

A facial keypoint detection system that takes in any image with faces, recognizes and detects faces, and predicts the location of 68 distinguishing keypoints on each face

4 commits

1 branch

0 packages

1 release

2 contributors

Branch: master

New pull request

Create new file

Upload files

Find file


Clone or download

Michael-Tu Merge pull request #2 from Michael-Tu/imgbot Latest commit 8546104 14 days ago

detector_architectures	migrate project into individual project	9 months ago
images	[ImgBot] Optimize images	14 days ago
research	migrate project into individual project	9 months ago
saved_models	migrate project into individual project	9 months ago
.gitignore	migrate project into individual project	9 months ago
1. Load and Visualize Data.ipynb	migrate project into individual project	9 months ago
2. Define the Network Architecture.ipynb	migrate project into individual project	9 months ago
3. Facial Keypoint Detection, Complete Pipeline.ipynb	migrate project into individual project	9 months ago
4. Fun with Keypoints.ipynb	migrate project into individual project	9 months ago
5. Zip Your Project Files and Submit.ipynb	migrate project into individual project	9 months ago
README.md	remove download code link	9 months ago
data_load.py	migrate project into individual project	9 months ago
filelist.txt	migrate project into individual project	9 months ago
models.py	migrate project into individual project	9 months ago
project1.zip	migrate project into individual project	9 months ago
workspace_utils.py	migrate project into individual project	9 months ago

README.md

Udacity CV ND



Facial Keypoint Detection

Project Objective

In this project, I'll build a facial keypoint detection system that takes in any image with faces, recognizes and detects faces, and predicts the location of 68 distinguishing keypoints on each face!

Background:

Facial keypoints include points around the eyes, nose, and mouth on a face and are used in many applications. These applications include: facial tracking, facial pose recognition, facial filters, and emotion recognition.

Getting the Files

Data

All of the data you'll need to train a neural network should be placed in the subdirectory `data` . To get the data, run the following commands in your terminal:

```
mkdir data
```

```
wget -P data/ https://s3.amazonaws.com/video.udacity-data.com/topher/2018/May/5aea1b91_train-test-data/train-test-data.zip
```

```
unzip -n data/train-test-data.zip -d data
```

Model Download

You can use my pre-trained model for your own experimentation. To use it, [download](#) the model and placed in the subdirectory `saved_models` .

Result

Here are some visualizations of the facial recognition, keypoints detection, CNN feature maps, and interesting sunglasses layover applications:



Dependencies

Before you can experiment with the code, you'll have to make sure that you have all the libraries and dependencies required to support this project. You will mainly need Python 3, PyTorch and its torchvision, OpenCV, Matplotlib, and tqdm.