APS360 Project Proposal

github.com/yAya-yns/Hand-Gesture-Recognition

Zili Ge 1003213845
Yannis He 1004769707
James Xu 1003056765
Yifan Zhai 1003115559

Oct. 18, 2020
Total Word Count: 1380

1 Introduction

The goal of this project is to categorize human hand gestures using a machine learning approach. The motivation is to offer an alternative method of human-computer interfacing to people with disabilities or the elderly, as well as average consumers, as shown in Figure 1. With this new method, we hope to promote inclusivity and expand usability in consumer electronics.

Traditional methods of hand gesture recognition and categorization involve additional devices such as specially designed gloves or depth camera [1], which are inconvenient, unintuitive, and require extra expenses on the user. Thus, RGB-based computer vision is the best solution. Considering the complexity and reliability requirement of the task, we propose to use deep learning based computer vision (CV) techniques, as it offers more flexibility, accuracy, and possibility of feature expansion compared to traditional CV techniques [7].

Figure 1: Hand gesture to Interact with smart home/AR

2 Background and Related Work

Hand gesture is an effective form of interpersonal communication. As a result, many researchers believe the interactive communication between humans and computers can also be significantly improved by using hand gestures [4]. This form of vision-based communication allows humans to remotely control a wide variety of devices.

Earlier works considered the recognition of significantly differentiable hand gestures, such as American Sign Languages (ASL) and achieved decent results; however, it has been noted that the complex background affects the results negatively for these visual approaches [1]. Some recent works applied deep learning to the hand gesture recognition for the 24 hand gestures obtained from Thomas Moeslund's gesture recognition database [4]. The Convolutional neural networks (CNNs) and stacked denoising autoencoders (SDAEs) achieved recognition accuracy of 91.33% and 92.93%. For real time applications, the complexity of the problem demands powerful embedded processors

which comes with a price tag. In [6], the researchers investigated low-cost implementations of hand gesture recognition, and reduced the computation requirement of the model by using a low-resolution thermal camera for input.

3 Data Processing

For out dataset, raw data will primarily be RGB image data, with no depth information [3]. The motivation is that the number of people with an RGB capable webcam is much larger than those with an RGB-D capable webcam, and as a result this data type is most broadly applicable to the largest group of people. An example is shown in Figure 2.

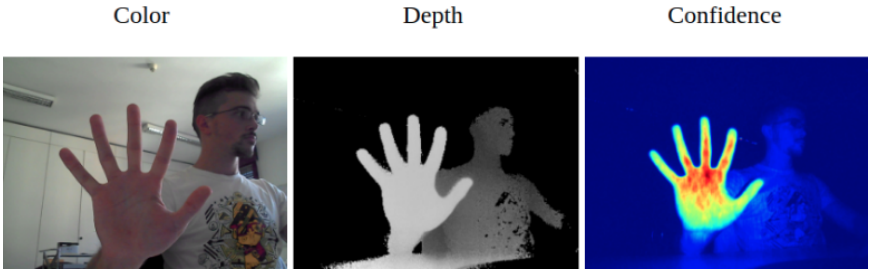

Figure 2: Example from Kinect Leap dataset

For preprocessing, we propose using a hand keypoint detector for generating keypoints, an example is shown in Figure 3.

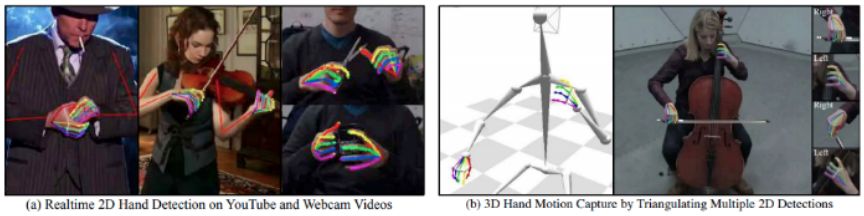

Figure 3: Hand keypoint detection using CV

We further process our keypoint data by including a step that normalizes for hand size by normalizing for distances between nodes in the keypoint detector. We can also consider normalizing for rotation, although we are unsure whether rotational invariance is a characteristic of hand gestures. This will be an ablation study done on our part.

We would also consider working in the RGB space; in this scenario, we will perform hand detection and cropping, then normalize the size of each hand using techniques such as bilinear interpolation. Images that are cut off will be padded with 0s, although different padding can be tried as well (such as taking the median pixel color for regions that are detected as not a hand). To account for differences in skin tone, we can convert images to grayscale and normalize brightness and variance to some value for the hand mask.

4 Architecture

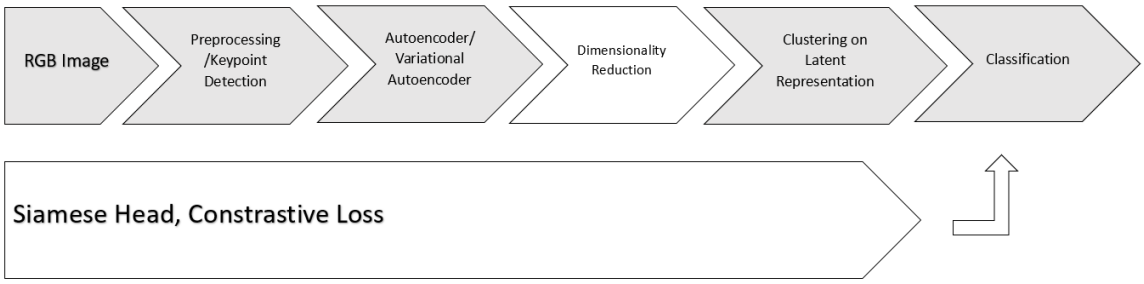

Figure 4: Proposed architecture diagram, with the main flowchart (gray) and other modules we intend to test (white).

Due to the large amount of classes and low amount of training samples per class, clustering techniques will be useful for this dataset.

One possibility is to cluster directly in the space of the hand pose estimates using dimensionality reduction techniques, followed by clustering using techniques.

We also plan to try training an autoencoder or a variational autoencoder (AE/VAE), then performing clustering on the latent representation of this model.

The main reasoning for our decision for both of these sections is the small number of samples per class; there is not enough information here to fine-tune a CNN. As a result, we take advantage of unsupervised techniques in the autoencoder in order to help us cluster results, even if there are very few results per cluster.

If time allows, we will attempt using a siamese network to detect whether a pose falls within a group of poses that we have already constructed. In this way, we can define our own poses.

5 Baseline Model

[3] proposed a SVM classifier and achieved 91% accuracy on the dataset using a fusion of two sensors (viz. Leap motion and Microsoft Kinect). Using only the Microsoft Kinect sensor, 65% accuracy was reported across the 10 gestures. We hope to achieve similar accuracy using only a RGB sensor (no access to depth map).

The dataset provided is collected using a Microsoft Kinect RGB-D camera and as a result has ground truth estimates for hand pose. We will take a small subset of 100 or so of these classes, perform clustering on the ground truth estimates of hand pose and assign clusters to the test set using techniques such as HDBSCAN. To prevent orientation related mishaps due to the simplicity of the model, we will take two keypoints, compute a homogeneous transform and transform each hand so that all the keypoints corresponding to the two that we chose are in the same point in space for all images. Then, we will find a cluster centroid and assign this to the most common class of each cluster. For inference, we will take the euclidean distance of each test point to the centroid of each cluster and assign the class to the nearest centroid.

Our future models will comment on the performance operating on the same subset of classes, as well as the full set of 1400 classes.

6 Ethical Considerations

While the aim of our project is to provide an alternative method of human-computer interfacing (i.e. hand-gesture control of smart phones, etc), there may be unintended consequences regarding these ethical issues:

- Privacy:
  As with most computer vision applications, privacy of the user can be potentially infringed at data-collection. Information such as facial and background images may be extracted at deployment of our model. While beyond the scope of the project, there are ways to mitigate this concern by techniques such as visual abstraction, data hiding, image filtering, etc [5].
- Fairness/Discrimination:
  Since a portion of our data-processing pipeline involves human hand detection, the skin tone of the user may impact the reliability and usability of the model [2]. To guarantee equal usability and fairness of our algorithm, as a future step, we will examine the variation of skin tones in training data to ensure an equal representation across the spectrum.

7 Project Plan

Stage #	Task	Task holder	Deadline
1	Data collection and preprocessing	James Xu	Oct 18
2	Implementation of existing architecture	Zili Ge	Oct 30
3	Neural network training and hyperparameter tuning	Yannis He	Nov 15
4	Model evaluation and testing	Daniel Zhai	Nov 30
5	Integrated HIL system testing	All members	Dec 5
6	Presentation and report preparation	All members	Dec 7

8 Risk Register

- No open source code available for architecture, or code with deprecated dependencies.
  Low risk. We have already found a few open source networks.
- Low number of training samples per class is not enough to perform well.
  Moderate risk. We can use data augmentation to artificially increase the size of the dataset.
- Diversity of dataset (skin color, background, lighting, etc)
  Moderate risk. We can modulate brightness and contrast to a certain degree. We could also combine multiple (similar) datasets.
- High memory requirement (hardware limitations)
  Moderate risk. We have the ability to use cloud computing resources when necessary.
- Team member dropping the course
  Low risk. All members intend to use this course for AI minor.

References

[1] Hardi Desai and Yask Patel. Survey paper on hand gesture recognition. 2016.

[2] Mikael Lauronen. Ethical issues in topical computer vision applications. 06 2017.

[3] G. Marin, F. Dominio, and P. Zanuttigh. Hand gesture recognition with leap motion and kinect devices. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 1565–1569, Oct 2014.

[4] Oyebade Oyedotun and Adnan Khashman. Deep learning in vision-based static hand gesture recognition. *Neural Computing and Applications*, 28, 12 2017.

[5] José Padilla-López, Alexandros Chaaraoui, and Francisco Flórez-Revuelta. Visual privacy protection methods: A survey. *Expert Systems with Applications*, 06 2015.

[6] Maarten Vandersteegen, Wouter Reusen, Kristof Van Beeck, and Toon Goedemé. Low-latency hand gesture recognition with a low resolution thermal imager. pages 440–449, 06 2020.

[7] Joseph Walsh, Niall O' Mahony, Sean Campbell, Anderson Carvalho, Lenka Krpalkova, Gustavo Velasco-Hernandez, Suman Harapanahalli, and Daniel Riordan. Deep learning vs. traditional computer vision. 04 2019.