Hand Gesture Recognition

Marin Andrei-Mihai AI

We aspire to develop an application that recognises gestures using a camera and software, without the need for a specialized glove, making it more practical and also reduce cost of the consumer, while also making it as reliable as possible.

We are going to use hand gesture recognition in home automation for people that are mute or speech impaired.

Such a system could perform some of the functions that a home automation systems that uses voice as a means to control the objects in the house. Complex voice commands such as "conversations" with Amazon Alexa won't be achievable, but simple, repetitive commands such as what's the weather today could be achieved.

Such functions could be customized by the user, and a mapping between hand gestures and functions could be established.

One of the key challenges it will be the fact that accurate gesture identification cannot be easily achieved from any angle and from any distance to the camera.

The way we can get around this is by using a mobile phone, the user could activate his front facing camera and make one or multiple signs. The pros about this approach is that variability in the background is reduced, and the hand is near the camera so that a higher accuracy could be achieved, the con is that only one hand will be able "active" at a time, so we have less functions that can be mapped.

Our application that identify the hand gestures will communicate over WiFi with other IoT devices in the house, even though in this PoC that feature will not be available.

We will use this paper[1] as a reference for the way we implement the hand recognition algorithm. There are multiple alternatives of both language and framework in order to build the application. For the hand gesture recognition part of the app, we will use Python and Tensorflow Lite.

Tensorflow Lite is good for inference on the mobile, due to it's reduced size. The model will be trained on the dataset from [2] . They have 47933 RGB-D gesture videos and 249 gesture labels. One of the problems with this dataset is that some gestures are executed using two hands.

This dataset is better for our case than the one presented in the paper because in [2] the images are more similar to the one that the phone camera will take.

We will do classification only on static hand images, meaning that the transition of the hands from one gesture to another with not be taken into account. Only when the user will stop moving, the model will classify the gesture.

The paper uses a hand segmentation solution using convolutional residual network. Unlike many previous methods that use threshold based techniques to do hand segmentation we will use machine learning.

The HDG-Net is composed of two stages:

1. A deep fully convolutional residual networks and then add on top of it a Atrous Spatial Pyramid Pooling module and after that upsample the resulting image
2. A two-stream CNN leading up to a fully connected layer and a softmax to classify the hand gestures.

The deep fully convolutional residual network is used to learn useful representations.

The atrous spatial pyramid pooling is used to capture multi-scale information.

After the first stage the result we will have is a black and white image of the hand without the initial background.

The second stage is used to classify the gestures using the two stream CNN approach.

One stream extracts features from the raw image, and the second extracts features from the segmented image.

The two of them are fused before the fully connected layer.

A probable challenge of this model is the number of parameters it has. The model seems complex and I'm not sure it will work on mobile. The training part will be done at the servers of the company, but the inference side on the Tensorflow Lite that is done at the customer device might be too slow.

Week1:

Getting familiarized with the technology such as Tensorflow Lite and convolutional residual networks. Also download the dataset and hand pick the images that represent the gestures that we want to classify.

Week2:

Start programming the HGR-Net into Tensorflow Lite and then test an initial model on a few mobile devices.

Week3-5:

Play with the model parameters and different depths and widths of the neural networks. And also fix bugs and resolving problems that are bound to happen.

Week6:

We will test the implementation on multiple android devices, with different android versions and multiple camera qualities, by now most of the implementation should be done and we should be in the testing and validation stage.

References:

[1] https://arxiv.org/pdf/1806.05653.pdf
[2] http://gesture.chalearn.org/2016-looking-at-people-cvpr-challenge/isogd-and-congd-datasets