

SAE Viewport Prediction

This project predicts where a person will look next in a 360° video using a model called Spatiotemporal Attentive Entropy (SAE).

Initial Setup:

- Make a folder called “SAE Viewport Prediction Folder” on your desktop
- Open a terminal and cd into that folder

Setup Instructions:

1. Clone the repository
 - git clone <https://github.com/IamArmanNikkhah/sae-viewport-prediction.git>
 - cd sae-viewport-prediction
 - git checkout -b feature/week-5-training-pipeline
origin/feature/week-5-training-pipeline → This will ensure that you are in the branch that has all the code and files needed to successfully run the pipeline
 - git branch

Create a virtual environment:

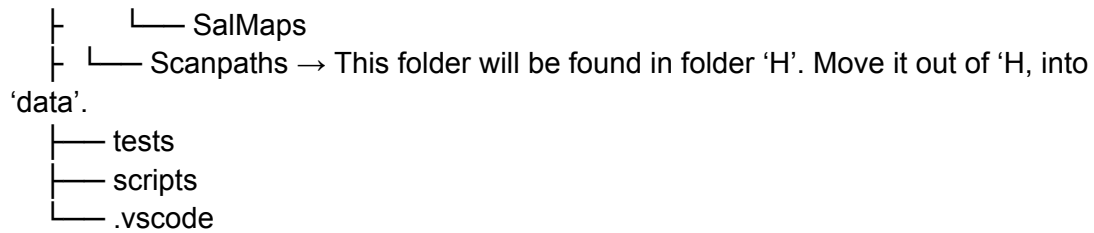
1. Mac/Linux:
 - python3 -m venv venv
 - source venv/bin/activate
2. Windows (PowerShell):
 - python -m venv venv
 - .\venv\Scripts\activate.ps1 → If this is not working, ensure that you don't have any PowerShell restrictions on.

Install dependencies:

- pip install -r requirements.txt

Download the Dataset:

- <https://zenodo.org/records/10650505>
- Download Image_H.7z manually from the Zenodo page. → Note: This folder might take around 10+ minutes to download, depending on computer.
- While waiting for the images to download, create a folder called 'data' inside the 'sae-viewport-prediction' folder.
- Extract the data using 7-Zip or any unzip tool.
- Move the extracted folder into the 'data' folder you just created.
- Your data folder should now look like this:
- SAE Viewport Prediction Folder
 - └─ sae-viewport-prediction/
 - └─ src
 - └─ data
 - └─ H → Extracted folder from Image_H.7z



Run the Code → Train the Model

1. Mac/Linux/Windows:

- python scripts/process_all_scanpaths.py
- python scripts/make_npy_from_cleaned.py
- python scripts/generate_vmf_lut.py
- Your data folder should now look like this:
- SAE Viewport Prediction Folder
 - └─ sae-viewport-prediction
 - └─ src
 - └─ data
 - └─ H
 - └─ SalMaps
 - └─ Processed
 - └─ Scanpaths
 - └─ tests
 - └─ scripts
 - └─ .vscode
- python -m src.training.train → This will start the training process using the datasets.

Evaluate the Model

1. Mac/Linux/Windows:

- python -m src.evaluation.evaluate → This will run the evaluation script and print the MAE and NLL metrics.