TSBB11- User Guide

Football positioning

December 20, 2018

1 Introduction

This is a user guide for the project 'Football positioning' in the course TSBB11. The code can be cloned from *https://github.com/signalityai/3d-ball-trajectory-CDIO*. The user guide will list what software is needed and how to run the application.

2 Software requirements

This is the list of required software that must be installed in order to run the program. It is strongly recommended to install the packages using the Anaconda Distribution, which can be downloaded at *https://www.anaconda.com/download*.

Library	**Version**
Python	3.6.6
Numpy	1.15.4
Tensorflow	1.10.0
Keras	2.2.2
OpenCv	3.4.2
Scipy	1.10.0
Matplotlib	3.0.0

3 Data acquisition

In order to run the software, the program needs data that it can train on.

3.1 Trajectory data

Script *create_trajectories.py* generates data for trajectories, both a set containing trajectories with height zero and a set with trajectories that always leave the ground. Make sure to go to the project folder in the terminal and run:

python create_trajectories.py data/trajectories/my_data.json 100 -ground

This will create a file *my_data.json* that contains a data set with 100 ground, and non-ground trajectories. The name of the file can be changed. To acquire a data set with 100 non-ground trajectories, type:

python create_trajectories.py data/trajectories/my_data.json 100

If the script is saving to the same file several times, the new coordinates will be added to the already existing file.

3.2 Images

To be able to train a CNN, some images needs to be created and classified to use as training data. This is done in Unity, version 2018.2.7f1.

To create a data set containing images and corresponding coordinates you can download the zipped file of the Unity project from *https://bit.ly/2BC0Wop*. Open the project and enter game mode. When the game is started a json file will be created where the world and image coordinates will be logged with an ID that is shared with the corresponding image. The screenshots are taken automatically with a constant framerate and the images are saved in the folder ImageData.

To be able to create images and save the corresponding coordinates you just have to go into game-mode. When the "game" starts, screenshots will be taken with a constant frame-rate.

4 Training

The development of networks was done in Jupyter Notebook, which is a server-client application that allows training networks in a browser. The Jupyter Notebook is included when installing Anaconda. To run Jupyter, enter the following command in terminal:

jupyter notebook

This will open a new tab in the browser which shows the notebook interface. From this view, enter the folder that contains the cloned files.

The files created for training networks are listed below with a description of the file. All network architectures can be changed by changing the layers added to the model.

File name	**Description**
fully_connected_classification.ipynb	Fully connected network for classification trained on coordinates from trajectories.
fully_connected_regression.ipynb	Fully connected network for regression trained on coordinates from trajectories.
lstm.ipynb	LSTM network using coordinate sequences.
multiple_point_input.ipynb	Fully connected network using coordinate sequences.
cnn_classification.ipynb	CNN with classification output, trained on images
cnn_regression.ipynb	CNN with regression output, trained on images