

Week 7+8

What has been done this week

I've tried to make a localization model based on the SVHN dataset. It doesn't work too well and especially not when trying to transfer it to the domain of the golf clubs. Previous monday I had to attend the burial of my of my step mothers urn and this monday I was sick. I have also started writing the report.*

- ☒ Create Annotated Dataset of golf clubs with the loca tion of the number
 - ☒ Annotated the test set (20%)
 - ☒ Modified and improved the Labelling tool to my need
- ☐ Report
 - ☐ Introduction
 - ☒ Short Introduction
 - ☐ Methodology
 - ☐ Data Chapter
 - ☒ Initial data section
 - ☒ annotation section
 - ☒ seller data section
- ☐ Make homemade localization model with the 4 output of the bounding box: Model:

```
FC_block_num = 2
FC_feature_dim = 256
dropout_ratio = 0.5
learning_rate = 0.0005
target_num = 4

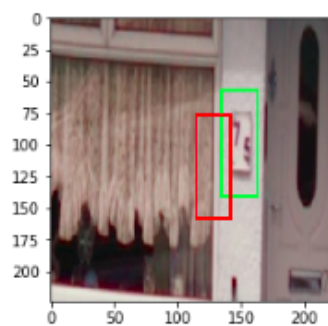
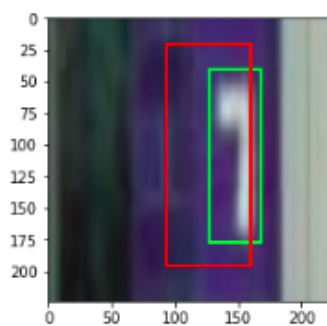
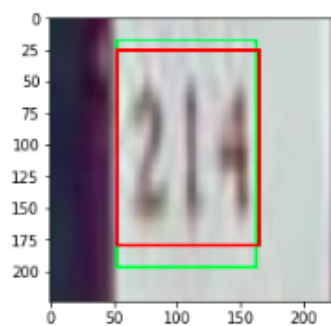
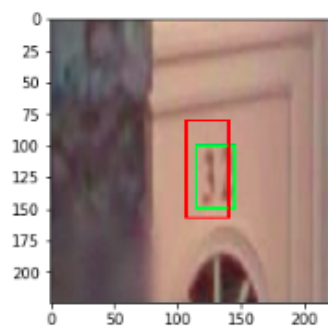
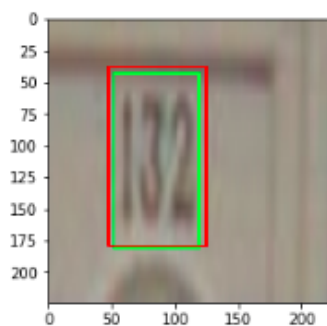
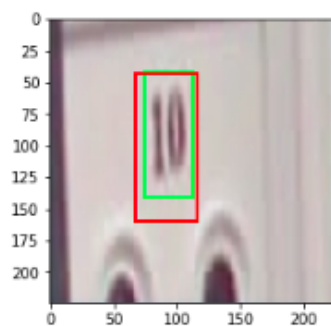
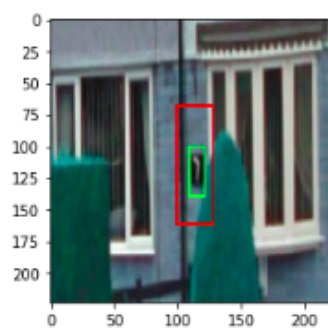
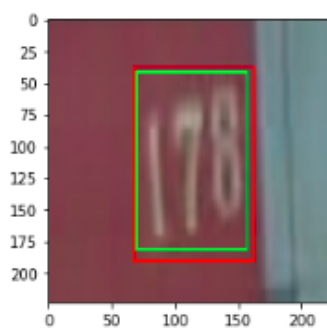
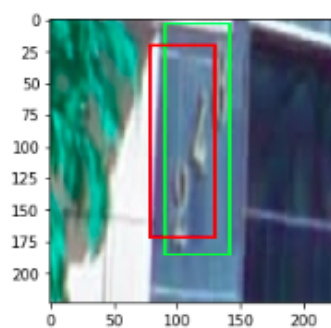
# build model
from classification_models import classification_models
ResNet18 = classification_models.ResNet18

pretrained_model = ResNet18(input_shape=(224,224,3), weights='imagenet', include_top=False)
for layer in pretrained_model.layers[:27]: #27 = stage2_unit1_bn1
    layer.trainable = False
output_tensor = pretrained_model.output

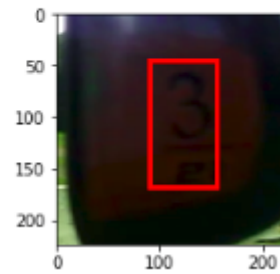
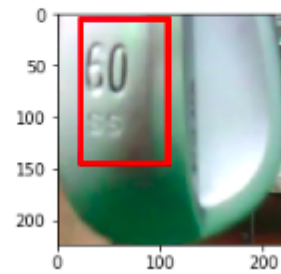
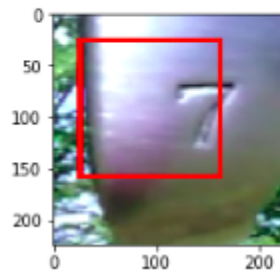
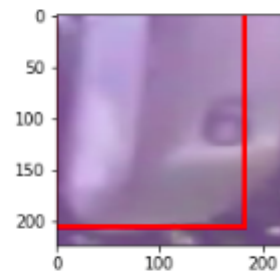
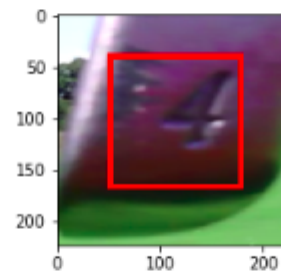
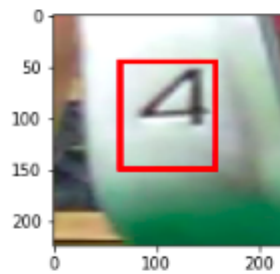
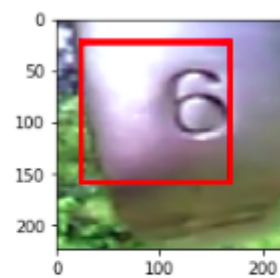
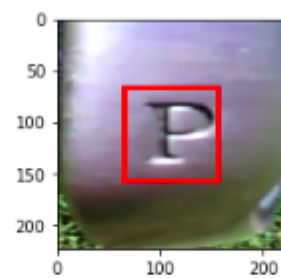
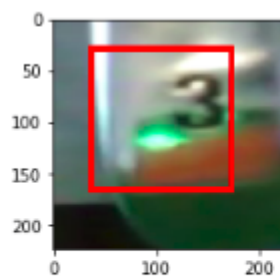
# FullyConnected blocks
output_tensor = Flatten()(output_tensor)
for _ in range(FC_block_num):
    output_tensor = Dense(FC_feature_dim, activation="relu")(output_tensor)
    output_tensor = BatchNormalization()(output_tensor)
    output_tensor = Dropout(dropout_ratio)(output_tensor)
output_tensor = Dense(target_num, name="out")(output_tensor)

model = keras.models.Model(pretrained_model.input, outputs=output_tensor)
model.compile(loss='mean_absolute_error', optimizer=Adam(lr=learning_rate), metrics=['mse', iou])
```

Wasn't really able to fully learn it:



Doesn't work that great on the golf data:



- ☑ Try to get STN-OCR (<https://github.com/Bartzi/stn-ocr>) to work to do localization of the club number
 - Wasn't able to get it up and running

Literature

STN-OCR: A single Neural Network for Text Detection and Text Recognition <https://arxiv.org/pdf/1707.08831.pdf>

- Looks like a pretty promising way to try and do localization + detection. Right now I do classification of number based on a model trained on the SVHN dataset, however that model requires a cropped photo of the number. Spatial Transformer Networks are quite nice, as they can learn an affine transformation matrix which will help localize the text. Pytorch has a nice guide on them: https://pytorch.org/tutorials/intermediate/spatial_transformer_tutorial.html

Status According to project plan

I need to revise my project plan and make a new one. Current one did not contain my desired steps with trying to do auto encoders etc.

Questions / Difficulties

For transformation: Which seems to be the best approach? I'm thinking training a model on the SVHN dataset and do fine tuning on my own. I will need some annotated data for that part. The **STN** networks could also be relevant and fun.

For embedding approach: I haven't tried throwing an auto encoder at the problem yet. It will require that I extract crops of a certain quality, since I don't expect it to be able to learn the data at all if it has to look at it in very varying configurations.

What to do next week

- Try to train an auto encoder on the cropped trainingset
- Try harder to get STN-OCR-ish approach to work.