**Machine Learning for COVID-19: Hand Gesture Recognition**

**CSC532 – Machine Learning**

**Nipat Phattarakijtham**

1. **Abstract**

This paper will be covering the methodology, results, and suggestions of hand gesture recognition. The purpose of this project is to help lessen the infection of COVID-19 through contact transmission. This research will cover the convex hull algorithm and a CNN model with custom dataset.

1. **Introduction**

Due to the COVID-19 pandemic outbreak people are resorting to self-isolation. Recent research has shown that COVID-19 can be transmitted through many different means. Contact transmission is one of the ways. This purpose of this project is to create a way to minimize physical contact with frequently used public objects through the use of image processing and machine learning. By minimizing the contact with frequently used objects such as BTS ticket booths, in theory, the infection rate of COVID-19 would decrease.

1. **Objectives**

* Create a hand gesture detection model
* Use the different gestures to interact with a system (controlling a mouse in this situation).

1. **Methodology**
   * Technologies used:
     + Python
     + Keras (Tensorflow)
     + OpenCV
     + PyautoGUI

* Pre-processing  
  All images used in training and predicting of the two methods have gone through the same steps of preprocessing. The preprocessing consists of gray scaling the images. Then we apply gaussian blur to the gray scaled images. After we blurred the images, it is then passed into Otsu’s binarization method. See figures 1-6 for the resulting images.

|  |  |
| --- | --- |
| Figure 1: No Finger | Figure 2: One Finger |
| Figure 3: Two Fingers | Figure 4: Three Fingers |
| Figure 5: Four Fingers | Figure 6: Five Fingers |

* Convex hull defects

The convex hull defects methodology uses a mathematical algorithm in finding the convex hull and its defects. The visualization of the convex hull defects is as shown below in Figure 7 where the acute angles are defects.

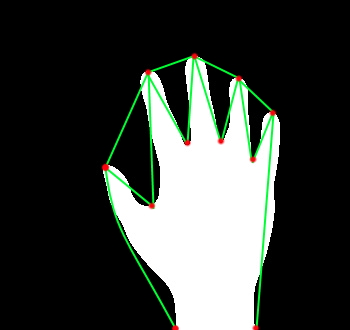


Figure 7: Convex Hull Visualization

* Convolution Neural Network

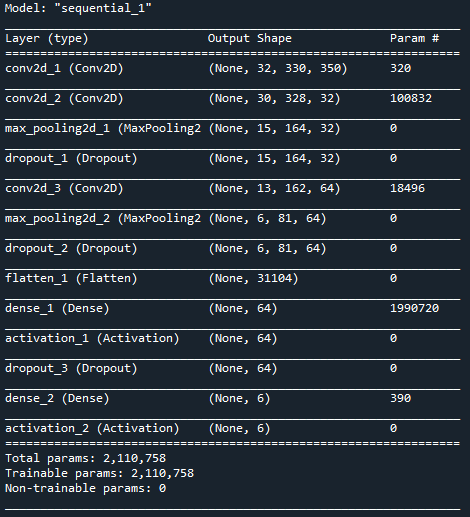
The convolution neural network is used to find patterns within pictures. The model was trained with the variations of pictures similar to figures 1-6. The model structure is as shown below in figure 8.  
****

Figure 8: CNN model structure

1. **Results and Findings**

Since the images were processed using the Otsu’s binarization method as a result, the accuracy was highly reliant on the result of Otsu’s binarization, this method has a huge flaw because it is very reliant on the lighting of the environment as well as the object.

* Convex Hull Defect Algorithm:
  + Pros:
    - Very consistent when there are no background disturbances
    - Not resource intensive
    - Edge points can be used
  + Cons:
    - This algorithm is not very flexible you are limited to 5 gestures
    - Very prone to error due to the shape of the hand
    - Result is fully reliant on the image processed from Otsu’s binarization
* Convolutional Neural Network:

In the convolution neural network, 6 different categories of images have been trained.

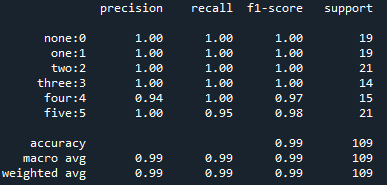


Figure 9: Accuracy table

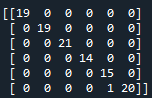


Figure 10: Confusion metrics

As can be seen from figure 9-10, the accuracy of the model is fairly high, note that this may be because of the small variation within the background. While testing, one of the flaws that could be noticed was that the lighting conditions caused the accuracy to lower greatly. In addition to this, while backgrounds still effect the accuracy, the CNN model performed better than the convex hull algorithm.

* Pros
* Great accuracy even with variance
* Not as susceptible to the change in environment
* Cons
* No localization

Suggestions: Because CNN is used for pattern recognition, if the CNN were to be trained without being processed by the binarization, it could result in a better detection rate for both varying lighting conditions and background.

The implementation with PyautoGUI:

* Convex hull defect algorithm works on a basis of points, this means that there are reference points that can be used to move the cursor. This resulted in a more intuitive control of the system. For example, mouse movements move corresponding to the tallest point of the finger. The negative of this method is due to the nature of the algorithm, once introduced with more defects than can normally be seen, it fails to work.
* CNN was able to detect and classify the gestures, but because there are no reference points the movements happened linearly. If somehow the hand could be localized, it would provide a reference point that could allow for the control of the system to be more intuitive while keeping the accuracy of the CNN.

1. **References**

Borba, F (2019), Tutorial: Using Deep Learning and CNNs to make a Hand Gesture recognition model, retrieved from: https://towardsdatascience.com/tutorial-using-deep-learning- and-cnns-to-make-a-hand-gesture-recognition-model-371770b63a51

Sadaival, Hand-Gestures (2018), GitHub repository, https://github.com/Sadaival/Hand-Gestures