using convolutional neural networks to classify facial expressions

doc	add paper as pdf	last year
model	remove old trained model	last year
notebooks	revising the paper	last year
saved_models	5500 steps and probbility output	last year
util	csv header for pandas	last year
.gitignore	add paper as pdf	last year
README.md	Update README.md	last year
model.py	prediction probability output, remove argmax	last year
predict.py	prediction probability output, remove argmax	last year
requirements.txt	add requirements	last year
train_ckplus.py	less steps	last year
train_fer.py	rename	last year
train_ferplus.py	training with probability distribution across labels	last year
webcam.py	top 3 emotions and bar visualization for probas	last year

README.md

Facial Expression Recognition

a convolutional neural network for recognizing facial expressions on live webcam images.

Installation

The implementation has been tested with Python 3.6.3. You can create a fresh virtual environment with conda or virtualenv if you like. TensorFlow is not supporting `conda` officially, so `pip` is used for package management. All the dependencies can be found in the `requirements.txt` file.

With your Python 3 environment activated, you can install the requirements with

```
pip install -r path/to/requirements.txt
```

Live prediction

If your computer has a webcam, you can compute predictions on the fly. Take off your glasses and hats and start the live-prediction with

```
python webcam.py
```

Training

If you would like to train the tensorflow CNN yourself, you need to obtain the FER2013 dataset from kaggle and, optionally, the CK+ dataset. For CK+, you can use the `ckplus_to_csv.py` script to automatically detect all the faces, parse the grayscale intensities and collect all the CK-images into a single CSV file.

The directory structure should look like this:

```
project root
│   README.md
│   ...
└───data
│   │   fer2013.csv
│   │   ckplus.csv
│   │
│   └───ckplus
│       │   images
│       │   labels
```

You can invoke the (somewhat lengthy) training process with

```
python train_fer.py
```