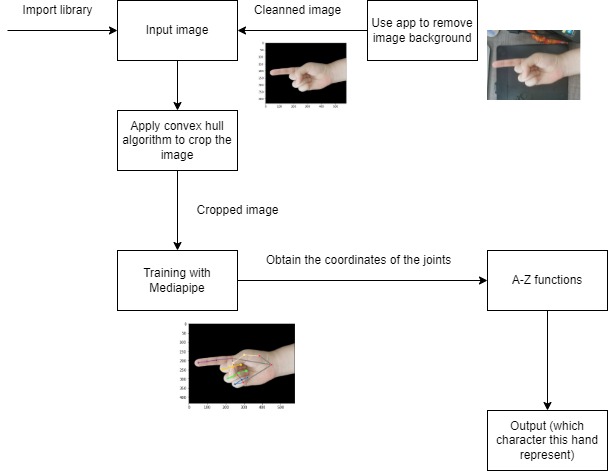
README

The purpose of the current project is to classify hand gestures into characters in American sign language. The brief procedure of our algorithm is illustrated in Fig. 2.

Graphical user interface, text, application

Description automatically generated

**Fig. 1** [**American Sign Language**](https://www.amazon.com/Gerard-Aflague-Collection-American-Language/dp/B076JTXSYG)

****

**Fig. 2 Program Flowchart**

**Step 1: Input**

The input for our code should be a hand image with its background cleaned. Our team members capture those hand images using their own hands, and we apply several methods to remove the background.

**The hand should have its palm facing toward the screen and its finger pointing to the left.**

A close-up of a hand

Description automatically generatedA close-up of a hand

Description automatically generated with medium confidence

**Fig. 3 Sample inputs**

Another kind of input is the 2d images converted from xyzm (3D file) using MATLAB. Using this input, you can also obtain the z coordinate data of every point on hand from xyzm. Currently, we are not using z data in our algorithm.

A picture containing person, hand

Description automatically generated A pair of hands holding each other

Description automatically generated with low confidence

**Fig. 4 2D outputs from xyzm file**

**Background removal:**

We use a manual method to obtain a cleaned hand image to remove the background. This includes using some apps such as photoshop, PPT to achieve the goal.

A hand with a thumb up

Description automatically generated with low confidence

**Fig. 5 Original photo and cleaned photo**

Also, if you are using the 2D images generated from xyzm files, you can apply masking in MATLAB to clean the image. Mask is an array that contains only 0 and 1.

A picture containing text, handwear, silhouette

Description automatically generatedA hand with a black background

Description automatically generated with medium confidence

**Fig. 6 Mask and image covered with the mask**

**Step 2: Training**

We use the Mediapipe library to find the landmark of a hand. The output for training can be shown in

A picture containing hand

Description automatically generated A picture containing indoor, colorful, dark

Description automatically generated

At every point on the hand landmark, we assign a number to it according to figure 8. After training, the training step will find the x and y coordinates of all those points(joints).

**Fig. 7 Hand with landmark**

**Text

Description automatically generated**

**Fig. 8** [**Landmark label**](https://google.github.io/mediapipe/solutions/hands)

**Step 3: Find the relative position**

Given the coordinate of the joints, we can find their relative position to figure out how fingers are oriented. For example, the figure below is illustrating a ‘P’ hand. It can be observed that points 5-8 are on index fingers. Point 8 has smaller x coordinates than point 7, and point 7 has smaller x coordinates than point 6, etc. Based on this, we can find all different fingers’ positions and compositions to distinguish different gestures.

A screenshot of a cellphone

Description automatically generated with low confidence

**Fig. 9 P Hand**

**Step 4: Test the hand**

We have finished writing all 26 characters’ functions, calling them will using the points(joints) coordinates to distinguish the character a gesture is representing. However, those functions have many limitations, such as the angles of how a hand is taken. So they are still under modification.