

Eye movement classification

NB: Det er bare Birk Øvstetun Narvhus som har jobbet på prosjektet, men ettersom deling av maskin på labb, er det opplastninger fra flere brukere.

Dette repositoryet inneholder kode for å klassifisere øyebevegelser. Dette innebærer også en rekke tester for å evaluere modellene. Modellene i modell 3-5 er trent på OpenEDS datasettet, mens modellene i ClustermetoderTesting er trent på MNIST datasettet. OpenEDS datasettet er ikke inkludert i dette repositoryet. Datasettet må lastes ned manuelt og legges i **data** mappen. Alle 3D modell scripts burde kjøres på Nvidia RTX 4090.

Nødvendige biblioteker

Modellene er bygd med de følgende bibliotekene:

- torch
- torchvision
- numpy
- pandas
- matplotlib
- sklearn
- tqdm

2D test modeller

Trening av modeller

Modeller av iterasjon 1. i rapporten ligger i mappen ClustermetoderTesting. De bruker alle datasettet MNIST.

Trene Kontrastiv modell 1:

```
python ./ClustermethodTesting/ClustermethodTesting/mnist_cnn_clr.py
```

Trene Autoencoder modell 1:

```
python ./ClustermethodTesting/ClustermethodTesting/Train_auto_encoder.py
```

Testing av modellene gjøres ved en rekke test scripts:

Klynger enkoder resultat fra autoencoder lastet fra **<checkpoint_path>**:

```
python ./ClustermethodTesting/ClusterEncodedData.py <checkpoint_path>
```

Klassifiserer enkoder resultat fra autoencoder lastet fra **<checkpoint_path>** med SVM Bruk **<model_type>** for å velge hvilken modell som skal brukes (**autoencoder/simclr**):

```
python ./ClustermethodTesting/Encode_svm.py <checkpoint_path> <optional: model_type>
```

Viser bilder fra auto enkoder lastet fra **<checkpoint_path>**:

```
python ./ClustermethodTesting/showEncoderImages.py <checkpoint_path>
```

Trener og tester en downstream finetuner model fra simclr resultat lastet fra **<checkpoint_path>**:

```
python ./ClustermethodTesting/sim_clr_ds_finetuner.py <checkpoint_path>
```

Trener og tester en downstream finetuner model fra enkoderen i autoencoder resultat lastet fra **<checkpoint_path>**:

```
python ./ClustermethodTesting/Testing_fine_tuned_model.py <checkpoint_path>
```

3D CNN modeller

underveis test moduler

I mappen **models/pre_tests** ligger det flere moduler som ble brukt under utarbeidingen av modell 2 og modell 1. Modulene ble ikke brukt i endelig implementasjon.

Trening av modeller

Parameter konfigurasjoner ligger i **configs/modelConfig.yaml**. Config filen inneholder parametere for modell 5. **modelConfig_model4.yaml** inneholder configs for model 4. Hvis man skal trene model 4, må man rename filen til **modelConfig.yaml**. Modell arkitekturer er lagert i **content/arc** mappen.

Bruk:

Trening av model 3-5 skjer via denne kommandoen. **<optional: model_file>** er en fil som inneholder en modell som kan lastes inn og trenes videre:

```
python ./models/finalImplementations/autoEncoderTrainer.py <optional: model_file>
```

Trene simClr modellen. Denne bruker ikke configs, siden metoden ikke fungerte under testing:

```
python ./models/finalImplementations/simClrTrainer.py
```

Testing av modellene gjøres ved en rekke test scripts:

Hoved test scriptet for modell 3-5. **<model_file>** er en fil som inneholder en modell som skal testes: **<optional: mode>**. **save** for å lagre video av testen, **kmeans** for å kjøre kmeans clustering på test data og **svm** for å kjøre svm på test data.

```
python ./util/testingUtils/test_model.py <model_file> <optional: mode>
```

Prosjekt struktur

- **models** - inneholder modellene som er brukt i prosjektet.
 - **finalImplementations** - inneholder komponenter som er brukt i modell 3-5.
 - **pre_tests** - inneholder moduler som ble brukt under utarbeidingen av modell 2 og modell 1.
- **data** - inneholder datasettet som er brukt i prosjektet.
- **util** - inneholder hjelpefunksjoner som er brukt i prosjektet.
 - **testingUtils** - inneholder tester for modellene.
 - **dataUtils** - inneholder funksjoner for å laste inn data.
 - **ivtUtils** - inneholder funksjoner for IVVT algoritmen og puppille uthening.
- **configs** - inneholder konfigurasjonsfiler for modellene.
- **ClustermethodTesting** - inneholder modeller og tester for iterasjon 1.
- **content** - inneholder content fra modellene.
 - **arc** - inneholder arkitekturer for modellene.
 - **saved_models** - inneholder lagrede modeller og checkpoints.
 - **saved_outputs** - inneholder lagrede output fra modellene og tester.