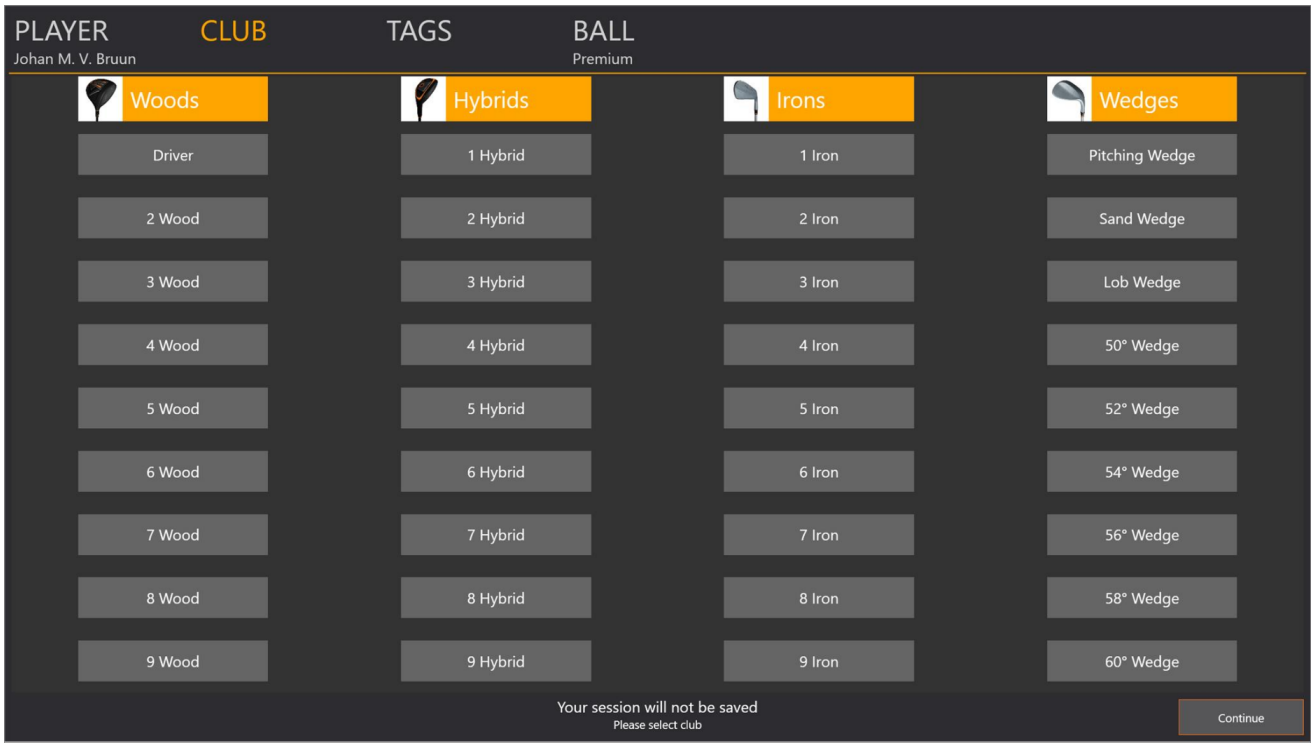
Week 2

What has been done this week

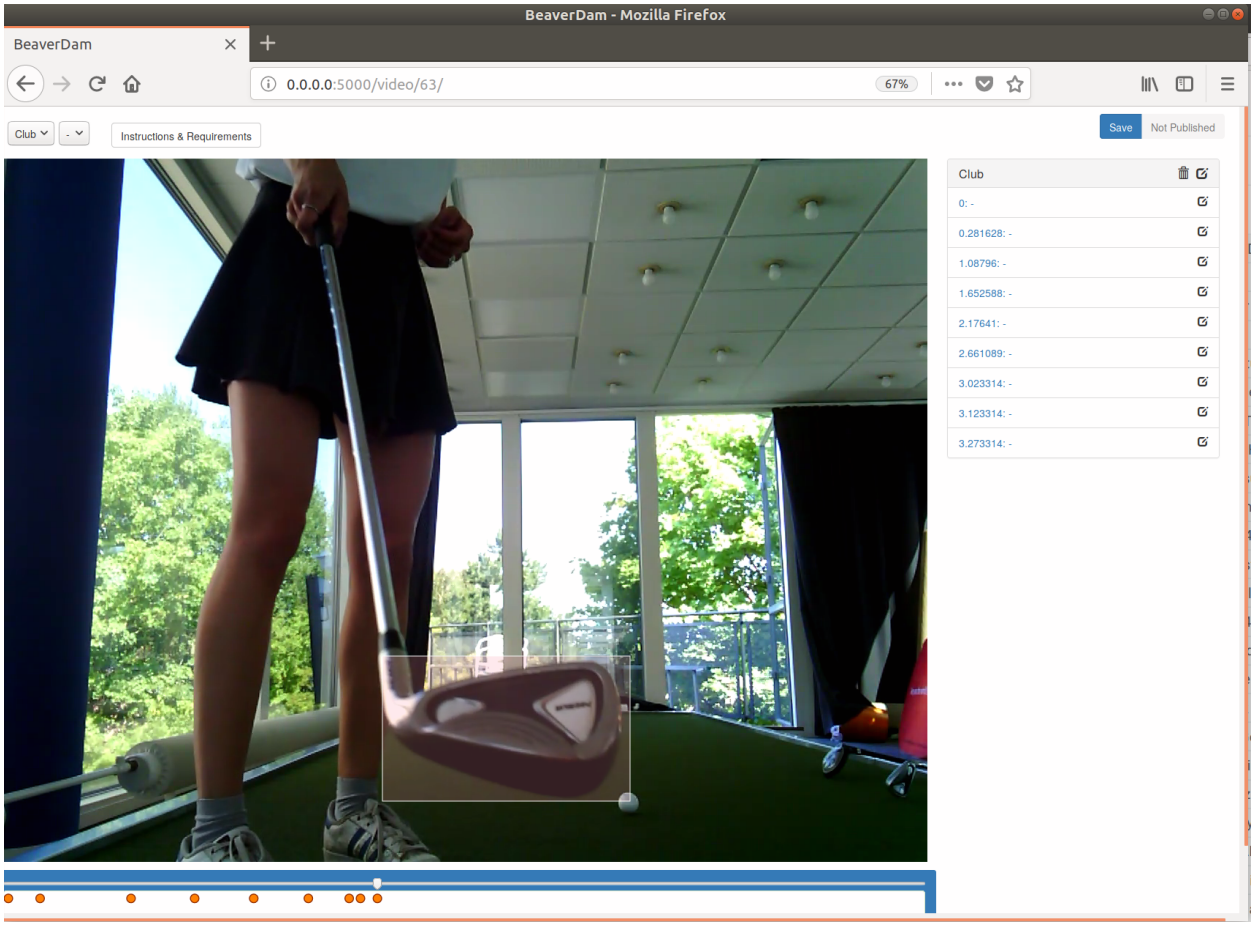
I was sick friday, saturday and sunday so not much work was done on these days.

I have been fighting more with beaverdam (annotation tool) this week but it is up and running nicely now and I am done annotation our base dataset of 142 videos. These can now be used to train a basic detection network that can assist in annotating future data.

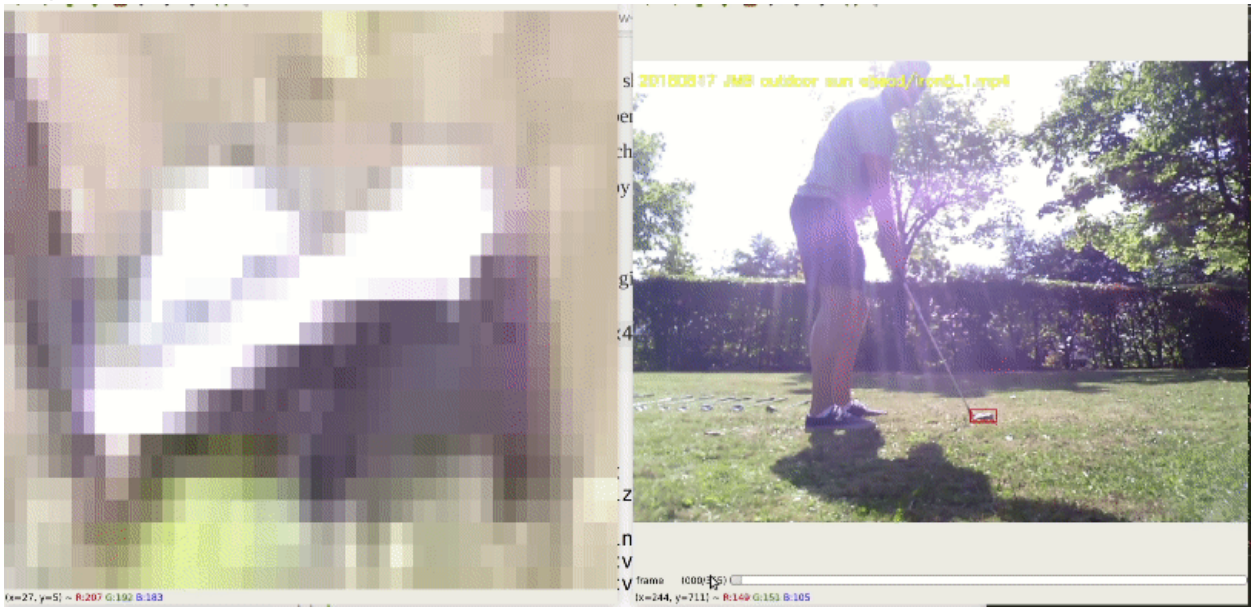


- [ ] BeaverDam:
  - [ ] Annotation guide
  - [x] Get annotations and put them into files
  - [x] Fix ID's
  - [x] Host videos locally via http using python
  - [x] Deploy on network?
  - [x] Get bounding box information
  - [x] Fix host.id on IP change for the files in the database
  - [x] Make sure that videos are deleted if they are deleted from the data folder
  - [x] Make sure videos are only added if there is not existing a video with that name already in the database
  - [x] Put into /thesis folder to have it in the same version control and easier structure for importing utilities into python etc.

- [x] Read Old Master Project
- [x] Setup Report
- [x] Upload Picture to Teams
- [x] Ensure renaming script is appropriate (Names does not change when repeatedly running the script)
- [x] Annotate Videos



- [x] Program to check annotations



Status according to project plan

This week is set to data annotation / literature study / project plan which is what I have been working on. Data has now been annotated and are ready to be used in a first iteration of a DL pipeline.

Literature

BeaverDam

To initiate database:

`.manage.py flush` | Clear the database

`./manage.py shell < ../thesis/load_data_BeaverDam.py` | Throw in own data

`./manage.py loaddata annotator/fixtures/mydata.yaml` | Put the club class into the database

`./manage.py createsuperuser --name test` | create an annotator

`scripts/serve` | serve the script

`python -m http.server` | host the videos locally

To put new data in:

Put the videos in `thesis/vidz` and run `./manage.py shell < ../thesis/load_data_BeaverDam.py`

```
>>> h.host
'https://s3-us-west-2.amazonaws.com/beaverdam/videos/'
>>> h.filename
'nscratch/bichen/videos-scenes-2016-06-19/video_0000/clip_000.mp4'
```

Janus Jensen and Morten Hannemose: "Computer Vision For Golf" 2016:

Project done in coorporation with trackman to detect the impact location between a golf ball and the club using vision and radar.

Table 4.1: Outline of our entire method.

Given a video of a golf stroke recorded at a high frame rate behind the golf player, we do the following.

1. Track the ball's flight path.
2. Determine the location of the initial ball position.
3. Use flight path and initial position to compute the impact time.
4. Determine the point on the shaft where the shaft ends and the head begins. This point is referred to as the anchor point.
   a) Detect the club shaft.
   b) Segment the club head.
   c) Determine the anchor point.
5. Fit a model to the anchor points detected in each frame of the swing.
6. Interpolate the anchor point at impact time.
7. Estimate a homography between the club face in the image plane and the club face coordinate system at impact. The club face coordinate system has origin at the impact point.
8. Use the homography to map the anchor point to the club face coordinate system.
9. Given the coordinate of the anchor point in the standard club face coordinate system with origin in the middle of the club, we can compute the impact point by subtracting the vectors from each other.

Nice method for getting a background image (maybe I can steal it, but I have a person walking around in mine too so it can be hard.)

Background image

In frames during the swing we will need to know a background image, i.e. an image with the moving parts removed. For frame f in the video we calculate this as

$$\mathbf{B}(f) = \underset{s \in \{-4,-2,0,2,4\}}{\text{median}} \mathbf{V}(f+s),$$

where $\mathbf{V}(f)$ is the f'th frame in the video. An example of a background image is shown in Figure 4.4 on page 28.

(a) $\mathbf{V}(f-4)$, $\mathbf{V}(f-2)$, $\mathbf{V}(f)$, $\mathbf{V}(f+2)$, $\mathbf{V}(f+4)$.

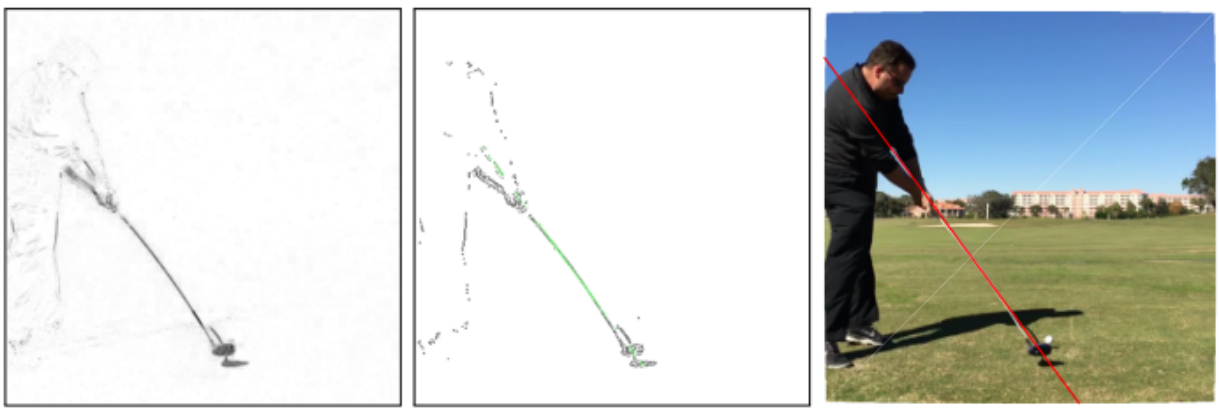
(b) Background image $\mathbf{B}(f)$.

Figure 4.4: Example of a computed background image $\mathbf{B}(f)$ and the video frames that have been used to compute it.

Method for finding the club. Basically use the PDF from 10!!! annotated images and use a 1-clique MRF to segment foreground and background. Doesn't seem very robust to me. PDF will be widely changing for edge case conditions and I think deep learning will be a more stable approach.

Table 4.8: Outline of the anchor point detection method.

1. Compute a difference image for the current frame.
2. Detect most dominant straight line in edges of difference image.
3. Find path with maximal sum of pixels along straight line.
4. Greedily search along straight line for path with high pixel sum.
5. Robustly fit quadratic to the two found paths.
6. Locate approximate center of club head.
7. Segment the club head.
8. Determine the anchor point by tracing the perimeter of the segmentation.



(a) Difference image (b) Edge pixels. Green are inliers (c) Detected line

Figure 4.10: Club shaft detection with RANSAC for straight lines.

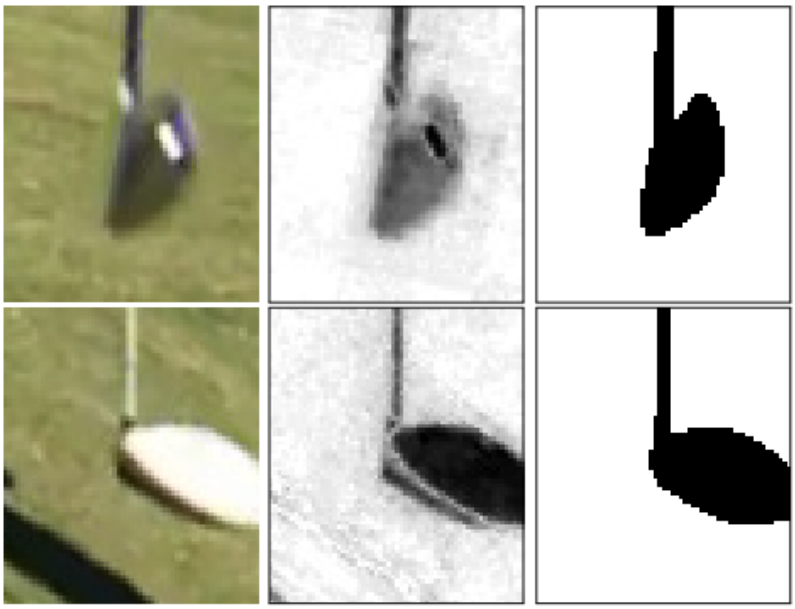
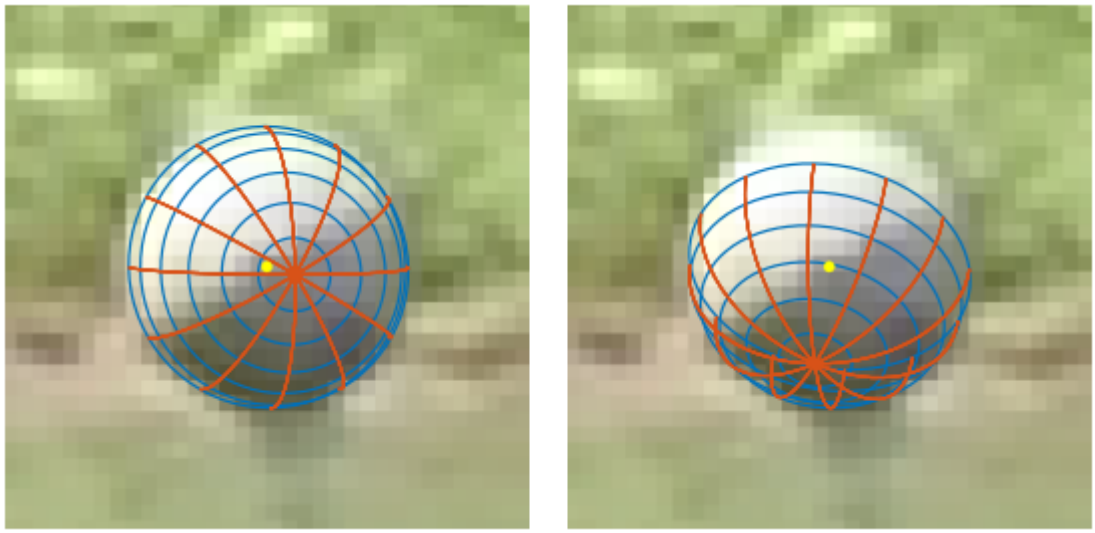


Figure 4.14: Examples of extracted ROI in color image, difference image and the annotated mask.

Super cool figure (not useful for this project though):



What to do next week

Work on basic detection model, literature search and enhanced project report

- [ ] Find best approach for one class detection network.
- [ ] Setup new computer
- [ ] Look into docker
- [ ] look into pix2pix to generate night training sets: https://github.com/phillipi/pix2pix

- [ ] Write introduction to the report.
- [ ] Figure out which computer to use for DL applications
- [ ] Figure out how to group the clubs into classification categories
- [ ] Script for extracting frames at specific points of the video