Face Detection API

`This repository includes the Face Detection API. Models used for Face Recognition are MTCNN (Multi-task Cascade Convolutional Neural Network) and OpenCV haar cascades. FastApi is used on the API side.`

Project Folder

- **`env/`**
  - This directory includes env files
- **`img/`**
  - This directory includes test image files
- **`models/`**
  - opencv
    - haarcascades
      - haarcascade_frontalface_default.xml
- **`src/`**
  - models.py
- .dockerignore
- .gitignore
- Dockerfile
- main.py
- ReadMe.md
- ReadMe.pdf

Install

1- Install in an Actual Env.

```
pip install env/requirements.txt
```

2- Install with a New Env. (Optional)

```
conda env create -f env/env.yaml
conda activate ENV_NAME
```

Run

1- Run on Local System

```
python main.py
```

2- Run on with Docker System

- **Step 1:** Docker Image Build

  ```
  docker build -t face_detection_api .
  ```

- **Step 1 Control:**

```
docker image ls
or
docker images
```

- **Step 2:** Create and Run Container from Docker Image

  ```
  docker run -d -p 8000:8000 face_detection_api
  or
  docker run --name face_detection_api_c -d -p 8000:8000 face_detection_api
  ```

- **Step 2.1 Control:**

  ```
  docker ps
  or
  docker ps -a
  ```

- **Step 2.2 Control:**

  ```
  docker logs <CONTAINER ID OR CONTAINER NAME>
  ```

- **Step 2.3 Control:**

  ```
  docker container ls -a
  docker rm <CONTAINER ID OR CONTAINER NAME>
  docker rmi face_detection_api
  ```

  Check Web Browser:

- `http://127.0.0.1:8000` or `http://localhost:8000`
- `http://127.0.0.1:8000/docs`
- `http://127.0.0.1:8000/redoc`

FaceDetection Service Test:

Step 1: Let's go to `http://127.0.0.1:8000/docs` on the browser and load the test image from the **img/** directory and test it.
Step 2: Please browse to files from the env TestScreen1.jpg and TestScreen2.jpg.
Step 3: Please browse to the response results.

Additional information

Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks (MTCCN) | https://arxiv.org/abs/1604.02878

ToDo

- [ ] Adult Detection
- [ ] Logging
- [ ] facenet-pytorch Implementation

Contact

- Metin Uslu