**SocialMind AI Documentation**

Github: <https://github.com/CarmiShimon/scene_understanding>

Drive: <https://drive.google.com/drive/folders/1GNTGVH8ORTDgjv8ySUj5346fkJrM9Fsa?usp=share_link>

1. **Scene understanding pipeline**
   1. Analyze frame-by-frame and place results in a .csv file
   2. Get Audio results from <https://github.com/MarvinLvn/voice-type-classifier> (Dana’s code, .csv file) and combine with video results.
   3. Analyze combined video-audio results, find relevant modules, and create an output video.
   4. Create a short video with relevant occurrences and a statistics summary.

**Code Structure:**

**Scene\_understanding directory**

1. **Main.py** analyzes each frame (skeleton, gaze, child, adult, hands, mouth, and action)
   * + Input: ‘sub\_dir’ location of videos to be analyzed
     + Outputs: .csv file and a processed video file
     + Note – to save time, it currently processes each second frame and duplicates results of the last frame.

**AI\_Feedback directory**

1. **Main\_feedback.py** gets both audio and video csv results and finds reinforcement/no-reinforcment, communicative speech occurrences.
   * 1. Input: ‘sub\_dir\_name’ a directory of resulted video .csv and the location of an audio .csv results
     2. Outputs: merged audio and video results .csv, another .csv of binary classification of (hand looking, face looking, hand interaction, reinforcement, communicative speech) for every half second.
2. **Summarize\_video.py** gets the binary classification .csv and outputs a short video with relevant occurrences and a statistics summary.
   * 1. Input: ‘sub\_dir\_name’ (it goes to original video file and binary classification file)
     2. Output: .csv file and a short video (inside summary directory)
   1. **Video\_creation.py** gets the original video file, audio file and the binary classification .csv and outputs a full video file with annotation on what is happening between adult and child.
3. Activity (video understanding)
   1. Follow <https://github.com/open-mmlab/mmaction2> installation
   2. Use tutorial on adding new dataset (**videos annotation**): <https://github.com/open-mmlab/mmaction2/blob/master/docs/tutorials/3_new_dataset.md>
   3. Label videos using Aegisub software: <https://aegisub.en.uptodown.com/windows>
      * Pre-defined labels are in this link: <https://docs.google.com/document/d/1SWvmQTXhJrrP6cRsd5XK-6Ah7q2F7uA9W72SfWmh7jA/edit?usp=share_link>
      * Place each subtitle label file next to its video file.
   4. Data preparation:
      * After labeling using Aegisub, Run subtitles.py <https://github.com/CarmiShimon/scene_understanding/tree/main/activity_in_video>

This will run on videos and subtitles files, convert videos to .mp4 files and split videos to short videos according to labels that found in the subtitle annotation file.

* + - Run create\_data\_files.py <https://github.com/CarmiShimon/scene_understanding/tree/main/activity_in_video>

This will create train, val and test list of video files which suppose to be read by mmaction2 training procedure using video annotation.

* 1. Use relevant config file under config dir (inside mmaction2) to point your new (val, train, test) files so you could train the model.

## I used TPN network ([configs](https://github.com/open-mmlab/mmaction2/tree/master/configs)/[recognition](https://github.com/open-mmlab/mmaction2/tree/master/configs/recognition)/[tpn](https://github.com/open-mmlab/mmaction2/tree/master/configs/recognition/tpn)/tpn\_imagenet\_pretrained\_slowonly\_r50\_8x8x1\_150e\_kinetics\_rgb.py)

* + - Make sure your virtual environment compatible with mmaction2 and has cuda in it.