Predicting Sign Language Alphabet Characters Live using Machine Learning

Mitchell Cootauco*, Owen Hunger*, Evan Yu*

Loyola Marymount University, USA
*Authors contributed equally to this effort

December 12, 2023

This report describes the development of a live sign language recognition application that utilizes computer vision techniques for real-time interpretation of specific hand gestures. This project leverages multiple libraries and a machine learning model to achieve accurate and efficient recognition, regardless of image quality or scenario. This process involves self-data collection, extraction of hand landmarks, model training, and a live camera. This report provides an overview of each stage, the analysis, and the results.

1 Introduction

Sign language serves as a critical communication method for people who are hearing impaired and do not have the widespread tech support that voice recognition does. Recognizing the inherent complexity and variability of sign language gestures, we embraced machine learning as a powerful tool to decipher patterns and relationships within the visual and spatial data essential for accurate interpretation.

2 Methodology

2.1 Data Collection

We decided to create the data ourselves to limit the dataset to a smaller size. A custom script was created that would open a camera "cv" window and then prompt the user to press a button, at which point the user holds a specific sign language alphabet hand symbol. When the button is pressed, 100 photos will be taken in about $\sim$5 seconds (depending on computer speed), and the user creates variation in the pictures by moving around their hand in $x/y/z$ dimensions. Photos were stored in folders of 100 photos each, representing each hand symbol. These were numbered 0, 1, 2, 3, ..., corresponding with its hand symbols A, B, C, D, ..., respectively.

2.2 Data Processing

The collected images were processed to extract the hand landmarks from the Mediapipe library. Mediapipe has a multitude of uses, including landmark tracking for body parts or hands, which eliminates a lot of the work for us in processing our data. This hand data includes 21 key points representing various parts of the hand, like knuckles, fingertips, and certain points in the palm. These landmarks give us a super descriptive look at the hand that significantly improves the accuracy of the model, in comparison to if we had to extract the hand data ourselves.

2.2.1 Coordinates

Each hand landmark's x and y coordinates obtained through Mediapipe were then processed and stored for each image. At this step, the visual information was formatted into a numerical format that made it suitable for our machine learning classifier. Each portion of the app was split up into a different file for clarity, so this data was sent to a pickle file for later use.

2.3 Modeling

The modeling phase of our live sign language recognition project was centered around the development and refinement of a machine learning model capable of accurately classifying sign language gestures. This phase was crucial in translating the processed hand landmark data into meaningful predictions.

2.3.1 Selection of Model

For our project, we chose the RandomForestClassifier from scikit-learn, a popular and versatile machine learning algorithm. The choice of RandomForest was driven by its inherent strengths in handling complex, high-dimensional data and its robustness against overfitting, which is essential when working with intricate patterns like hand gestures. RandomForest operates by constructing a multitude of decision trees during training and outputting the class that is the mode of the classes of the individual trees, thereby providing both accuracy and reliability in its predictions.

2.3.2 Model Training

In the `train_classifier.py` script, the `RandomForestClassifier` was trained using hand landmark data extracted in the previous phases. The training involved feeding the model with normalized coordinates of hand landmarks as features, and the corresponding gesture labels as targets. We opted for the default parameters of the classifier, relying on its capability to discern patterns in the data without extensive fine-tuning. This approach simplified the model's implementation while still achieving high accuracy.

2.3.3 Model Evaluation

The performance of the model was evaluated using a split of the collected data into training and testing sets, with 20% of the data reserved for testing. This approach allowed us to assess how well the model generalized to new, unseen data. The key metric used for evaluation was accuracy, which provided a straightforward measure of the model's effectiveness in correctly classifying the sign language gestures.

2.3.4 Hyperparameter Tuning

While hyperparameter tuning was considered, as evidenced by commented-out sections in the code about GridSearchCV, it was not utilized in the final implementation. This decision was based on the satisfactory performance of the model with its default settings and the desire to maintain model simplicity.

Through this modeling phase, we established a robust framework for sign language recognition that could effectively interpret hand gestures in real time, paving the way for the application's successful deployment in practical scenarios.

3 Outcome

3.1 Parameters

In the development of our live sign language recognition application, several key parameters and settings were crucial in each stage of the project.

3.1.1 collect_imgs.py Script

In the `collect_imgs.py` script, we defined `number_of_classes` as 5 and `dataset_size` as 100 to collect a balanced dataset of hand gestures. The script uses a webcam (`cv2.VideoCapture(0)`) to capture images of hand gestures, storing them in a structured directory for each class of gestures.

3.1.2 create_dataset.py Script

In `create_dataset.py`, we utilized MediaPipe's `mp_hands.Hands` module with `static_image_mode` set to `True` and a `min_detection_confidence` of 0.3. This configuration was instrumental in accurately detecting hand landmarks in each image. The extracted landmarks, comprising 21 key points per hand, were normalized and stored in a pickle file for model training.

3.1.3 train_classifier.py Script

The `train_classifier.py` script reveals our choice of machine learning model: the `RandomForestClassifier` from `scikit-learn`. Key parameters for the `RandomForestClassifier` were left at their default settings, indicating our reliance on the robustness of the default configuration. The dataset was split into training and testing sets with a test size of 20%, and stratified sampling was employed to ensure a balanced representation of each class in both sets.

3.1.4 inference_classifier.py Script

Finally, in `inference_classifier.py`, the trained model is deployed in a real-time application using a webcam. The script processes live video frames to detect hand landmarks, using the same MediaPipe configuration as in the data processing stage, and predicts the sign language gesture using the `RandomForestClassifier`.

Each of these parameters and configurations played a pivotal role in ensuring the efficiency and accuracy of our sign language recognition application, from data collection and processing to model training and real-time inference.

3.2 Metrics

For the evaluation of our sign language recognition model, we primarily focused on the qualitative analysis of the model's performance during real-time gesture recognition. Given the nature of our application, where immediate feedback and practical usability are crucial, the effectiveness of the model was assessed based on its ability to accurately interpret and classify live hand gestures.

While a quantitative metric like accuracy was initially considered, our primary evaluation method involved hands-on testing with various sign language gestures to observe the model's responsiveness and reliability in real-world scenarios. This approach allowed us to directly gauge the model's effectiveness in a practical setting, which is often more telling for applications like ours that are intended for real-time use. The hands-on testing involved multiple users performing a range of sign language gestures in front of the webcam, and the model's predictions were monitored for correctness and consistency.

This qualitative assessment was crucial in ensuring that our model not only performs well statistically but also meets the practical needs of users in real-life situations, particularly for those who rely on sign language as their primary mode of communication. Future iterations of the project could include more formalized quantitative metrics, such as precision and recall, especially as the model's complexity and the diversity of the gestures it can recognize are expanded.

3.3 Evaluation

The evaluation of our sign language recognition model was primarily qualitative, focusing on real-time performance. We conducted hands-on tests where users performed various sign language gestures to assess the model's accuracy and responsiveness in a live environment. This approach allowed us to directly observe the model's effectiveness in practical scenarios.

In these tests, we noted the model's proficiency in recognizing distinct gestures, while also identifying challenges with similar or complex gestures, which occasionally led to misclassifications. This real-world testing provided insights into the model's capabilities and highlighted areas for improvement, especially in distinguishing subtle gesture differences.

This qualitative evaluation aligns with the application's intended use, offering a practical understanding of its performance. Future enhancements could include quantitative metrics for a more comprehensive assessment as the model's gesture recognition range expands.

4 Ethical Considerations

The development and deployment of a live sign language recognition application carry several ethical considerations that must be addressed to ensure responsible and respectful use:

- **Privacy and Data Security:** Handling sensitive user data, especially when recording sign language gestures, necessitates stringent data protection measures to safeguard personal privacy.
- **Inclusivity and Representation:** Ensuring the sign language dataset is diverse and inclusive of different sign languages and dialects to avoid cultural or regional biases.
- **Accessibility:** The technology should be accessible to all users, including those with varying degrees of hearing impairment and different socioeconomic backgrounds.
- **Consent and Awareness:** Users should be fully informed about how their data will be used, and their consent must be obtained, particularly in public or semi-public settings where the application is deployed.
- **Accuracy and Misinterpretation:** Recognizing the limitations of the technology in accurately interpreting all gestures and the potential consequences of misinterpretation in critical communication scenarios.

These considerations are crucial in ensuring that the application is used ethically and responsibly, respecting the rights and needs of all users.

5 Conclusions

5.1 Future Work

In our future endeavors, we aspire to introduce an enhanced feature that visually communicates the accuracy of hand symbols in real-time sign language interpretation. This entails incorporating a percentage score, elegantly displayed alongside the user's hand, within a designated box. This innovative addition aims to provide users with a quick and intuitive understanding of the precision and fidelity of each interpreted sign to its corresponding letter. Additionally, it would be very useful if, in real-time, the computer program was able to record what somebody was saying and save that to a file. This would be a much more viable real-life application of our program with a broader scope, such as interpreting a sign language conversation between two people.

5.2 Applications

The live sign language recognition application developed in this project holds significant potential in various practical and impactful applications. Some of these applications include:

- **Education and Learning:** The app can serve as an educational tool for both deaf and hearing individuals. It can assist in learning sign language by providing real-time feedback on sign accuracy, making it a valuable resource for sign language students and educators.

- **Accessibility in Communication:** By translating sign language into text or speech in real-time, the app can facilitate communication between deaf or hard-of-hearing individuals and those who do not understand sign language. This application is crucial in everyday interactions, social gatherings, and professional settings.

- **Telecommunication and Online Content:** The app can be integrated into video conferencing platforms or social media, enabling real-time sign language translation during live streams, webinars, or online meetings, thus enhancing accessibility for deaf participants.

- **Healthcare and Emergency Services:** In medical settings or emergencies where communication is vital, the app can aid healthcare professionals in understanding and assisting patients who rely on sign language, ensuring they receive accurate and prompt care.

- **Customer Service and Public Services:** The app can be used in customer service kiosks, public service centers, and information desks to help staff interact with deaf or hard-of-hearing customers, promoting inclusivity and equal service access.

By leveraging machine learning and computer vision, this live sign language recognition application not only bridges communication gaps but also opens new avenues for inclusion, accessibility, and understanding in a diverse range of fields.

5.3 Lessons Learned

We learned that the bulk of the work running machine learning classifiers on data is in the data processing step. We made the mistake of working ahead when our data was not processed correctly, trying to fix errors and bugs in places where they did not originate from. When making a machine learning project, code should be kept neat and as simple as possible to avoid these errors and oversights.

Conclusion

In conclusion, a live sign language recognition app can be utilized and expanded further to aid in accessibility based on our successful results. Although the scope of this project is tame in comparison to the vast variability of the entire sign language alphabet/dictionary, the precise results indicate that desiring full-fledged sign language computer recognition may be possible. This project also underscores the positive impact technology can have on communication accessibility for the impaired.