Sign-Language Interpreting: A Hands-On Application of Computer Vision

Kashif Bandali, Sahas Veera, Safin Rashid, Paul Clauss

1. Problem Description

Our project aims to develop a computer vision program to recognize American Sign Language (ASL) hand signs from images. One of the main challenges this project faces is the inherent diversity and subtlety in hand sign execution, including variations in hand shapes, positions, and movements, which can affect recognition accuracy. Additionally, we must consider obstacles related to the quality and availability of datasets for training the computer vision and machine learning models, ensuring that these models can generalize well across different signers and environments. By addressing these challenges, the project aspires to lay the groundwork for more advanced systems that could eventually operate in real-time and understand gestures, allowing individuals who use ASL to communicate more effectively with people who do not know sign language.

2. Preliminary Literature Survey

A significant portion of this project relies on the ability to detect a hand amidst a variety of backgrounds. Hand identification is done by pinpointing key features of one's hand – chiefly the palm and the five fingers. After detecting these key features, an outline is made of the hand, and points are strategically placed to encapsulate all the individual fingers to capture complex movements.

In aggregate, the recognition process is done in four steps: hand detection, fingers/palm segmentation, finger recognition, and hand gesture recognition. For hand detection, the core part is identifying the skin color and using it to stitch together an isolated hand by alienating other pixels that don't match the identified skin tone. Frameworks such as MediaPipe and OpenPose help utilize keypoint detection to identify the aforementioned hand landmarks [1].

For finger and palm segmentation, the process begins by finding the palm point (center of the palm) via distance transformation. A circle is then drawn around this point based on the calculated maximal radius. Following the identification of this circle, the "palm mask" is generated to turn the general circle into a more detailed shape that identifies the starting point of each finger, as well as the wrist. The final step is finger segmentation, which constitutes removing the palm mast from the image to leave the finders

Following the segmentation process, the fingers are isolated and marked by creating a minimal bounding box. Each finger is identified by first finding the palm line (starting from the wrist line, look for a consecutive layer of horizontal pixels until finger separation is identified by a break). Then, the palm line can be divided into four segments representing each finger. Meanwhile, the thumb is identified by the sharpest degree different from the wrist line.

At this point, hand recognition itself has been completed. The next step is recognizing gestures, or more complex signs, representing words and general concepts. This is typically done by rule classifiers that look to identify based on finger presence. In the provided literature, the simple task of identifying numbers was done. However, for sign language, this process might prove to be more difficult as more than just finger presence is required—the positioning and relative location of each finger has to be considered and compared.

Image segmentation is the process of partitioning an image into multiple segments or regions to simplify its representation, making it more meaningful for analysis [2]. It is crucial in tasks such as object detection, which can be applied to hand and gesture segmentation. Deep learning, particularly convolutional neural networks (CNNs), is a powerful tool for image segmentation due to its ability to learn patterns and features from data. The article points out techniques such as thresholding, clustering, edge detection, etc. For this application, image segmentation is essential to extract the main subject (hands) from the image environment. But, we must also be able to extract and outline fingers, knuckles, palms, etc., and their features from the hands. Because of this, there may be multiple processing methods we may have to use in conjunction with deep learning to teach our model sign language classification.

For a comprehensive overview of the methods of segmentation, including thresholding, region-based, edge, and clustering segmentation; as well as the challenges posed to data scientists who aim to implement such techniques using deep learning models, see An Introduction to Image Segmentation: Deep Learning vs. Traditional [+Examples] [3].

A potential shortcoming we identified is the model not being able to handle motion-based gestures and only being capable of handling individual signs (single frames); others have run into similar issues before, whether it be the 3D modeling or rendering of a video game charac-

ter through motion capture technology, and even ASL-to-audio motion-tracking gloves (similar to speech-to-text) [4]. These methods are effective but require additional, supplemental technology to be able to execute properly. Further, additional hardware is restrictive; that is, something like motion-tracking gloves may limit the desired movement of the speaker. Hence, to remedy this issue, and to create a purely-compute Vision (CV) solution, we searched for a different solution. Among the potential solutions was a subset of CV image recognition strategies called pose estimation, which is best explained as a mathematical graph, that places vertices at strongly-identifiable locations, and connects them with edges. For instance, since we're looking to interpret sign language, the vertices might be the separating regions between each phalanx of one's fingers, with the phalanges themselves being the edges. Mapping one's hand to a graph such as this could make tracking hand movements easier. This isn't foolproof, though; a pose-estimation approach might mean higher confidence with 2-dimensional movements (i.e., left and right, and up and down movements), but lower confidence with 3-dimensional movements (i.e., hand movements to and from the camera). To accommodate this lower, 3-dimensional confidence, hardware distance sensors may be required; however, we have already broken a barrier in translating ASL with CV alone.

To make it abundantly clear, our team does not envision an implementation of pose estimation for this project; rather, we will merely be implementing a "letter sign" recognition model. We simply felt it necessary to point out potential limitations and improvements to our project idea and to expand on why these improvements matter. For instance, with a perfect model, one could implement this software into cameras placed onto glasses lenses, making real-time conversation with the differently-abled seamless.

3. Possible Technical Plan

1. Image processing:
    - → Image Acquisition: Gather high-quality photographs of hand forms, skin tones, lighting conditions, and backgrounds of ASL gestures for a diverse dataset.
    - → Image Preprocessing: Methods like noise removal, scaling, and normalization.
    - → Hand Segmentation: Methods such as thresholding, edge detection, etc. to isolate the gestures and separate the hands from the background, add rotations.
2. Modeling:
    - → Convolutional Neural Network (CNN): To extract features and categorize gestures specifically for sign language identification.
    - → Transfer Learning: Use pre-trained models as feature extractors, then refine them using the sign language dataset to make use of their acquired representations.
    - → Image Classification: The general goal, classifying an image using key points.
3. Training:
    - → Loss Function Selection: Select suitable loss functions.
    - → Optimization: Update model parameters, and use methods such as gridsearch.
    - → Hyperparameter tuning: Experiment with parameters based on training results.
    - → Regularization: Use strategies like dropout or L2 to prevent overfitting.
4. Testing:
    - → Evaluation Metrics: Suitable metrics like accuracy, precision, F-1, and recall.
    - → Cross-Validation: K-fold cross-validation to evaluate the resilience and capacity.
    - → Error analysis: Examine misclassifications and errors to find patterns.
    - → Real-world Testing: Conduct real-world testing on unseen ASL images.

4. References

1. Real-Time Hand Gesture Recognition Using Finger Segmentation
2. What is Image Segmentation?
3. An Introduction to Image Segmentation: Deep Learning vs. Traditional
4. UW undergraduate team wins $10,000 Lemelson-MIT Student Prize for gloves that translate sign language

Additional, uncited references...

1. Comparative Study of Skin Color Detection and Segmentation in HSV and YCbCr Color Space
2. Vision-based hand pose estimation: A review
3. Real Time Finger Counting Application using Computer Vision: Step-by-Step Guide with Code and Video Saving
4. Human Pose Estimation with Deep Learning – Ultimate Overview in 2024
5. Pose Estimation: Concepts, Techniques, and How to Get Started
6. Anatomy of the Hand