

HOW TO RUN THE CODE

1. Requirements

Clone the repository

Install python 3.8

Install anaconda3

Install required libraries:

```
pip install opencv-contrib-python
```

```
pip install --upgrade keras
```

```
pip install ffmpeg
```

```
pip install numpy
```

```
pip install pandas
```

```
pip install scikit-learn
```

```
pip install matplotlib
```

```
pip install scipy
```

```
pip install seaborn
```

Install the anaconda environment. This can be done using the provided requirement file `requirements_anaconda.txt`. Run the following command.

```
conda create --name <env> --file requirements_anaconda.txt
```

2. Program execution

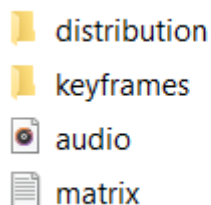
2.1. Segmentation of learning units

1. Run the segmentation tool by using the following command:

```
python main.py PATH_TO_RECORDINGS WHITEBOARD_PROF_IN_FULL_SCREEN
```

example: `python main.py data/records/ False`

2. Wait for the program execution.
3. Check the result in `PATH_TO_RECORDINGS`
For each analyzed video, a folder `RECORD_NAME` is created. Inside this folder following artifacts are created:



The folder “distribution” contains the final program output. This includes the learning units, the lecture note and the LMS interaction data.

The “keyframes” folder contains the keyframes of the video. These files are temporal artifacts. Also the files “audio” and “matrix” are temporal artifact.

Recommendersystem

4. Open the Jupyter programm
5. Open the Jupyter notebook *recommenderSystem_data_Preparation.ipynb*
6. Import the lecture note and interaction data in the notebook by adapting the following paths:

```
scenes_df = pd.read_csv('data/example/scenes.csv')
ratings_df = pd.read_csv('data/example/scenes_ratings.csv')
```

7. Run the notebook

A file `data_preprocessed.csv` will be created in the same location with the notebook.

8. Run the following notebooks to create recommendation of learning unit using different approaches.

recommenderSystem_itembased-collaborativefiltering,
recommenderSystem_userbased-collaborativefiltering
recommenderSystem_modelbased-collaborativefiltering.