**Reference:**

[*https://pythonprogramming.net/haar-cascade-object-detection-python-opencv-tutorial/*](https://pythonprogramming.net/haar-cascade-object-detection-python-opencv-tutorial/)

The hand images aren’t available in image-net. Hence dataset is obtained from kaggle.

**September 24, 2020**

Datasets Downloaded

Positive images:

1. <https://www.kaggle.com/ashish8898/hand-gestures>?
2. <https://www.kaggle.com/ahmedkhanak1995/sign-language-gesture-images-dataset>
3. <https://www.kaggle.com/datamunge/sign-language-mnist>

(one of these which is in csv format is not used)

Negative images:

1. <https://www.kaggle.com/alxmamaev/flowers-recognition>
2. <https://www.kaggle.com/alessiocorrado99/animals10?select=raw-img>

Data preparation

1. Handpicked the positive images with diff gestures, lighting, background and skin tone.
2. Choose any of the pics for negative images.
3. pos/ contains 1521 images and neg/ has 722 images.
4. Convert all images to grayscale.
5. Resize all positive and negative images to 50x50 pixels.
6. Rename all images in pos/ in sequential number form and obtain the info.lst file.

Eg. Location/filename.jpg 1 0 0 50 50

1. Create another file bg.txt for images in neg/

Training

1. *opencv\_createsamples -info info.lst -num 1520 -w 20 -h 20 -vec positives.vec*

This is to get positives.vec file from info.lst file. We don’t need to create new samples since we already have our positive images predefined.

1. *nohup opencv\_traincascade -data data -vec positives.vec -bg bg.txt -numPos 1400 -numNeg 700 -numStages 10 -w 20 -h 20 &*

This is the training command. It requires new positive images while training. Hence always give the positives in the cmd to be less than the total number of positives present. Here, 1521 positives are present. But 1400 is given as numpos in the cmd. The numneg should be half the numpos. Nohup records all the training logs in nohup.out. If I need to continue the training till 20 stages, then the training will continue from 10th stage if all other parameters provided are same except the numStages. The output is stored in data/.