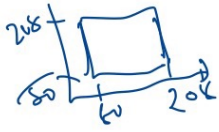
Day 15

16th December

Day 15 Tic Tac Toe

```
import joblib,cv2
import numpy as np
model = joblib.load("model/svm_6label_rbf")

import pyscreenshot as ImageGrab
import time

images_folder = "temp/"
fout = open("testing_x","w+")
for i in range (0,100):
    img = ImageGrab.grab(bbox=(80, 80, 208, 208)) # X1,Y1,X2,Y2
    img.save(images_folder+"test_orig.png")
    im = cv2.imread(images_folder+"test_orig.png")
    im_gray = cv2.cvtColor(im, cv2.COLOR_BGR2GRAY)
    im_gray = cv2.GaussianBlur(im_gray, (15, 15), 0)

    # Threshold the image
    ret, im_th = cv2.threshold(im_gray, 100, 255, cv2.THRESH_BINARY)

    roi = cv2.resize(im_th, (28, 28), interpolation=cv2.INTER_AREA)

    cv2.imwrite(images_folder+"segmented.png", roi)

    rows,cols = roi.shape

    X=[]

    # #Add pixel one-by-one into data Array.
    for i in range(rows):
        for j in range(cols):
            k = roi[i,j]
            if k>100:
                k=1
            else:
                k=0
            X.append(k)

    #scaling = MinMaxScaler(feature_range=(-1, 1)).fit([X])

    #X = scaling.transform([X])
    fout.write(str(X))
    predictions = model.predict([X])
    print "Prediction: ", predictions[0]
    cv2.putText(im, "Prediction is: "+str(predictions[0]), (20, 20), 0, 0.8,
     (0, 255, 0), 2, cv2.LINE_AA)

    #cv2.startWindowThread()

    cv2.namedWindow("Result")
    cv2.imshow("Result", im)
    cv2.waitKey(10000)
    #time.sleep(4)
```



Loading the model that was saved

Image pre-processing

grabs this dimension

save this in images-folder, and call

read the image

convert from BGR to gray

apply a blur

why are we doing all this

2 outputs

img size (rows, cols)

converting pixels to black & white

??

predicting y!

Show Predictions