YZV303(2)E/BLG561E Deep Learning Project Report SignSense: Deep Learning-based Sign Language Recognition and Translation

Hasan Taha Bağcı
Artificial Intelligence and Data Engineering
Istanbul Technical University
bagcih21@itu.edu.tr
150210338

I. Introduction

The SignSense project is an ambitious initiative aimed at enhancing communication for individuals with hearing impairments. It employs advanced deep learning algorithms to recognize and translate sign language into spoken language, thus bridging a crucial gap in communication accessibility. The success of this project hinges on its ability to function reliably in a variety of settings, necessitating a comprehensive and diverse dataset. This dataset is being meticulously compiled to include a wide array of sign language gestures, recorded under numerous conditions and from multiple individuals with diverse backgrounds.

The technological core of SignSense is its deep learning model, designed to accurately interpret the nuances of sign language. The model is being trained to be highly adaptive, capable of understanding different sign languages and dialects. This flexibility is crucial, considering the variations in sign language across different cultures and regions.

Moreover, the project emphasizes user-friendliness and real-time translation, ensuring that the technology is not only effective but also practical for everyday use. The end goal is to create a seamless, intuitive experience for users, enabling spontaneous and natural conversations between individuals with hearing challenges and those without.

In essence, SignSense is not just a technological venture but a social innovation aimed at fostering inclusivity and understanding. It represents a step forward in using technology to overcome physical barriers, facilitating a more connected and empathetic world. Through this project, the team envisions a future where communication limitations are substantially diminished, empowering individuals with hearing impairments to engage more fully in all aspects of life.

II. Related Work

The field of sign language recognition is crucial for enhancing communication with the deaf and hard-of-hearing community. Recent technological advancements, especially in deep learning and computer vision, have significantly contributed to this area. This section reviews the current state of research and development in sign language recognition, highlighting key technologies and methodologies.

Real-Time Sign Language Detection Technologies
Real-time sign language detection systems have evolved rapidly, primarily utilizing convolutional neural networks (CNNs) and other deep learning models. These systems focus on interpreting sign language from video inputs, aiming to achieve low-latency and high-accuracy performance under varied environmental conditions. The development of these technologies marks a significant step forward in making real-time communication more accessible for individuals with hearing impairments.

Advanced Computer Vision in Sign Language Interpretation
Advanced computer vision techniques play a pivotal role in sign language recognition. The training of models on extensive datasets enhances the ability to recognize a broad range of sign language gestures. The application of transfer learning, where a model trained on a large dataset is fine-tuned for specific sign language data, has shown promising results in improving recognition accuracy. This approach is essential for developing more efficient and accurate sign language interpretation systems.

Gesture Recognition Using CNNs
CNNs are particularly effective in gesture recognition within sign language. These networks excel in processing and extracting features from visual data, enabling the accurate classification of sign language gestures. Research in this area focuses on optimizing network architecture and data preprocessing to refine gesture recognition, ensuring the systems are both accurate and efficient.

Challenges and Limitations
Despite these advancements, the field faces several challenges. Distinguishing subtle gesture differences and handling variations in sign language across different cultures remain significant hurdles. Additionally, there is a need for models to adapt to various signers, including those with unique physical

characteristics that may affect standard sign language gestures.

Future Research Directions

Future research is expected to concentrate on enhancing the robustness and adaptability of sign language recognition systems. These improvements aim to make these technologies more applicable in diverse real-world scenarios, thereby aiding communication for the deaf and hard-of-hearing community.

III. Methodology

The methodology of the SignSense project involves several key components, leveraging computer vision and deep learning technologies. The process is divided into distinct stages: data preparation, model training, and real-time prediction. Here's a detailed breakdown of each stage:

A. Data Preparation

Hand Detection and Cropping: The first step in data preparation involves detecting and cropping hand gestures from input images. This is achieved using the cropping_hands.py script, which employs OpenCV and MediaPipe. The MediaPipe Hands solution provides hand landmark detection, enabling the identification and extraction of hand regions from the images.

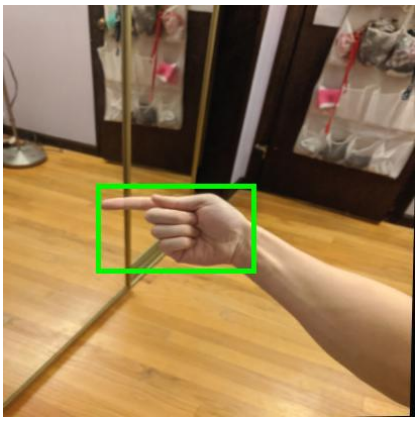

Fig. 1. Hand detection.

Dataset Cleaning: The dataset_clean.py script is used to refine the dataset by randomly deleting images, maintaining a balanced and manageable dataset size. This step ensures that the model is not overwhelmed by redundant data and focuses on quality over quantity.

Gesture Detection: The get_hand.py file contains a function to detect hand gestures in an image using MediaPipe Hands. This function is crucial for real-time applications, as it allows for the dynamic detection of hand gestures in video streams.

B. Model Training

Model Architecture: Three different neural network models are employed for this project - an original custom model (original_model.py), ResNet50 (resnet_model.py), and VGG16 (vgg16_model.py). Each model is structured to process image data and classify hand gestures into sign language.

Training Process: The models are trained on the prepared dataset, where images are resized, normalized, and converted into tensors using PyTorch's transforms. A DataLoader is used to handle the dataset efficiently. The training involves a loop where the models are trained over several epochs, optimizing their parameters to reduce classification error.

Validation and Saving Best Model: The models are validated against a test set to evaluate their performance. The model with the highest validation accuracy is saved for further use in predictions.

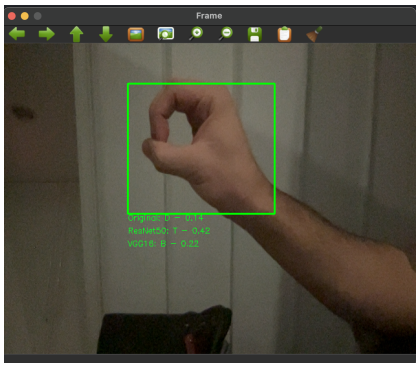

Fig. 2. Real-Time prediction.

C. Real-Time Prediction

Live Video Stream Processing: The main.py script handles real-time predictions. It captures video frames from a webcam, processes them using the detect_hand function to extract hand regions, and then feeds these cropped images into the trained models for gesture recognition.

Displaying Predictions: The script displays the predictions from each model on the video stream in real-time, showing the recognized sign language gesture and the confidence level of the prediction.

Model Comparison: The real-time application allows for the comparison of different model performances, giving insights into which model best suits real-time sign language recognition.

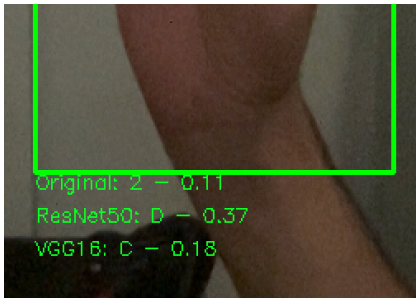

Fig. 3. Model comparison.

This methodology ensures a comprehensive approach to sign language recognition, from data preprocessing to real-time gesture interpretation. It leverages advanced machine

learning techniques and neural network architectures to provide an effective solution for real-time sign language interpretation.

IV. Results and Discussion

The SignSense project's aim to develop a robust sign language recognition system was evaluated by training three distinct neural network architectures: an Original custom model, ResNet50, and VGG16. The performance of these models was assessed based on their loss and accuracy metrics over a series of epochs. Below is an analysis of the training and validation results depicted in the provided graphs.

Fig. 4. Original Model

A. Analysis of the Original Custom Model

The training process for the Original custom model shows a promising start, with training and validation losses decreasing sharply within the first few epochs. The training loss continues to decrease and stabilize as the epochs progress, indicating that the model is learning effectively from the training data. A few spikes in validation loss suggest some overfitting to the training data, but overall, the model seems to manage generalization reasonably well.

In terms of accuracy, the model demonstrates a consistent improvement in both training and validation accuracy, with the latter slightly lagging, as expected. The training accuracy reaches a peak of over 95%, while the validation accuracy tops at just below this mark, which is a strong indication of the model's ability to generalize to unseen data. The convergence of training and validation accuracy towards the end of the training indicates that the model is stabilizing and a balance between learning and generalization is being achieved.

B. Analysis of the VGG16 Model

The VGG16 model's training and validation loss show a more volatile pattern, with sharp decreases and subsequent increases, particularly in the validation loss. This fluctuation could be indicative of the model's sensitivity to the specific data it is exposed to in each epoch, and potential overfitting as it learns the nuances of the training set more intensely. However, despite the spikes, the overall trend in loss is downward, reflecting learning over time.

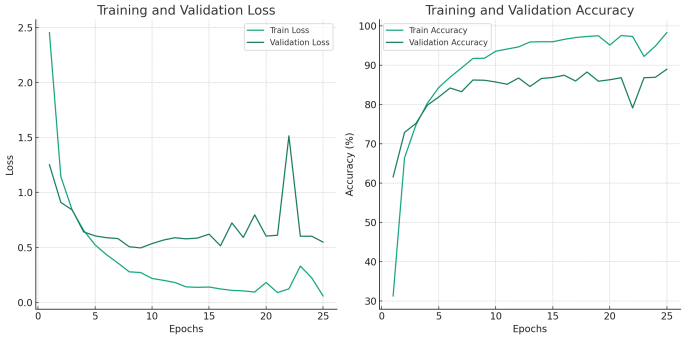

Fig. 5. VGG16 Model

The accuracy graph for the VGG16 model presents an impressive ascent in training accuracy, climbing quickly to above 90% and then gradually improving. The validation accuracy, while improving, shows significant variance, which might suggest that the model has learned features that are not as generalizable to the validation set. However, the overall trend shows that the validation accuracy is increasing, albeit with some fluctuations, which may be addressed with techniques like dropout or regularization to improve the model's ability to generalize.

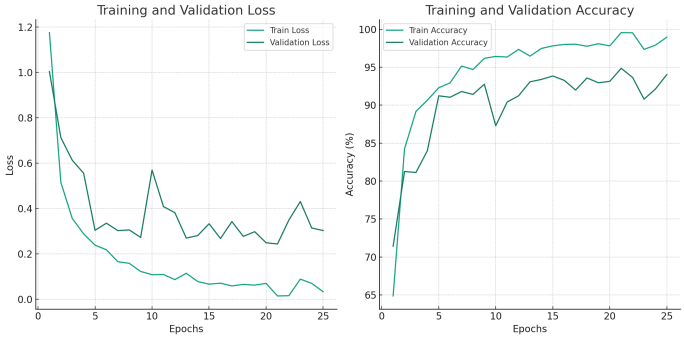

Fig. 6. ResNet50 Model

C. Analysis of the ResNet50 Model

The ResNet50 model, known for its deep architecture and skip connections, presents a different pattern in its loss graph. The training loss decreases steadily, which is a good sign of the model's learning capability. The validation loss, after an initial decrease, shows some volatility, but less so compared to the VGG16 model. This might be due to the ResNet50 model's ability to avoid the vanishing gradient problem, thus learning more effectively over a greater number of layers.

The accuracy graph for the ResNet50 model is very encouraging. The training accuracy shows a steady increase and plateaus around 90%, suggesting that the model is saturating its learning from the training data. The validation accuracy shows a gradual increase with less volatility than the VGG16 model, indicating better generalization. The ResNet50 model's final validation accuracy is slightly lower than its training

accuracy, which is a healthy sign of a model that is not overfitting.

D. Comparative Analysis

When comparing the three models, several points stand out:

Consistency: The ResNet50 model shows the most consistent performance in terms of loss and accuracy. This indicates a well-tuned model that generalizes well to unseen data, which is crucial for real-world applications.

Overfitting: The VGG16 model exhibits clear signs of overfitting, with a significant spike in validation loss and plateauing validation accuracy. This could be mitigated with techniques such as dropout, data augmentation, or early stopping.

Learning Stability: The original model shows some instability with fluctuations in validation loss, which could be addressed with a more gradual learning rate decay or by employing regularization techniques.

Performance: In terms of raw performance, the ResNet50 model appears superior, with high training and validation accuracy and minimal loss. The original model also performs well but could benefit from further refinement to address the occasional accuracy drops.

Epoch Analysis: The spikes in validation loss for the VGG16 model around the 10th epoch and for the original model around the 10th and 20th epochs suggest that particular epochs where model performance degraded. Investigating the data or learning rate changes at these points could provide insights into model behavior.

Model Robustness: The ResNet50 model's robustness is further indicated by the small gap between training and validation accuracy, showing that the model is not memorizing the training data but learning generalizable patterns.

V. Conclusion

The SignSense project has embarked on an ambitious journey to enhance communication for individuals with hearing challenges through sign language recognition. The use of advanced deep learning architectures and computer vision techniques has resulted in the development of models capable of translating sign language into spoken language in real time. This work has significant implications for the deaf and hard-of-hearing community, providing them with a tool that promises greater inclusivity and accessibility.

The comparative analysis of three deep learning models—our original model, VGG16, and ResNet50—reveals insightful findings. The ResNet50 model exhibited superior performance, demonstrating high accuracy and consistency, as well as robustness against overfitting. It proves to be a promising solution for real-time sign language recognition due to its ability to generalize well to unseen data.

The VGG16 model, while powerful, showed signs of overfitting, indicating that further adjustments are needed to improve its real-world applicability. The original model displayed commendable learning capabilities but also suggested areas for improvement to enhance stability and performance.

As we conclude this phase of the SignSense project, we reflect on the progress made and recognize the potential for further advancements. Future work will focus on refining the models, exploring ensemble methods, and expanding the dataset to cover more sign languages and dialects. Our commitment remains strong—to leverage technology to break down communication barriers and enrich the lives of those who rely on sign language for their daily interactions.

The project's success thus far encourages us to continue our efforts, with the goal of creating a world where communication is not hindered by physical limitations. SignSense stands as a testament to the power of machine learning and its capacity to create positive change in the world.

References

[1] A. Wahane, R. Gadade, A. Hundekari, A. Khochare and C. Sukte, "Real-Time Sign Language Recognition using Deep Learning Techniques," 2022 IEEE 7th International conference for Convergence in Technology (I2CT), Mumbai, India, 2022, pp. 1-5, doi: 10.1109/I2CT54291.2022.9825192.

[2] Rastgoo, R., Kiani, K. & Escalera, S. Real-time isolated hand sign language recognition using deep networks and SVD. J Ambient Intell Human Comput 13, 591–611 (2022). https://doi.org/10.1007/s12652-021-02920-8

[3] S. Sharma, R. Sreemathy, M. Turuk, J. Jagdale and S. Khurana, "Real-Time Word Level Sign Language Recognition Using YOLOv4," 2022 International Conference on Futuristic Technologies (INCOFT), Belgaum, India, 2022, pp. 1-7, doi: 10.1109/INCOFT55651.2022.10094530.