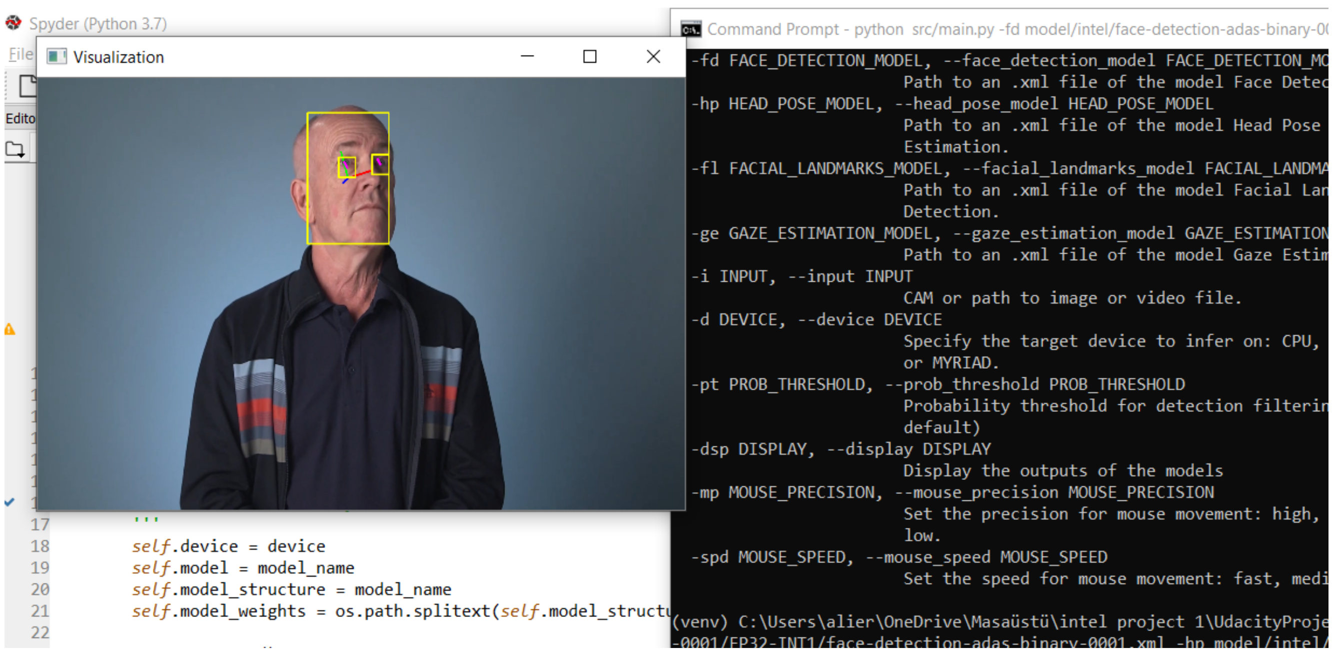
Ali Eren

Computer Pointer Controller

TODO: Write a short introduction to your project



This project is about an application that uses a gaze detection model to control the mouse pointer of a computer with the user's head pose and eyes using an input video or a live webcam stream.

Gaze Estimation model is used to follow the gaze of the user's head pose and eyes and change the position of the pointer according to the movement of these.

This project can be run in the same machine using different and multiple models, and the performance can be compared between these models.

This project is using networks from these OpenVINO pretrained models:

- Face Detection Model

- Head Pose Estimation Model

- Facial Landmarks Detection Model

- Gaze Estimation Model

Project Set Up and Installation

TODO: Explain the setup procedures to run your project. For instance, this can include your project directory structure, the models you need to download and where to place them etc. Also include details about how to install the dependencies your project requires.

Operation System: Windows 10 Pro

Configuration: Intel Core i7-8550 CPU @ 1.80 GHz

Python: v3.6

OpenVINO: v2020.3

Device: CPU

Step 1:

OpenVINO Toolkit was installed and run in the computer.

Command Prompt was opened and setupvars.bat batch file was entered in the command to set the environment variables which OpenVINO bin folder included.

Step 2:

Virtual environment was created: virtualenv venv

Virtual environment was activated: venv\Scripts\activate

Project dependencies were installed in the project directory: pip install requirements.txt

Step 3:

OpenVINO pretrained models were downloaded using OpenVINO model downloader script:

cd C:\Program Files (x86)\IntelSWTools\openvino\deployment_tools\tools\model_downloader\intel

python downloader.py --name face-detection-adas-binary-0001

python downloader.py --name landmarks-regression-retail-0009

python downloader.py --name head-pose-estimation-adas-0001

python downloader.py --name gaze-estimation-adas-0002

These downloaded models were copied to the project directory under the \model folder.

Demo

TODO: Explain how to run a basic demo of your model.

First, project directory was entered in the command line:

cd <project path>

CPU – FP32:

python src/main.py -fd model/intel/face-detection-adas-binary-0001/FP32-INT1/face-detection-adas-binary-0001.xml -hp model/intel/head-pose-estimation-adas-0001/FP32/head-pose-estimation-adas-0001.xml -fl model/intel/landmarks-regression-retail-0009/FP32/landmarks-regression-retail-0009.xml -ge model/intel/gaze-estimation-adas-0002/FP32/gaze-estimation-adas-0002.xml -i bin/demo.mp4

CPU – FP16:

python src/main.py -fd model/intel/face-detection-adas-binary-0001/FP32-INT1/face-detection-adas-binary-0001.xml -hp model/intel/head-pose-estimation-adas-0001/FP16/head-pose-estimation-adas-0001.xml -fl model/intel/landmarks-regression-retail-0009/FP16/landmarks-regression-retail-0009.xml -ge model/intel/gaze-estimation-adas-0002/FP16/gaze-estimation-adas-0002.xml -i bin/demo.mp4

CPU – FP32-INT8:

python src/main.py -fd model/intel/face-detection-adas-binary-0001/FP32-INT1/face-detection-adas-binary-0001.xml -hp model/intel/head-pose-estimation-adas-0001/FP32-INT8/head-pose-estimation-adas-0001.xml -fl model/intel/landmarks-regression-retail-0009/FP32-INT8/landmarks-regression-retail-0009.xml -ge model/intel/gaze-estimation-adas-0002/FP32-INT8/gaze-estimation-adas-0002.xml -i bin/demo.mp4

GPU – FP32:

python src/main.py -fd model/intel/face-detection-adas-binary-0001/FP32-INT1/face-detection-adas-binary-0001.xml -hp model/intel/head-pose-estimation-adas-0001/FP32/head-pose-estimation-adas-0001.xml -fl model/intel/landmarks-regression-retail-0009/FP32/landmarks-regression-retail-0009.xml -ge model/intel/gaze-estimation-adas-0002/FP32/gaze-estimation-adas-0002.xml -i bin/demo.mp4 -d GPU

GPU – FP16:

python src/main.py -fd model/intel/face-detection-adas-binary-0001/FP32-INT1/face-detection-adas-binary-0001.xml -hp model/intel/head-pose-estimation-adas-0001/FP16/head-pose-estimation-adas-0001.xml -fl model/intel/landmarks-regression-retail-0009/FP16/landmarks-regression-retail-0009.xml -ge model/intel/gaze-estimation-adas-0002/FP16/gaze-estimation-adas-0002.xml -i bin/demo.mp4 -d GPU

GPU – FP32-INT8:

python src/main.py -fd model/intel/face-detection-adas-binary-0001/FP32-INT1/face-detection-adas-binary-0001.xml -hp model/intel/head-pose-estimation-adas-0001/FP32-INT8/head-pose-estimation-adas-0001.xml -fl model/intel/landmarks-regression-retail-0009/FP32-INT8/landmarks-regression-retail-0009.xml -ge model/intel/gaze-estimation-adas-0002/FP32-INT8/gaze-estimation-adas-0002.xml -i bin/demo.mp4 -d GPU

Documentation

TODO: Include any documentation that users might need to better understand your project code. For instance, this is a good place to explain the command line arguments that your project supports.

Tree of the project files:

```
│   .Instructions.md.swp
│   README.md
│   requirements.txt
│
├───bin
│       .gitkeep
│       demo.mp4
│
├───model
│   └───intel
│       ├───face-detection-adas-binary-0001
│       │   └───FP32-INT1
│       │           face-detection-adas-binary-0001.bin
│       │           face-detection-adas-binary-0001.xml
│       │
│       ├───gaze-estimation-adas-0002
│       │   ├───FP16
│       │   │       gaze-estimation-adas-0002.bin
│       │   │       gaze-estimation-adas-0002.xml
│       │   │
│       │   ├───FP32
│       │   │       gaze-estimation-adas-0002.bin
│       │   │       gaze-estimation-adas-0002.xml
│       │   │
│       │   └───FP32-INT8
│       │           gaze-estimation-adas-0002.bin
│       │           gaze-estimation-adas-0002.xml
│       │
│       ├───head-pose-estimation-adas-0001
│       │   ├───FP16
│       │   │       head-pose-estimation-adas-0001.bin
│       │   │       head-pose-estimation-adas-0001.xml
│       │   │
│       │   ├───FP32
│       │   │       head-pose-estimation-adas-0001.bin
│       │   │       head-pose-estimation-adas-0001.xml
│       │   │
│       │   └───FP32-INT8
│       │           head-pose-estimation-adas-0001.bin
│       │           head-pose-estimation-adas-0001.xml
│       │
│       └───landmarks-regression-retail-0009
│           ├───FP16
│           │       landmarks-regression-retail-0009.bin
│           │       landmarks-regression-retail-0009.xml
│           │
│           ├───FP32
│           │       landmarks-regression-retail-0009.bin
│           │       landmarks-regression-retail-0009.xml
│           │
│           └───FP32-INT8
│                   landmarks-regression-retail-0009.bin
│                   landmarks-regression-retail-0009.xml
│
└───src
    │   face_detection.py
    │   facial_landmarks_detection.py
    │   gaze_estimation.py
    │   head_pose_estimation.py
    │   input_feeder.py
    │   main.py
    │   mouse_controller.py
    │
    └───__pycache__
            face_detection.cpython-36.pyc
            facial_landmarks_detection.cpython-36.pyc
            gaze_estimation.cpython-36.pyc
            head_pose_estimation.cpython-36.pyc
            input_feeder.cpython-36.pyc
            mouse_controller.cpython-36.pyc
```

main.py: Main project application code file

face_detection.py: Face detection prediction file

facial_landmarks_detection.py: Face landmarks detection prediction file

gaze_estimation.py: Gaze estimation prediction file

head_pose_estimation.py: Head pose estimation prediction file

input_feeder.py: Input video stream processing file

mouse_controller.py: Mouse movement controller file based on the output

```
usage: main.py [-h] -fd FACE_DETECTION_MODEL -hp HEAD_POSE_MODEL -fl
       FACIAL_LANDMARKS_MODEL -ge GAZE_ESTIMATION_MODEL -i INPUT
       [-d DEVICE] [-pt PROB_THRESHOLD] [-dsp DISPLAY]
       [-mp MOUSE_PRECISION] [-spd MOUSE_SPEED]
```

Required parameter details can be called by entering command line:

```
python3 src/main.py -h
```

```
usage: main.py [-h] -fd FACE_DETECTION_MODEL -hp HEAD_POSE_MODEL -fl
               FACIAL_LANDMARKS_MODEL -ge GAZE_ESTIMATION_MODEL -i INPUT
               [-d DEVICE] [-pt PROB_THRESHOLD] [-dsp DISPLAY]
               [-mp MOUSE_PRECISION] [-spd MOUSE_SPEED]

optional arguments:
  -h, --help            show this help message and exit
  -fd FACE_DETECTION_MODEL, --face_detection_model FACE_DETECTION_MODEL
                        Path to an .xml file of the model Face Detection.
  -hp HEAD_POSE_MODEL, --head_pose_model HEAD_POSE_MODEL
                        Path to an .xml file of the model Head Pose
                        Estimation.
  -fl FACIAL_LANDMARKS_MODEL, --facial_landmarks_model FACIAL_LANDMARKS_MODEL
                        Path to an .xml file of the model Facial Landmarks
                        Detection.
  -ge GAZE_ESTIMATION_MODEL, --gaze_estimation_model GAZE_ESTIMATION_MODEL
                        Path to an .xml file of the model Gaze Estimation.
  -i INPUT, --input INPUT
                        CAM or path to image or video file.
  -d DEVICE, --device DEVICE
                        Specify the target device to infer on: CPU, GPU, FPGA
                        or MYRIAD.
  -pt PROB_THRESHOLD, --prob_threshold PROB_THRESHOLD
                        Probability threshold for detection filtering(0.6 by
                        default)
  -dsp DISPLAY, --display DISPLAY
                        Display the outputs of the models
  -mp MOUSE_PRECISION, --mouse_precision MOUSE_PRECISION
                        Set the precision for mouse movement: high, medium,
                        low.
  -spd MOUSE_SPEED, --mouse_speed MOUSE_SPEED
                        Set the speed for mouse movement: fast, medium, slow.
```

Benchmarks

TODO: Include the benchmark results of running your model on multiple hardwares and multiple model precisions. Your benchmarks can include: model loading time, input/output processing time, model inference time etc.

Configuration	Loading Time	Inference Time
CPU – FP32	1.0317049026489258	96.82043957710266
CPU – FP16	1.119312047958374	96.75655817985535
CPU – FP32-INT8	3.1636881828308105	96.66419768333435
GPU – FP32	81.63898611068726	94.2278106212616
GPU – FP16	82.85934448242188	93.74738836288452
GPU – FP32-INT8	95.18037939071655	94.14373850822449

Results

TODO: Discuss the benchmark results and explain why you are getting the results you are getting. For instance, explain why there is difference in inference time for FP32, FP16 and INT8 models.

FP32 is a single-precision floating point format. CPUs and GPUs can run 32-bit floating point operations efficiently.

FP16 is a half-precision floating point format, which uses the half of the bits that FP32 uses. FP16 has a lower precision level, however it can still perform inference tasks successfully. FP16 requires less space and time than FP32.

INT8 is an 8-bit integer data type. INT8 data is better on performing calculations than floating point data format, however the range is smaller than FP16 or FP32. INT8 precision can decrease latency and increase throughput on some occasions, but sure it causes a loss of accuracy in the model performance. INT8 precision should be used if needed for the speed not accuracy.

In this project, FP32, FP16 and INT8 precisions provided similar inference time, but INT8 precision provided longer loading time. For decreasing the required memory, INT8 would be a good choice despite longer loading time.

However, when using GPU, loading times were increased highly, but inference times were decreased a little bit. GPU can be considered if inference time is important for the project.