

# Week 6

## What has been done this week

This week I have tried to see if I can identify the number / letter on the club. I have also trained the detection model some more and annotated the rest of the data.

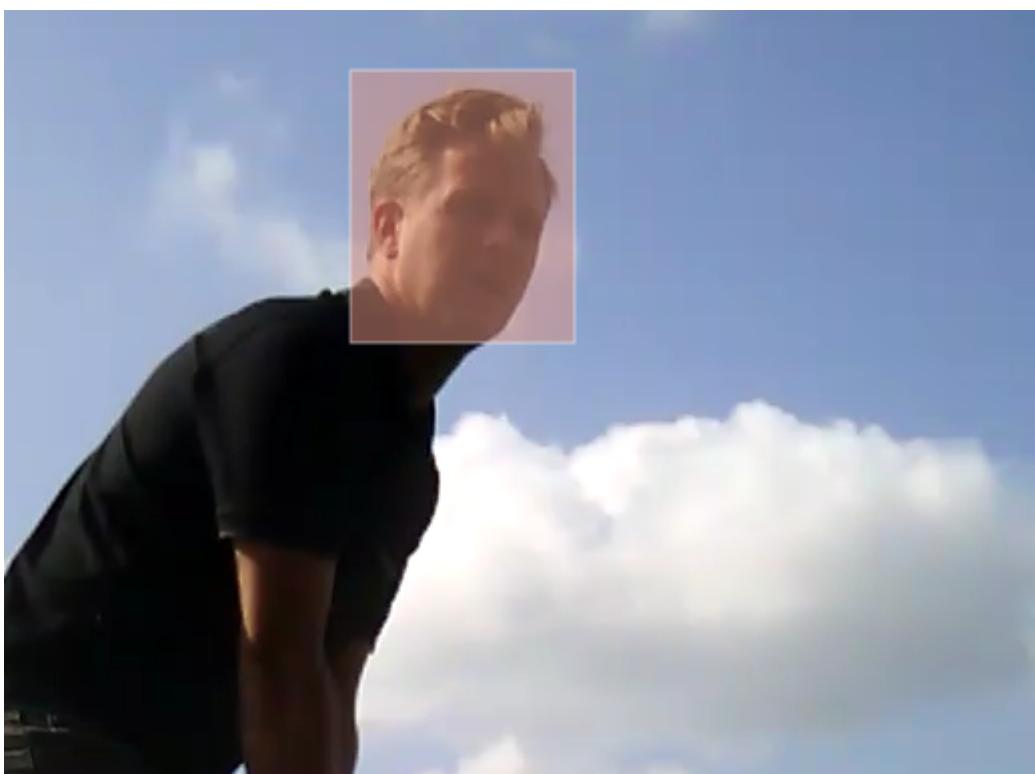
**Number of videos: 281**

**Duration (HH:MM:SS): 00:37:55**

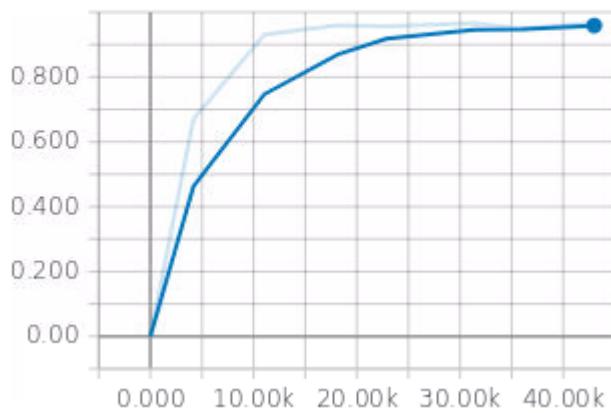
**Frames: 102454**

- Rename all the video to have the correct format
- Regenerate all the annotations with latest up to date model

Finding: Model seems to detect faces as clubs quite frequently:



AP@0.5IOU/b'club'  
tag: PascalBoxes\_PerformanceByCategory/AP@0.5IOU  
/b'club'



- Grab a larger set of club images for identification.
- Grab set of cropped images to see performance with tesseract.

Getting accuracy of 70% on a smaller set by combining two models (number 8 and 10 in tesserect), models individually have 55% accuracy, but combining them we get to 70%.

```
matrix = np.zeros_like(df, dtype=np.float)
correct = [0]*len(methods)

rr = 0
for index, row in df.iterrows():
    for i, r in enumerate(row):
        if row.name in r.lower():
            correct[i] += 1
            matrix[rr][i] = 1
    rr += 1

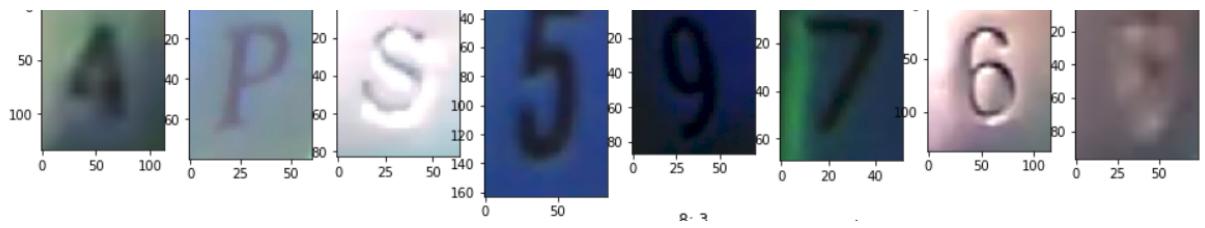
for i in range(len(correct)):
    correct[i] /= len(df.index)

print(np.sum(np.sum(matrix == 1, axis = 1) != 0) / len(df.index))

df2 = pd.DataFrame(columns=methods)
df2.loc[0] = correct
df2
```

0.7037037037037037





- Add a neural net trained on house number data: Combining the Neural net with tesseract, we get to 85% accuracy. The net is much better at digits, but can obviously not detect any letters.

```
matrix = np.zeros_like(df, dtype=np.float)
correct = [0]*(len(methods)+1)

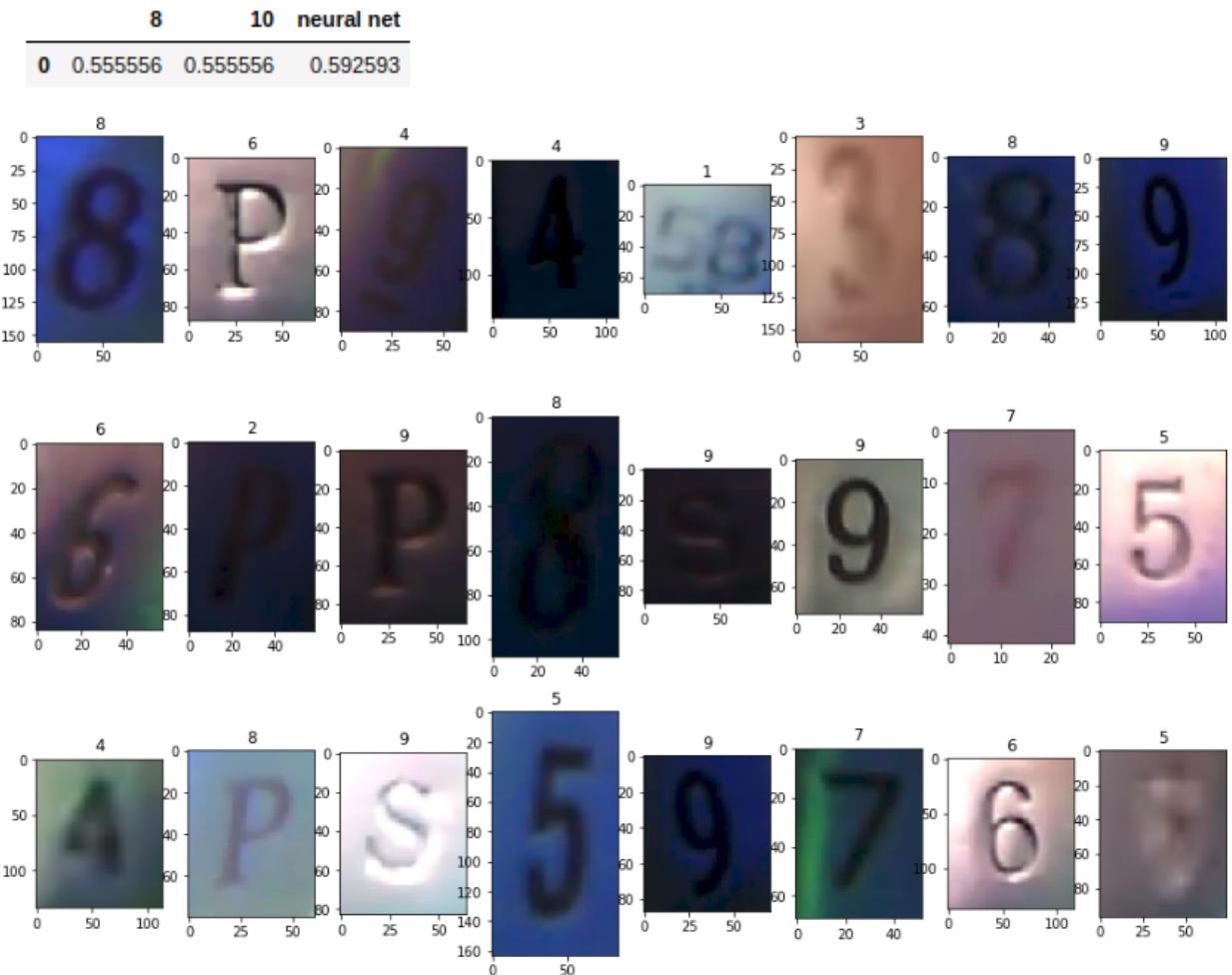
rr = 0
for index, row in df.iterrows():
    for i, r in enumerate(row):
        if row.name in r.lower():
            correct[i] += 1
            matrix[rr][i] = 1
    rr += 1

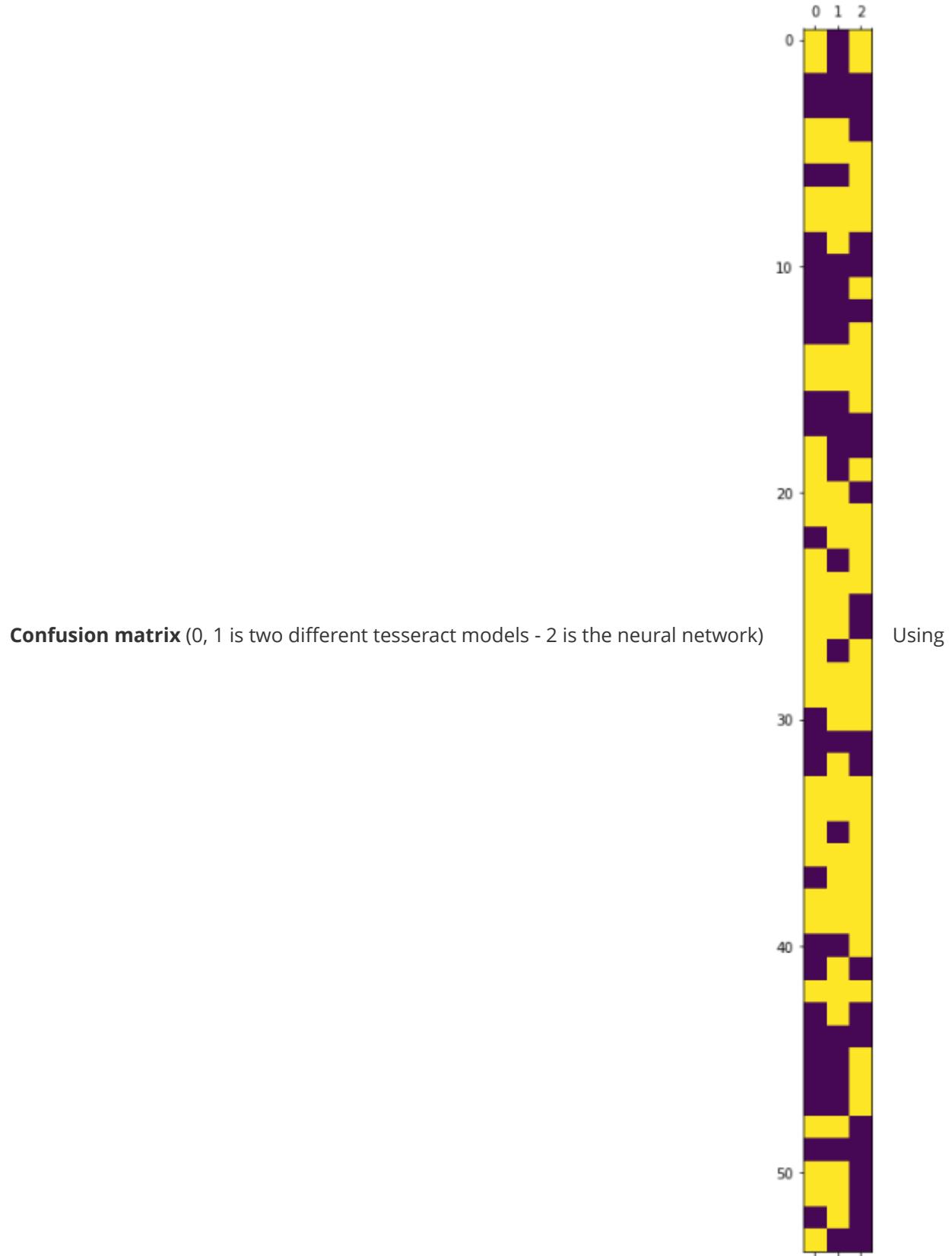
for i in range(len(correct)):
    correct[i] /= len(df.index)

print(np.sum(np.sum(matrix == 1, axis = 1) != 0) / len(df.index))

df2 = pd.DataFrame(columns=methods + ['neural net'])
df2.loc[0] = correct
df2
```

0.8518518518518519





an even better neural network (trained myself on the same data, but a better model) we get 92.6% acc:

```

matrix = np.zeros_like(df, dtype=np.float)
correct = [0]*(len(methods)+1)

rr = 0
for index, row in df.iterrows():
    for i, r in enumerate(row):
        if row.name in r.lower():
            correct[i] += 1
            matrix[rr][i] = 1
    rr += 1

for i in range(len(correct)):
    correct[i] /= len(df.index)

print(np.sum(np.sum(matrix == 1, axis = 1) != 0) / len(df.index))

df2 = pd.DataFrame(columns=methods + ['neural net'])
df2.loc[0] = correct
df2

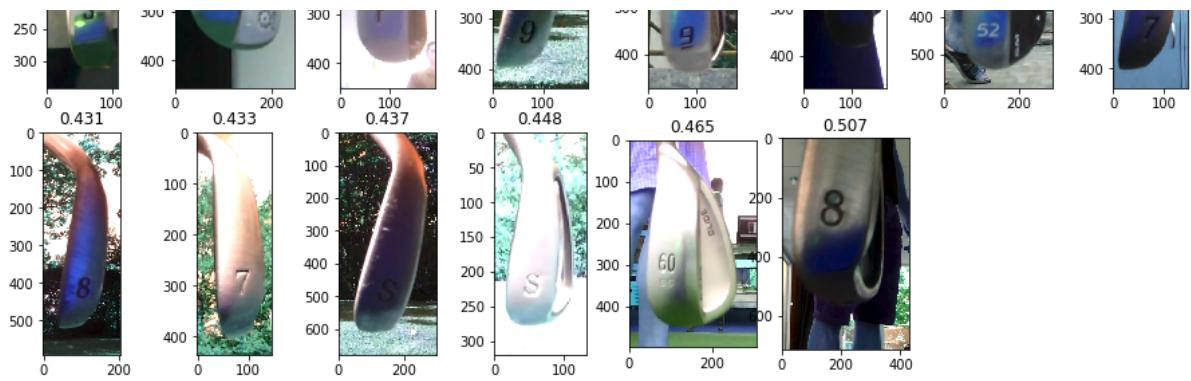
```

0.9259259259259259

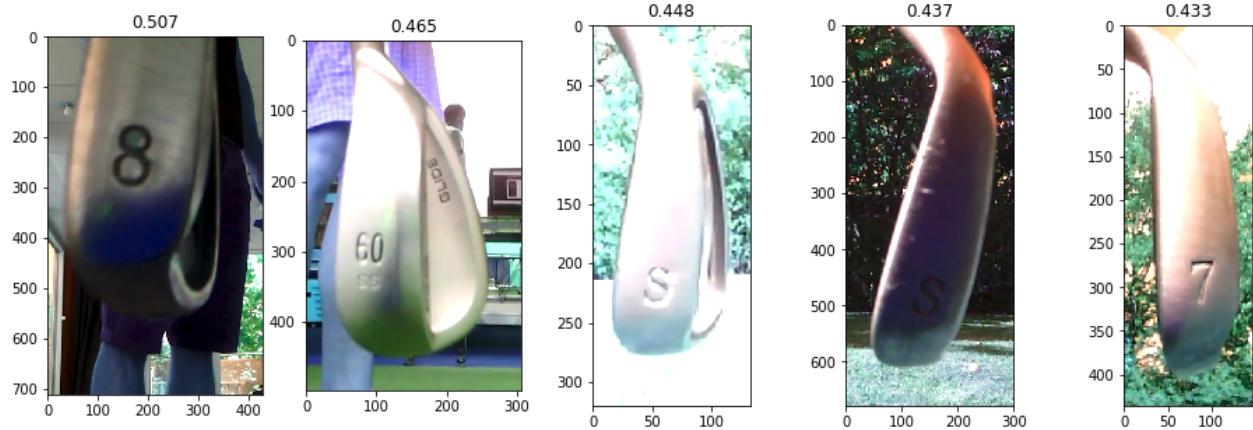
	8	10	neural net
0	0.555556	0.555556	0.703704

- Research literature of multi digit classification from images. From some quick research it seems models trained on house numbers is the best approach here.
- Try facenet encoding of the clubs Did actually work without any retraining:

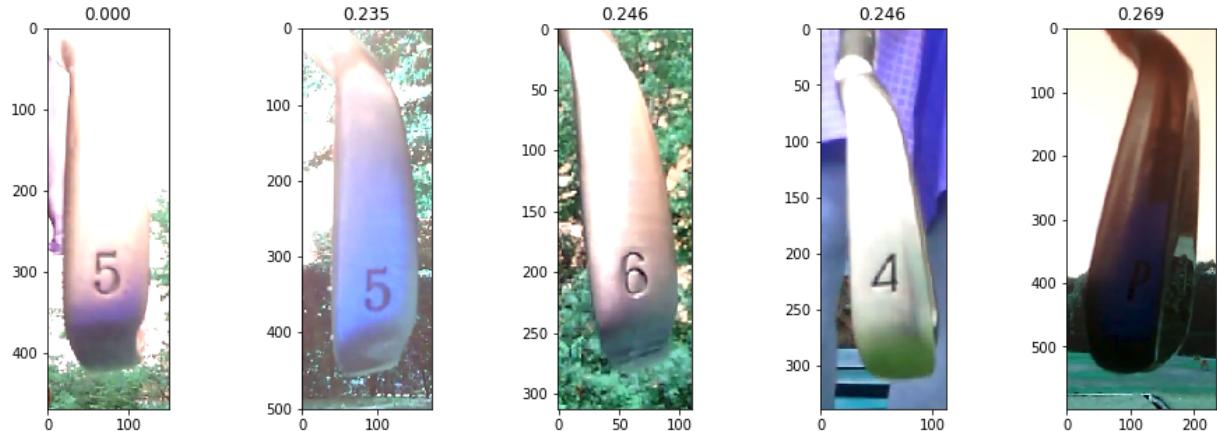




LONGEST DISTANCES



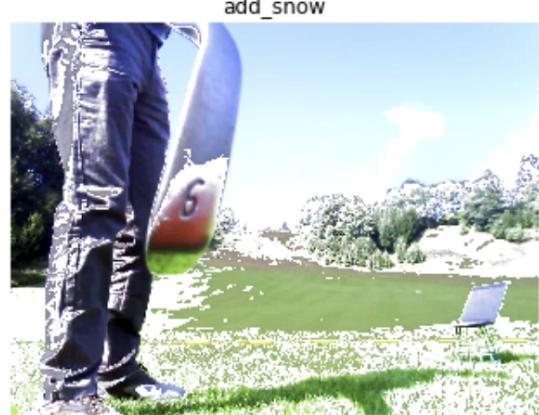
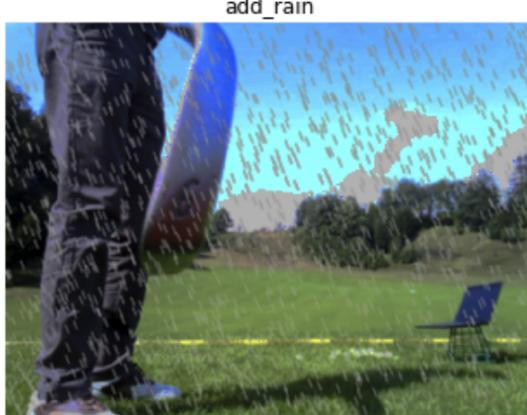
SHORTEST DISTANCES



## Questions / Difficulties

- The main problem with the project is still data. I can artificially generate some data that might help with detection, but classification is still hard to generate data for. From an engineering perspective, the absolute best and more sensible thing to do would be to go out and collect more data. A stupid model

with more data almost always beats a clever model with little data.



- Encoding the image of the club and storing a database of known club encodings seems like a much more sensible approach - This way it would also be possible to do recognition of drivers / woods etc. However FaceNet and similar networks need ALOT of data (DeepFace is trained on 3 million images). So again we have a case of limited data making it really hard to do anything.

## Status according to project plan

Week 6 should actually have been classification of main categories of clubs but I feel that I have to spend some time right now to see which direction we can take this project. Week 6-10, should in general be solving the main problem. I think the detection model using RCNN is good where it is now - we could try to get a good model to work with SSD, but that have quite a lot of problems learning small objects.

## What to do next week

- Classification of 4 main categories just so see if we can get to our desired 95% accuracy on the current data.
- Record a bit of validation data (30-40 videos in various conditions) and get accuracy metrics for the different detection models that has been trained.

- Make a method to find the number in the cropped image of the club to pass to model trained on the house numbers.