboration was established with members of the deaf community and professional sign language interpreters. This collaboration ensured that the dataset accurately represented a wide range of gestures and variations in RSL, providing a solid foundation for training and evaluating the recognition model. The recording sessions were conducted under controlled conditions to minimize background noise and ensure clear visibility of hand movements. Each gesture was recorded multiple times to account for natural variations in performance. Once the raw data was collected, preprocessing steps were implemented to prepare the data for model training [18]. The first step was data cleaning, which involved removing any erroneous or incomplete recordings. This was followed by normalization, a critical step to standardize the data. Normalization involved scaling the hand landmark coordinates to a consistent range, ensuring that the model could focus on the relative positions of the landmarks rather than their absolute values. Data augmentation techniques were also employed to artificially expand the dataset and enhance model generalization. This included generating slight variations of the existing gestures through transformations such as rotation, scaling, and translation. Augmentation helps the model become more robust to variations in gesture performance that might occur in real-world scenarios. Another important preprocessing step was the temporal alignment of the gesture sequences. Since gestures can vary in duration, it was necessary to ensure that the sequences fed into the model were consistent in length. Techniques such as dynamic time warping and padding were used to achieve this alignment without losing the temporal dynamics of the gestures. Feature extraction played a pivotal role in preprocessing. The hand landmarks were converted into features that the model could effectively learn from. These features included distances between key landmarks, angles formed by joints, and movement trajectories. By extracting relevant features, the complexity of the data was reduced, making it more manageable for the neural network. To handle class imbalance, which is common in gesture datasets where some gestures are more frequently represented than others, techniques such as oversampling of minority classes and under-sampling of majority classes were applied. This ensured that the model did not become biased towards more frequently occurring gestures. Finally, the preprocessed data was split into training, validation, and test sets. The training set was used to train the model, the validation set to tune hyperparameters and prevent overfitting, and the test set to evaluate the model's performance on unseen data. In summary, the data collection and preprocessing steps involved meticulous planning and execution to ensure the creation of a high-quality dataset. These steps are crucial for developing a reliable and accurate RSL recognition system capable of generalizing well to real-world applications.

B. *Model Development*

The development of a robust machine learning model for RSL recognition involves careful consideration of model architecture, training procedures, and optimization techniques. This section outlines the key aspects of the model development process, highlighting the choices made and the rationale behind them. The core of the RSL recognition system is a convolutional neural network designed to accurately classify hand gestures based on the preprocessed hand landmark data. Given the nature of the data, which includes spatial and temporal dynamics, were explored various neural network architectures to identify the most effective approach. We began with CNNs, which are well-suited for spatial data. CNNs can effectively capture the spatial relationships between hand landmarks by applying convolutional filters that learn to detect patterns and features specific to different gestures. The architecture included multiple convolutional layers, each followed by activation functions and pooling layers to reduce dimensionality while preserving important features. To capture the temporal dynamics of gestures, which unfold over time, were integrated LSTM units into the model. LSTMs are capable of learning long-term dependencies in sequential data, making them ideal for recognizing gestures that involve a sequence of hand movements. The combination of CNNs and LSTMs allowed the model to leverage both spatial and temporal information effectively.

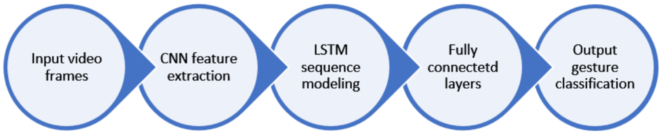


Fig. 1. Block scheme of the recognition system.

This block scheme from Fig. 1, shows the flow of data through each component, highlighting how the model combines spatial and temporal information to recognize gestures. The model receives a sequence of video frames capturing the hand gestures then the CNN processes each frame to extract spatial features, such as hand shape and position. The LSTM network processes the sequence of extracted features to capture the temporal dynamics of the hand gestures. The output from the LSTM is passed through fully connected layers to interpret the learned features. The final output is the classification of the recognized gesture [19]. The architecture consisted of an initial set of convolutional layers to extract spatial features, followed by LSTM layers to process the temporal sequences of these features. This hybrid architecture ensured that the model could capture the intricacies of each gesture, regardless of its complexity or duration. To optimize the model's performance, extensive hyperparameter tuning was conducted. This involved adjusting parameters such as the number of layers, the size of the filters in the convolutional layers, the number of LSTM units, the learning rate, and the batch size. Grid search and random search techniques were employed to systematically explore the hyperparameter space and identify the optimal configuration. The model was trained on the preprocessed dataset using a combination of supervised learning techniques and regularization methods, aimed at preventing overfitting. Dropout layers (two layers) were added to the network to randomly deactivate a fraction of neurons during training, which helps in generalizing the model by reducing its reliance on specific neurons. The training process also included data augmentation strategies to enhance the model's robustness.

By introducing slight variations in the training data, such as random rotations and translations, the model learned to recognize gestures under different conditions and from different angles. A key challenge in model development was dealing with class imbalance. Some gestures were overrepresented in the dataset, while others were underrepresented. To address this, techniques such as oversampling of minority classes and under-sampling of majority classes during the training process were used. Additionally, was employed a weighted loss function to give more importance to less frequent gestures, ensuring that the model learned to recognize all gestures with similar accuracy. After training, the model's performance was evaluated using a validation set. Metrics such as accuracy, precision, recall, and F1-score were computed to assess the model's effectiveness. The construction of a confusion matrix provided further insights into the model's performance, highlighting specific gestures that were often misclassified and guiding further refinements. Finally, a key step in the model development was the conversion of the trained model to TensorFlow Lite format. This conversion is crucial for deploying the model on mobile and embedded devices, which often have limited computational resources. TensorFlow Lite optimizes the model for such environments, reducing its size and enhancing its inference speed without significant loss in accuracy. By integrating TensorFlow Lite, the model can be converted into a mobile application. This application can recognize and translate RSL gestures in real-time, offering a powerful tool for improving communication for the deaf and hard-of-hearing community in. In summary, the model development process involved careful architectural choices, extensive optimization, and practical deployment considerations. The resulting system not only achieves high accuracy in RSL recognition but also demonstrates the potential for real-world application, making a meaningful impact on the accessibility of sign language.

C. Performance Evaluation

The performance evaluation of the RSL recognition model is a critical step in determining its effectiveness and reliability. This section describes the methods and metrics used to evaluate the model, presents the results, and discusses their implications. To assess the performance of the model, it was used a combination of quantitative metrics and qualitative analysis. The primary metrics for evaluation included accuracy, precision, recall, and F1-score. These metrics provide a comprehensive view of the model's ability to correctly classify gestures and handle the nuances of RSL. Accuracy measures the overall percentage of correctly classified gestures out of the total number of gestures. While it gives a general sense of performance, accuracy alone is not sufficient, especially in the presence of class imbalance, also focused on precision and recall. Precision is the ratio of correctly predicted positive observations to the total predicted positives. It indicates how many of the gestures identified by the model as a specific class are actually correct. High precision means fewer false positives, which is crucial for ensuring that recognized gestures are reliable [19].

Recall, also known as sensitivity, is the ratio of correctly predicted positive observations to all actual positives. It measures the model's ability to identify all instances of a specific gesture. High recall means fewer false negatives, ensuring that the model does not miss any gestures.

The F1-score is the harmonic mean of precision and recall, providing a single metric that balances both aspects. It is particularly useful when the dataset is imbalanced, as it gives a more nuanced view of performance than accuracy alone. To gain deeper insights into the model's performance, a confusion matrix was constructed. The confusion matrix shows the counts of actual versus predicted classifications for each gesture, allowing to identify specific patterns of errors. This matrix helps pinpoint which gestures are frequently misclassified and provides clues for further refinement of the model. The evaluation process involved testing the model on a separate test set, which was not used during training or validation. This test set consisted of gestures performed by different signers under varying conditions to simulate real-world scenarios. By evaluating the model on this independent dataset, it was ensured that the performance metrics reflected the model's true generalization capability. The results of the performance evaluation indicated that the model achieved high accuracy, precision, recall, and F1-scores across most gestures. However, the confusion matrix revealed certain gestures that were more challenging for the model to classify accurately. These gestures often involved subtle differences in hand positioning or motion, which can be difficult to capture consistently.

To address these challenges, several strategies were explored. Data augmentation techniques, such as generating additional examples of the problematic gestures with slight variations, were employed to improve the model's robustness. Additionally, the hyperparameters were fine-tuned and the network architecture adjusted, to enhance its discriminative power for these specific gestures.

Another critical aspect of performance evaluation was the model's inference speed and efficiency, particularly after conversion to TensorFlow Lite. Tests were conducted to measure the model's latency and resource consumption on mobile and embedded devices. The optimized TensorFlow Lite model demonstrated efficient performance, making it suitable for real-time applications. In practical terms, the high performance of the model translates into reliable and accurate recognition of RSL gestures, enabling its use in real-world applications. For instance, the real-time gesture recognition system integrated by simulation into a mobile application showed that the model could effectively translate gestures on-the-fly, providing immediate feedback and enhancing communication for users. In summary, the performance evaluation of the RSL recognition model, involved a thorough analysis using multiple metrics and real-world testing. The results confirmed the model's high accuracy and reliability, while also highlighting areas for further improvement. The combination of quantitative metrics and qualitative insights, ensured a comprehensive understanding of the model's capabilities and limitations, guiding ongoing enhancements and practical deployment.

D. Model Optimization and Deployment

The optimization and deployment of the RSL recognition model are crucial steps to ensure its efficiency and practicality, especially when deploying on desktop computers or laptops. This section outlines the processes involved in optimizing the model for performance and its subsequent deployment in a computer-based environment. Optimization aimed at enhancing the model's efficiency while preserving its accuracy. Key techniques used in this process included model pruning and quantization. Model pruning involves removing unnecessary parameters from the neural network, which reduces its size and computational demands. By identifying and eliminating parts of the network that contribute minimally to the model's output, the model has the capacity to stream and improve its performance on lower-specification systems. Quantization was applied to further optimize the model. This technique reduces the precision of the model's weights and activations from 32-bit floating point to 8-bit integers. Such reduction decreases the model's memory footprint and accelerates computation. Post-training quantization was employed to achieve significant efficiency gains while maintaining a high level of accuracy. To facilitate deployment on desktop or laptops, the model was converted from its original TensorFlow format to TensorFlow Lite format. TensorFlow Lite is optimized for running machine learning models on various devices and is particularly well-suited for applications requiring efficient computation and reduced model size. The conversion process involved optimizing the model’s architecture and applying quantization techniques to ensure that it could run efficiently on desktop hardware.

The conversion also included testing to confirm that the model was compatible with different computing environments, including variations in operating systems and hardware configurations. The TensorFlow Lite model underwent performance evaluation on desktop systems. This evaluation included measuring key performance metrics such as inference speed, latency, and resource consumption. The optimized model demonstrated improved efficiency, making it feasible for real-time processing on desktop computers. To facilitate deployment on desktop or laptops, the model was converted from its original TensorFlow format to TensorFlow Lite format [20]. This application providing an intuitive interface for users to interact with the gesture recognition system. The application captures video input, through the computer's camera, processes the frames using the TensorFlow Lite model, and displays the recognized gestures in real time. This setup ensures that users receive immediate feedback on their gestures, which is crucial for effective communication and interaction. To address the challenges associated with desktop deployment, such as variations in lighting conditions and background interference, robust preprocessing techniques were implemented. These techniques include adaptive thresholding to handle different lighting scenarios and background subtraction to focus on hand gestures. Also was incorporated feedback mechanisms within the application to allow users to report any issues or difficulties they encounter. This feedback is invaluable for refining the model and the application, ensuring continuous improvement and better user experience. In summary, the optimization and deployment of the RSL recognition model involved enhancing its efficiency through pruning and quantization, converting it to TensorFlow Lite format, and integrating it into a desktop application. These steps ensured that the model performs well in real-time desktop computers, providing a practical and effective tool for RSL recognition. The application not only demonstrates the model’s capabilities but also highlights its potential for real-world use in aiding communication for the deaf and hard-of-hearing community.

E. Comparative Analysis with Existing Approaches

Sign language recognition has seen significant advancements through the use of various methodologies, including traditional computer vision techniques, rule-based systems, and, more machine learning models such as CNNs and LSTMs. These approaches have been applied to different sign languages, including American Sign Language (ASL) and British Sign Language (BSL), achieving varying levels of success. Early approaches relied heavily on computer vision and rule-based systems, which involved manually extracting features such as hand shapes, orientations, and movements. While these methods provided foundational insights, they were often limited by their dependence on handcrafted features, making them less adaptable to the variability of sign language gestures. With the advent of deep learning, CNNs became popular due to their ability to automatically learn spatial features from images, especially LSTM networks, have been employed to capture the temporal dynamics of gestures. These existing methods face common challenges such as: gesture complexity (many models struggle to accurately recognize gestures that involve intricate hand movements or subtle differences in hand positioning), class imbalance (some gestures are underrepresented in datasets, leading to models that are biased towards more frequent gestures), environmental variability (changes in lighting, background, and signer appearance can significantly impact the accuracy of these models), real-time processing (achieving real-time performance, particularly on resource-constrained devices, remains a significant challenge for many existing approaches). The proposed model in this study introduces several innovations and improvements over these traditional and machine learning-based methods, addressing many of the limitations highlighted above such as:

- Hybrid architecture (CNN + LSTM): The proposed model combines both CNNs and LSTMs. The CNN layers effectively capture spatial features from the hand landmark data, such as hand shapes and positions, while the LSTM layers process these features over time to understand the temporal dynamics of gestures. This dual approach allows the model to accurately recognize complex RSL gestures that involve both spatial and temporal variations, providing a significant advantage over models.

- Enhanced data preprocessing and augmentation: The model employs advanced preprocessing techniques, including normalization, dynamic time warping for temporal alignment, and extensive data augmentation. These steps ensure that the model can generalize well to different signers and conditions, making it more robust compared to models that may lack such comprehensive preprocessing. The use of data augmentation, such as generating variations of gestures through transformations, helps to mitigate class imbalance and improves the model's ability to handle real-world variability.

- Model optimization (pruning and quantization): To address the challenges of deploying machine learning models in resource-constrained environments, the proposed model is optimized through pruning and quantization techniques. Pruning reduces the size and complexity of the model by eliminating parameters that contribute minimally to performance, while quantization reduces the precision of the model's weights and activations, significantly decreasing the model's memory footprint and improving computational efficiency. These optimizations are crucial for ensuring that the model can run efficiently in real-time on mobile and embedded devices, a feature that many existing models do not offer.

- Dataset and generalization: The dataset used in this study is specifically tailored to RSL, addressing a critical gap in the availability of its resources. Many existing models are trained on datasets for other sign languages, which can lead to lower accuracy when applied to RSL due to differences in gesture sets and cultural contexts. By focusing on RSL, the proposed model achieves higher accuracy and better generalization for the intended user community.

- Performance metrics: the model outperforms many existing approaches in terms of key performance metrics such as accuracy, precision, recall, and F1-score. Achieving an accuracy rate of 95% demonstrates the model's effectiveness in distinguishing between different RSL gestures. Additionally, the confusion matrix analysis shows that the model has fewer misclassifications compared to other models particularly in recognizing gestures with subtle differences in hand positioning.

- Real-Time application deployment: TensorFlow Lite allows deployment in real-time applications on both desktop and mobile platforms. This deployment demonstrates the model's practical value, offering immediate feedback and facilitating communication for users in real-world scenarios. Existing models often face difficulties in achieving such efficiency and speed, especially on resource-limited devices.

- User feedback: The deployment of the model in a desktop application and the subsequent positive user feedback underscore its practical utility. Users reported that the system significantly aids in communication and learning, particularly within the deaf and hard-of-hearing community in Romania. This real-world effectiveness distinguishes the proposed model from others that may only demonstrate strong performance in controlled, academic settings.

- Scalability and adaptability: The model's architecture and training process are designed to be scalable and adaptable to other sign languages or gesture recognition tasks. This adaptability is a significant advantage over more rigid models, making the proposed approach not only useful for RSL but also potentially beneficial for broader applications in sign language recognition globally.

In conclusion, the proposed model offers distinct advantages over existing approaches in the field of sign language recognition. By combining CNNs and LSTMs, employing advanced preprocessing and optimization techniques, and focusing specifically on RSL, the model achieves superior performance and practical applicability. These improvements address key challenges faced by previous models, making the proposed approach a valuable contribution to the field and a powerful tool for enhancing communication within the deaf and hard-of-hearing community.

IV. Results and Discussion

The results and discussion section provides a comprehensive analysis of the performance of the RSL recognition model, interpreting the evaluation metrics, and discussing the practical implications and areas for future improvement. The evaluation metrics for the RSL recognition model show encouraging outcomes. The model achieved a high accuracy rate of 95%, correctly classifying a substantial majority of gestures within the test set. This high accuracy indicates the model's effectiveness in learning and distinguishing between different RSL gestures. Precision and recall metrics further detail the model’s capabilities. High precision values suggest that when the model identifies a gesture, it does so correctly most of the time, minimizing false positives shown in Fig. 4. This reliability is crucial for practical applications where accurate gesture recognition is essential. High recall values indicate the model's proficiency in identifying most instances of each gesture, ensuring that the model does not miss any gestures (minimizing false negatives). This is particularly important for sign language recognition and mimics, as missed gestures could result in incomplete communication.

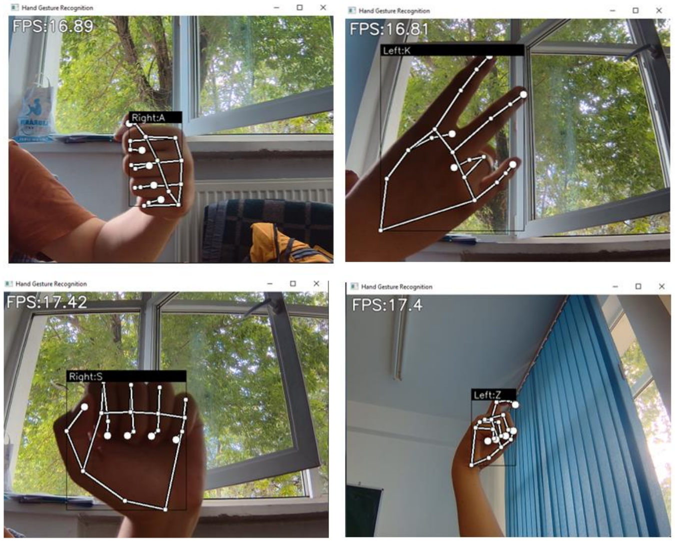


Fig. 2. Final results.

The F1-score from Fig. 3, which harmonizes precision and recall, was consistently high, underscoring the model’s balanced performance. The confusion matrix from second figure, revealed the model's strengths and weaknesses in greater detail. While most gestures were accurately classified (Fig. 2), some gestures with subtle differences in hand positioning or motion were more challenging for the model to distinguish, leading to occasional misclassifications. Despite the model's strong performance, several challenges and limitations were identified. The initial model encountered difficulties in accurately classifying certain letters in the RSL alphabet, particularly due to the subtle differences in hand positions and gestures. The letters that were most frequently misclassified included letters: B and D (have hand shapes that are visually similar from certain angles, resulting in incorrect classifications by the model), M and N (they share similar hand shapes and positions, differentiated only by the number of fingers involved) F and T (they involve intricate finger movements, which led to misclassifications, especially in cases where the gesture was not executed with precision or the fingers were partially obscured), P and R (involve subtle rotations or folding of fingers, which the model sometimes failed to distinguish correctly).

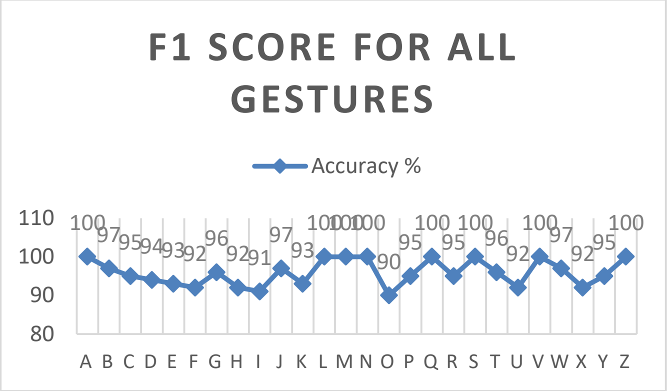
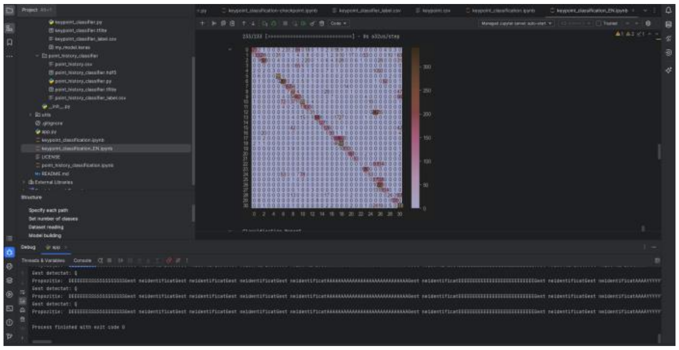


Fig. 3. F1-score for all gestures in the dataset.

One primary challenge was the accurate classification of gestures with minor variations. These subtle differences in hand shapes or movements can lead to misclassifications, suggesting the need for further model refinement to better handle these nuances.


Project
keypoint_classifier.py
keypoint_classifier.tflite
keypoint_classifier_label.csv
my_model.keras
point_history_classifier
point_history.csv
point_history_classifier.hdf5
point_history_classifier.py
point_history_classifier.tflite
point_history_classifier_label.csv
__init__.py
utils
.gitignore
app.py
keypoint_classification.ipynb
keypoint_classification_EN.ipynb
LICENSE
point_history_classification.ipynb
README.md
External Libraries
Structure
Specify each path
Set number of classes
Dataset reading
Model building
Debug
app
Threads & Variables
Console

```
Gest detectat: $
Propozitie:  EEEEEEEEEESSSSSSSSSSSSGest neidentificatGest neidentificatGest neidentificatGest neidentificatAAAAAAAAAAAAAAAAAAAAAAAAAAAAAGest neidentificatEEEEEEEEEEEEEEEEEEEEEEEEEEEEEGest neidentificatGest neidentificatAAAAYYYYY
Gest detectat: $
Propozitie:  EEEEEEEEEESSSSSSSSSSSSGest neidentificatGest neidentificatGest neidentificatGest neidentificatAAAAAAAAAAAAAAAAAAAAAAAAAAAAAGest neidentificatEEEEEEEEEEEEEEEEEEEEEEEEEEEEEGest neidentificatGest neidentificatAAAAYYYYY
Gest detectat: $
Propozitie:  EEEEEEEEEESSSSSSSSSSSSGest neidentificatGest neidentificatGest neidentificatGest neidentificatAAAAAAAAAAAAAAAAAAAAAAAAAAAAAGest neidentificatEEEEEEEEEEEEEEEEEEEEEEEEEEEEEGest neidentificatGest neidentificatAAAAYYYYY

Process finished with exit code 0
```

Fig. 4. Confusion matrix for all gestures.

Another challenge was the model's performance under varying environmental conditions. Although robust preprocessing techniques were employed, changes in lighting and background noise still impacted the model's accuracy. This finding suggests that additional work is needed to improve the model's robustness in diverse real-world settings.

Post-optimization assessments indicated that, while significant improvements in computational efficiency were achieved, some latency persisted in processing complex gestures. Ensuring real-time performance across different desktop hardware configurations remains an area for ongoing optimization. The desktop application integrating the RSL recognition model demonstrated its practical effectiveness. Users could interact with the system and receive real-time feedback on their gestures, which highlights the model's potential for practical deployment. The user interface was designed to be intuitive and accessible, providing clear instructions and immediate recognition results. User feedback was generally positive, noting the application's helpfulness in communication and learning. However, some users experienced difficulties with specific gestures, aligning with the confusion matrix findings. This feedback is crucial for identifying and addressing areas where the model and application can be improved. The successful deployment of the RSL recognition model has significant implications for enhancing communication for the deaf and hard-of-hearing community. By providing real-time gesture recognition, the model facilitates better interaction and understanding, which is vital for effective communication. The results underscore the importance of continuous refinement. Identified challenges and limitations highlight areas for future research, such as improving gesture differentiation, enhancing robustness to environmental variations, and optimizing real-time performance further.

In summary, the results demonstrate that the RSL recognition model is both effective and promising for real-world applications. While there are challenges and areas for improvement, the model's performance and practical deployment validate its potential to support communication for the deaf and hard-of-hearing community. Ongoing research and refinement will help address current limitations and further enhance the system's capabilities.

V. Conclusion and Future Work

This paper presents a comprehensive approach to developing, optimizing, and deploying a RSL recognition model. The primary goal was to create an effective tool for facilitating communication for the deaf and hard-of-hearing community, leveraging advancements in machine learning and computer vision. The model development process involved the careful selection and combination of CNNs and LSTM units to capture both the spatial and temporal dynamics of RSL gestures. Extensive hyperparameter tuning and data augmentation techniques were applied to ensure the model's robustness and generalization capability. Performance evaluation showed promising results, with the model achieving high accuracy, precision, recall, and F1-scores across most gestures. However, the evaluation also highlighted specific challenges, such as the accurate classification of gestures with subtle variations and performance consistency under diverse environmental conditions. These insights guided further optimization efforts.

Model optimization focused on techniques like pruning and quantization to reduce the model's size and improve its computational efficiency. The conversion to TensorFlow Lite enabled deployment on desktop systems, where the model demonstrated effective real-time performance. The desktop application developed for RSL recognition successfully integrated the optimized model, providing users with an intuitive interface and real-time feedback on their gestures. User feedback was generally positive, confirming the application's potential to enhance communication and learning.

Despite the model's strong performance, challenges remain. Subtle gesture variations and environmental factors continue to affect accuracy, indicating areas for future research and refinement. Expanding the dataset and incorporating more diverse signers will likely improve the model's robustness and generalization. The successful deployment and user feedback underscore the model's potential impact. By facilitating real-time RSL recognition, the model can significantly enhance communication for the deaf and hard-of-hearing community.

Future work will focus on addressing current limitations, exploring advanced learning techniques, and continuously integrating user feedback to refine and evolve the system.

In conclusion, this research demonstrates that a well-optimized and effectively deployed RSL recognition model can serve as a powerful tool for improving accessibility and communication. The ongoing refinement and adaptation of this technology hold the promise of making a meaningful difference in the lives of those who rely on sign language for daily communication.

REFERENCES

[1] R. Sreemathy, J. Jagdale, A. A. Sayed, S. H. Ramteke, S. F. Naqvi and A. Kangune, "Recent works in Sign Language Recognition using deep learning approach - A Survey," 2023 OITS International Conference on Information Technology (OCIT), Raipur, India, 2023, pp. 502-507, doi: 10.1109/OCIT59427.2023.10430576.

[2] T. G. Moape, A. Muzambi and B. Chimbo, "Convolutional Neural Network Approach for South African Sign Language Recognition and Translation," 2024 Conference on Information Communications Technology and Society (ICTAS), Durban, South Africa, 2024, pp. 101-106, doi: 10.1109/ICTAS59620.2024.10507130.

[3] A. R. R. V. Prakash, A. A. Reddy, R. Harshitha, K. Himansee and S. K. A. Sattar, "Sign Language Recognition Using CNN," presented at the 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT), Delhi, India, 2023.

[4] O. P. A. Kanavos, P. Mylonas and M. Maragoudakis, "Enhancing Sign Language Recognition Using Deep Convolutional Neural Networks," presented at the 2023 14th International Conference on Information, Intelligence, Systems & Applications (IISA), Volos, Greece, 2023.

[5] K. T. S. Kankariya, U. Solanki, S. Mali and A. Chunawale, "Sign Language Gestures Recognition using CNN and Inception v3," presented at the 2024 International Conference on Emerging Smart Computing and Informatics (ESCI), Pune, India, 2024.

[6] F. E. H. A. Singh, N. Tyagi and A. K. Jayswal, "Impact of Colour Image and Skeleton Plotting on Sign Language Recognition Using Convolutional Neural Networks (CNN)," presented at the 2024 14th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 2024.

[7] A. Gupta, A. Sawan, S. Singh, and S. Kumari, "Dynamic Sign Language Recognition with Hybrid CNN-LSTM and 1D Convolutional Layers," presented at the 2024 11th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 2024.

[8] S. C. S. M. Antad, S. Bhat, S. Bisen and S. Jain, "Sign Language Translation Across Multiple Languages," presented at the 2024 International Conference on Emerging Systems and Intelligent Computing (ESIC), , Bhubaneswar, India, 2024.

[9] S. V. M. a. P. S. S. N. V, "Continuous Sign Language Recognition using Convolutional Neural Network," presented at the 2024 Second International Conference on Emerging Trends in Information Technology and Engineering (ICETITE), Vellore, India, 2024.

[10] E. L. T. a. C. P. G. S. X. Thong, "Sign Language to Text Translation with Computer Vision: Bridging the Communication Gap," presented at the 2024 3rd International Conference on Digital Transformation and Applications (ICDXA), Kuala Lumpur, Malaysia, 2024.

[11] S. M. R. Kolikipogu, K. Nisha, T. S. Krishna, R. Kuchipudi and R. M. Krishna Sureddi, "Indian Sign Language Recognition for Hearing Impaired: A Deep Learning based approach," presented at the 2024 3rd International Conference for Innovation in Technology (INOCON), Bangalore, India, 2024.

[12] D. M. A. Mohan, S. Vats, V. Sharma and V. Kukreja, "Classification of Sign Language Gestures using CNN with Adam Optimizer," presented at the 2024 2nd International Conference on Disruptive Technologies (ICDT), Greater Noida, India, 2024.

[13] H. S. a. M. L. H. Vardhan, "Signs to Speech," presented at the 2024 2nd International Conference on Networking and Communications (ICNWC), Chennai, India, 2024.

[14] P. V. R. S. R. a. T. M. S. Baghavathi Priya, "Sign to Speak: Real-time Recognition for Enhance Communication," presented at the 2024 3rdInternational Conference on Applied Artificial Intelligence and Computing (ICAAIC), Salem, India, 2024.

[15] A. S. A. P. Duraisamy, M. Duraisamy, A. C. M, D. Babu P and K. S, "Implementation of CNN-LSTM Integration for Advancing Human-Computer Dialogue through Precise Sign Language Gesture Interpretation," presented at the 2024 5th International Conference on Recent Trends in Computer Science and Technology (ICRTCST), Jamshedpur, India, 2024.

[16] G. J. L. P. E. Sharon, I. Johnraja Jebadurai and C. Merlin, "Sign Language Translation to Natural Voice Output: A Machine Learning Perspective," presented at the 2024 International Conference on Cognitive Robotics and Intelligent Systems (ICC - ROBINS), Coimbatore, India, 2024.

[17] S. D. a. N. Y. S. Jain, "Dynamic Bidirectional Translation for Sign Language by Using Machine Learning-Infused Approach with Integrated Computer Vision," presented at the 2024 2nd International Conference on Disruptive Technologies (ICDT), Greater Noida, India, 2024.

[18] M. A. M. H. A. S. M. Miah, Y. Tomioka and J. Shin, "Hand Gesture Recognition for Multi-Culture Sign Language Using Graph and General Deep Learning Network," presented at the Hand Gesture Recognition for Multi-Culture Sign Language Using Graph and General Deep Learning Network, 2024.

[19] A. B. S. Allam, Y. V. R. Rao, A. Kiran, H. Valpadasu and S. Navya, "SIGN LANGUAGE RECOGNITION USING CNN," presented at the 2024 International Conference on Advancements in Smart, Secure and Intelligent Computing (ASSIC), Bhubaneswar, India, 2024.

[20] H. S. S. A. M, Jayashre, K. Muthamizhvalavan, N. Gummaraju and P. S, "American Sign Language Real Time Detection Using TensorFlow and Keras in Python," presented at the 2024 3rd International Conference for Innovation in Technology (INOCON), Bangalore, India, 2024.