

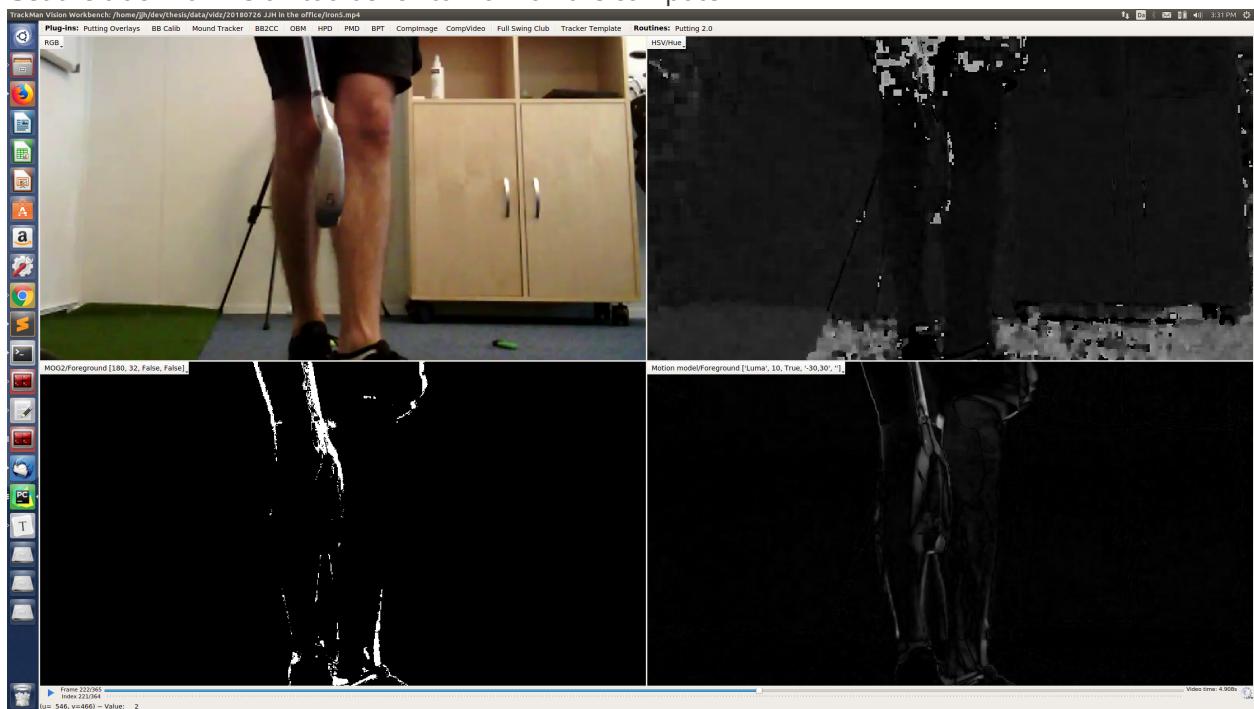
# Week 4

## What has been done this week

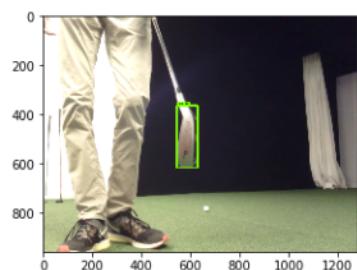
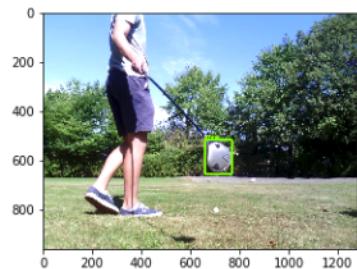
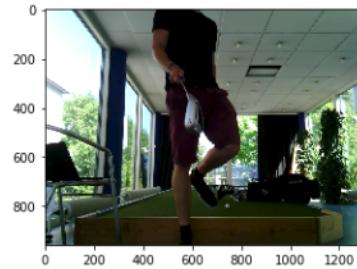
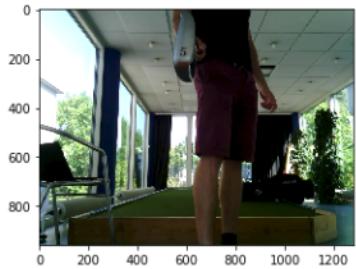
I'm still having trouble focusing so I try to just grab the low hanging fruits in the project to have some kind of progress.

I've tried making a bit better detection model but a restructure is needed and that seems like a bigger job. I've also tried to train a basis classification model which didn't give too promising results and lastly I have been playing with creative ways to generate more data.

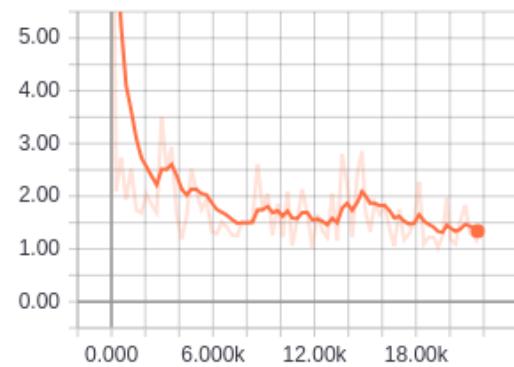
- Write introduction to the report.
- First iteration
- Modify frame extraction script to be able to crop the images
- Try out pytesseract (Google OCR tool) **Results didn't look very promising, retraining or image preprocessing needed for the clubs**
- Finish up project plan
- Get the trackman vision toolbench to work on the computer



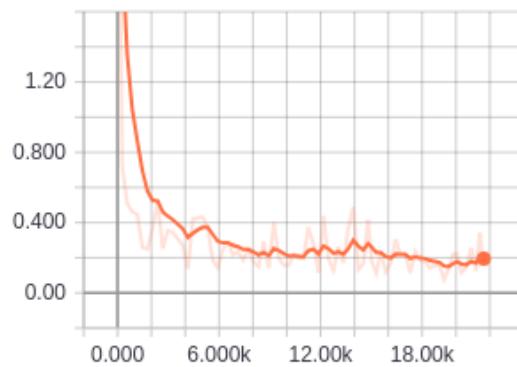
Train detection model for longer



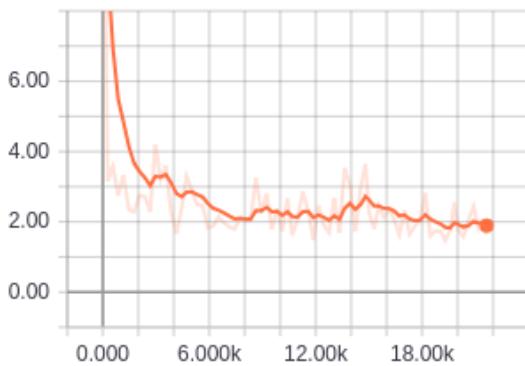
Losses/Loss/classification\_loss



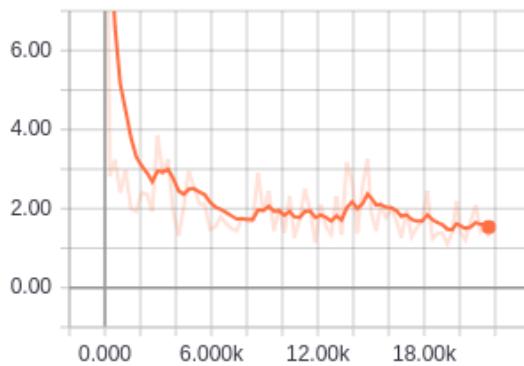
Losses/Loss/localization\_loss



Losses/TotalLoss



Losses/clone\_loss

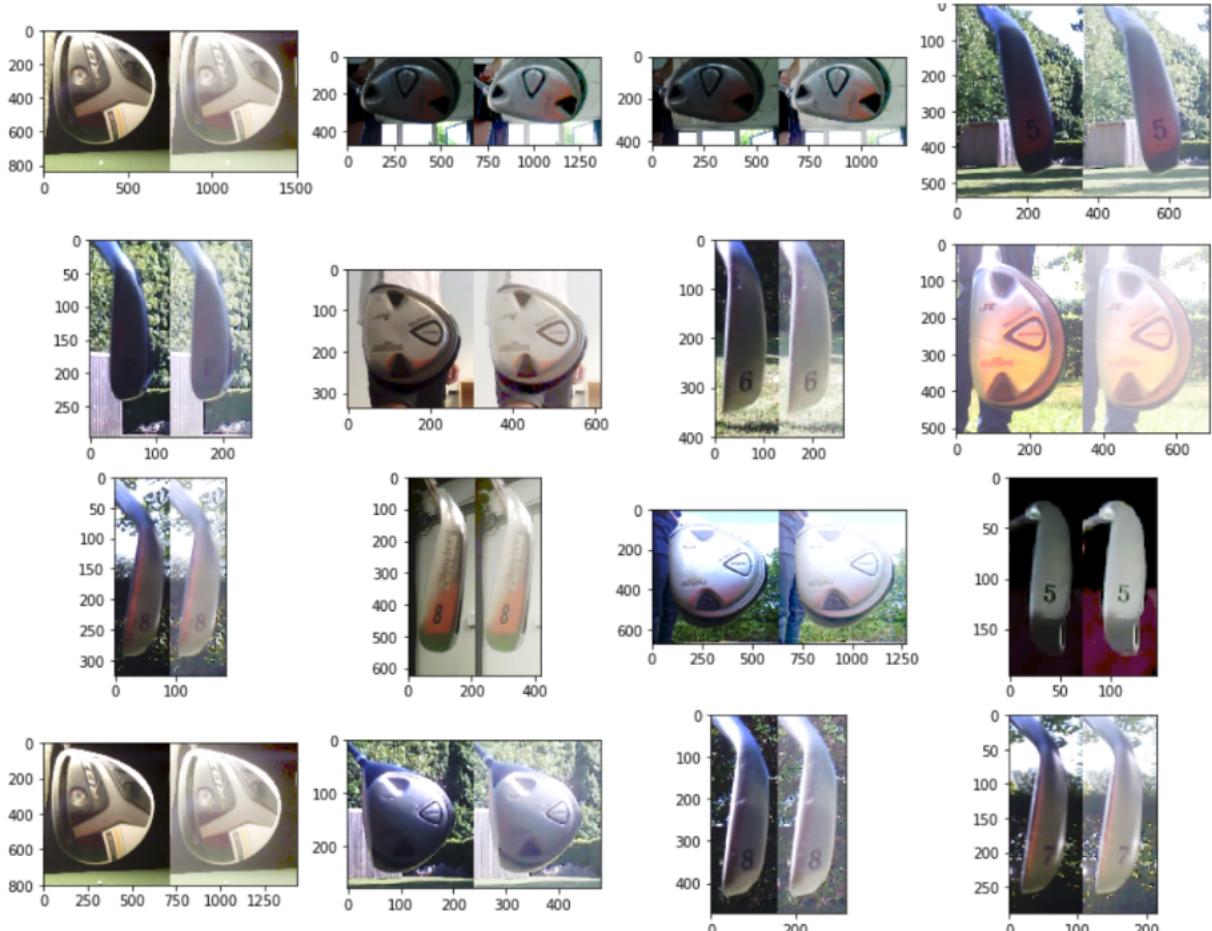


- Try different models for subtracting backgrounds Pretty basic but just take the initial image, blur it and then compare with blurred versions of frames captured later. If there is a large difference, there must be a change from the background.





Methods for color correcting the images



It seems like we can't learn the difference between a 5 and a 9 iron (**2 class classification problem**) right now: (DenseNet121, trained on Imagenet with a new top layer). Network was trained on the cropped images. It is able to overfit the dataset, but it doesn't seem to learn the number.

```

Epoch 00021: val_acc did not improve from 0.62637
Epoch 22/50
26/26 [=====] - 7s 251ms/step - loss: 0.2946 - acc: 0.8700 - val_loss: 0.8529 - val_acc: 0.5465

Epoch 00022: val_acc did not improve from 0.62637
Epoch 23/50
26/26 [=====] - 6s 234ms/step - loss: 0.3174 - acc: 0.8724 - val_loss: 1.1927 - val_acc: 0.5055

Epoch 00023: val_acc did not improve from 0.62637
Epoch 24/50
26/26 [=====] - 7s 252ms/step - loss: 0.3225 - acc: 0.8558 - val_loss: 0.8279 - val_acc: 0.5465

Epoch 00024: val_acc did not improve from 0.62637
Epoch 25/50
26/26 [=====] - 6s 231ms/step - loss: 0.2588 - acc: 0.8941 - val_loss: 1.1393 - val_acc: 0.4945

```

**However:** Training a model on classifying **driver/wood vs iron/wedge/putter** gets 97% validation accuracy in 5 epochs (2 min training time) - Super easy problem:

```
Epoch 1/10
90/90 [=====] - 30s 330ms/step - loss: 0.4538 - acc: 0.8506 - val_loss: 0.1861 - val_acc: 0.9160
Epoch 00001: val_acc improved from -inf to 0.91598, saving model to vgg16_1.h5
Epoch 2/10
90/90 [=====] - 25s 283ms/step - loss: 0.1426 - acc: 0.9486 - val_loss: 0.0674 - val_acc: 0.9707
Epoch 00002: val_acc improved from 0.91598 to 0.97067, saving model to vgg16_1.h5
Epoch 3/10
90/90 [=====] - 25s 280ms/step - loss: 0.1009 - acc: 0.9638 - val_loss: 0.0448 - val_acc: 0.9804
Epoch 00003: val_acc improved from 0.97067 to 0.98045, saving model to vgg16_1.h5
Epoch 4/10
90/90 [=====] - 25s 280ms/step - loss: 0.0628 - acc: 0.9806 - val_loss: 0.0544 - val_acc: 0.9735
Epoch 00004: val_acc did not improve from 0.98045
Epoch 5/10
90/90 [=====] - 25s 280ms/step - loss: 0.0512 - acc: 0.9806 - val_loss: 0.0505 - val_acc: 0.9763
Epoch 00005: val_acc did not improve from 0.98045
```

## Status according to project plan

---

I'm not completely done with detection, that still requires some work that I haven't really felt like looking into. A basic model is up and running however. On the other hand, I've starting looking at data generation which is first scheduled for next week as well as classification which is first scheduled in 2 weeks.

## Questions / Difficulties

---

- There is still the persisting problem of a huge variation in the club heads compared to the size of our dataset. From day 1 I will have to abandon the ambition from TrackMan that they wish to classify all the club categories. I might be able to detect the different irons, but that's about it. Many clubs, especially wedges, has little or no information on the bottom of the club making the task impossible.

It's quite demotivating working on the project because I'm working towards an end goal that is not really fantastic. The project in itself is not groundbreaking in any way, just applying CNN detection/classification to images and now the end result will not have that great value either. Currently it feels much more like a job than a project I would work on in my spare time, mainly because I'm unable to really find an angle on the project that is interesting to me. It's a hard problem, but it's mainly hard because of the lack of data and because the information on the bottom of the club is not enough to solve the general case of club identification.



- I don't think it makes sense to do a large classification model. We just need a model that can tell **iron** from **not-iron**, and then do OCR on the image to try and detect the number. I think there is too little data for the model to actually pick up that it has to look at the numbers in the images of the irons to tell them apart. An OCR trained on billions of images will furthermore be better than anything I will be able to train myself with this dataset size.

## What to do next Week

Try to do more "green-screen" recordings. Found out that it works okay in some conditions, I just need to find a background that is not white/gray which is the same color of many of the clubs. A strong Blue/Green/Red would be nice, or just any color that sticks our a lot.

- BeaverDam: Make annotation guide
- Finish up object detection
- Get Azure cloud solution up and running (trackman has a deal with MS, will get a P100)
- look into pix2pix to generate night training sets: <https://github.com/phillipi/pix2pix>
- Annotate important frames in video to feed into classification step
- Detection: Improve folder structure
- Get validation script to run and validate the results on the test set