Image Understanding WS 2019
Hand-gesture recognition

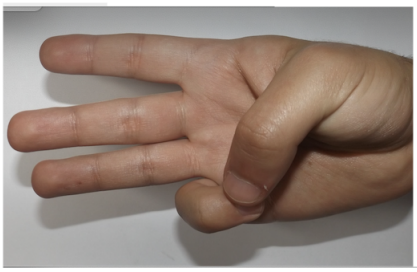
Shalini Maiti 11834150, Chetan Srinivasa Kumar 11839024

January 2020

1 Problem description

We attempt to classify static hand gestures from an image. We used four gestures for a proof of concept of our approach.



2 Our Approach

We take inspiration from a paper titled A Simple and Effective Method for Hand Gesture Recognition¡add reference¿. We detect the following gestures: **open_hand, thumbs_up, v, and three**
It works in three steps:

a) First, the hand region is segmented from the background by thresholding pixels in the HSV colorspace with respect to an empirically obtained skin HSV value. This yields a binary segmentation of the hand assuming the skin color range is respected in the background.

b) Second, on this segmented mask, a palm point and mass center of the hand region are calculated. The line joining the two is called baseline. Using this baseline, something the authors call distance signature of each point on the hand is calculated. This is our feature descriptor for the gesture.

Computing the palm point position: The palm point is computed as the point on the distance map of the segmented hand, that has the maximal value.

Computing the mass center(mc) position: It is defined as the sum of the pixel positions contained in the hand divided by the number of pixels, i.e, $(x_{mc}, y_{mc}) = (\frac{\Sigma x_i}{n}, \frac{\Sigma y_i}{n})$
where (x_i, y_i) = position of pixel in the hand region, n=no. of pixels.

Computing the feature descriptor: We use the baseline(described above) to compute the distance signature of each pixel in the boundary of the hand region. We parameterize each value with (d, θ), where d=distance of the edge pixel from the palm point and θ = angle subtended by the line joining the pixel and the palm point. It can be seen that this particular descriptor is invariant to rotation of the hand. Additionally, bin averaging and normalisation is done across different bins of this histogram. This is done to make the descriptor invariant to image scale.

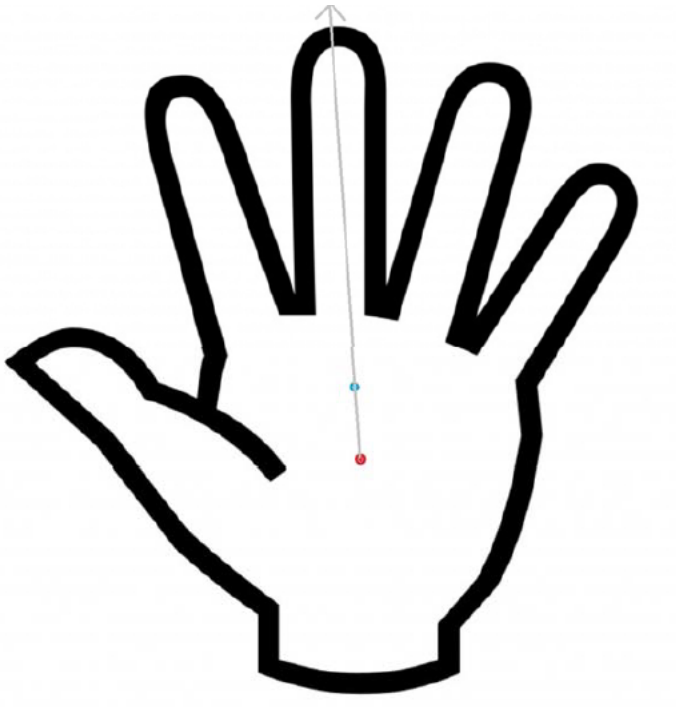


Figure 4. The palm point(the red point), the mass center(the green point) and the baseline(the arrow)

c) Third, we use KNN to train the classifier. We have found that the optimal hyperparameters are 8 neighbours, with inverse distance weighting and euclidean distance metric.

d) Note: All the images taken for training and testing are taken against a homogeneous, white background without producing shadows. We also center our hands in the frame in a bounding box to ensure such that the wrist doesn't feature into the images too much.

3 An aside - Segmentation

We began the project with an idea to find a fast hand segmentation method, and started with using HSV values to detect the skin of the hand. However, this method proved sensitive to shadows, and yielded incorrect segmentations if the background HSV was in our target HSV value range.

To deal with these issues, we developed the following approach:

1. Run HSV segmentation and obtain a rough mask.
2. Use the GrabCut algorithm to refine the mask for 5 iterations.
3. Run the HSV segmentation again on the refined mask to cut away any cluttered "probable" foreground regions.

This approach yielded much better segmentations, though with problems that shall be detailed in the coming sections. A tradeoff in this approach was speed - our code, already hampered by python bindings, had to sacrifice more time for the grabcut refinement.

However, it must be noted that the code was run for larger images. It is indeed very possible to make the system run at framerate by using smaller images, and optimizing grabcut to run on GPU.

4 Environment and Code Structure

We require python and opencv-python to be installed on the host. As this is a proof of concept, we use python entirely to prototype our ideas. The file **train.py** contains interfaces to generate descriptors for the training images as described in the above sections, and save said descriptors and associated labels.

test.py contains a function to fit a KNN model to the descriptor/label pairs.A **main** function is provided where the prediction can be run for any folder(s) of test images.

Without modification of the in-code paths, the code will run the tests on the folder training_012, a custom dataset we have collected ourselves.

5 Dataset

To fulfill our initial requirements of plain backgrounds and centered palm pictures, we created our own training and test datasets. For each of the gestures **open_hand, thumbs_up, v, and three**, we collect more than 60 images differing in orientations, translation and scale.

It must be noted that we chose these 4 gestures to test how well the system distinguishes gestures with inter-class variance.

6 A Discussion of the Results

We report the best accuracies for detections of the four gestures below with our datasets, and propose reasons for these figures.

open-hand : 73%
thumbs-up : 51%
v: 30%
three : 51%

We note the unsatisfactory results of the 'v' gesture. As our descriptor is one that encodes angles of the hand-contour pixels with the line joining the palm and center of mass points, it is quite sensitive to the segmentation that yields the contours.

It is possible for a 'v' gesture segmentation to look like our 'three' gesture segmentation if all the fingers are not properly delineated in the mask. Indeed, we see while running the classification that most of the wrong predictions for 'v' are mis-classifications as 'three'.

Similar effects abound, apparently, with the rest of the data set. The 'three' gestures can look similar to 'open-hand' without precise segmentation.

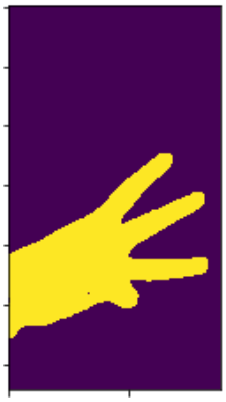
The major issue, however, is the very nature of the descriptor we use. "v", and "three" gestures, as we see from the mask, have similar boundaries but for the one extra finger. This is a problem, as while creating our descriptor which is a histogram of angles made by the hand boundary with the baseline. We show segmentations of such misclassifications here for each class to illustrate this point.

Open-hand Misclassified as three:

Thumbs-up misclassified as open-hand:

V misclassified as three:

Three misclassified as open-hand:



7 Conclusions and Possibilities

We conclude that the descriptor could perform better with small inter-class variances. It is possible that there is a selection bias in our dataset, as one of us(Chetan Kumar) collected the training set entirely with my left hand, with more palm and less arm. The test set was by Shalini Maiti which consisted of more arm with left hand. Thus, a more balanced dataset for training could perhaps aid with the training.

It is also to be noted that the hand segmentation heavily influences the process. Our Segmentation appears to perform well with plain background hands, but we sacrifice time for this refined mask.

Also pending is an exhaustive hyper-parameter search (kNN params, histogram bins).

8 References

A Simple and Effective Method for Hand Gesture Recognition - Chunying Quang, Jianning Liang