Setup Guide for Identification of 3D Attention Points Project

By Sandeep Tandra, Vineel Gannu
April 17, 2020

This setup guide gives you clear instructions on how to execute the project code and all the requirements to run.

1 Resources Required

1.1 Softwares Required

- Unity
- Tobii eye tracking software
- Visual studio
- Anaconda package (All the libraries are included along with python)
- Jupyter lab

1.2 Hardware Required

- Tobii Eye Tracker 4C

2 Steps

2.1 Installation steps

- Download and install UnityHub here. Then install unity from the UnityHub.
- Download Tobii eye tracking software here.
- Download and install Visual studio here.
- Download and install Anaconda python distribution that is compatible with your system here.
- Download and install Jupyter lab by running **pip install jupyterlab** in Anaconda cmd prompt.

2.2 Creation of dataset

- Choose an environment based on your requirement in which you want to identify the attention points and build a virtual environment using unity software.
- Using Tobii eye tracker and Unity, create the dataset by displaying the virtual environment to a participant and capture the gaze positions of the participant and other features. A script has been written in unity that can capture the properties of an object and write the data to a text file.
- The above step can be repeated and required number of instances can be created.

2.3 Running the model

- Firstly, import the required libraries in Jupyter lab.
  - Pandas for operations on dataframes
  - Numpy for numerical operations
  - sklearn for the preprocessing, data splitting , metrics
  - matplotlib for data visualization

  - keras to build the model
  - pickle to serialize or save model to a file
  - and some math libraries
- All the versions of the libraries used are shown in the **versions.ipynb** in the setup files.
- The train and test dataset files are to be saved in the same location as that of .ipynb file.
- Read the csv file using pandas and store it as a dataframe.
- Now, run the cells in the .ipnb file successively to preprocess the data, visualize the data and run the model.
- Once the above cells are executed, the trained model is saved into a finalized_model.sav file so that you can evaluate the test data by running only evaluate function every time to predict the attention point(s).