

---

## face2face-demo

This is a pix2pix demo that learns from facial landmarks and translates this into a face. A webcam-enabled application is also provided that translates your face to the trained face in real-time.

### Getting Started

#### 1. Prepare Environment

```
1 # Clone this repo
2 git clone git@github.com:datitran/face2face-demo.git
3
4 # Create the conda environment from file (Mac OSX)
5 conda env create -f environment.yml
```

#### 2. Generate Training Data

```
1 python generate_train_data.py --file angela_merkel_speech.mp4 --num 400
   --landmark-model shape_predictor_68_face_landmarks.dat
```

Input:

- `file` is the name of the video file from which you want to create the data set.
- `num` is the number of train data to be created.
- `landmark-model` is the facial landmark model that is used to detect the landmarks. A pre-trained facial landmark model is provided here.

Output:

- Two folders `original` and `landmarks` will be created.

If you want to download my dataset, here is also the video file that I used and the generated training dataset (400 images already split into training and validation).

#### 3. Train Model

```
1 # Clone the repo from Christopher Hesse's pix2pix TensorFlow
   implementation
2 git clone https://github.com/affinelayer/pix2pix-tensorflow.git
3
4 # Move the original and landmarks folder into the pix2pix-tensorflow
   folder
5 mv face2face-demo/landmarks face2face-demo/original pix2pix-tensorflow/
   photos
6
7 # Go into the pix2pix-tensorflow folder
8 cd pix2pix-tensorflow/
9
10 # Resize original images
```

---

```
11 python tools/process.py \  
12 --input_dir photos/original \  
13 --operation resize \  
14 --output_dir photos/original_resized  
15  
16 # Resize landmark images  
17 python tools/process.py \  
18 --input_dir photos/landmarks \  
19 --operation resize \  
20 --output_dir photos/landmarks_resized  
21  
22 # Combine both resized original and landmark images  
23 python tools/process.py \  
24 --input_dir photos/landmarks_resized \  
25 --b_dir photos/original_resized \  
26 --operation combine \  
27 --output_dir photos/combined  
28  
29 # Split into train/val set  
30 python tools/split.py \  
31 --dir photos/combined  
32  
33 # Train the model on the data  
34 python pix2pix.py \  
35 --mode train \  
36 --output_dir face2face-model \  
37 --max_epochs 200 \  
38 --input_dir photos/combined/train \  
39 --which_direction AtoB
```

For more information around training, have a look at Christopher Hesse's [pix2pix-tensorflow](#) implementation.

#### 4. Export Model

1. First, we need to reduce the trained model so that we can use an image tensor as input:

```
python reduce_model.py --model-input face2face-model --model-  
output face2face-reduced-model
```

Input:

- `model-input` is the model folder to be imported.
- `model-output` is the model (reduced) folder to be exported.

Output:

- It returns a reduced model with less weights file size than the original model.

- 
2. Second, we freeze the reduced model to a single file. `python freeze_model.py --model-folder face2face-reduced-model`

Input:

- `model-folder` is the model folder of the reduced model.

Output:

- It returns a frozen model file `frozen_model.pb` in the model folder.

I have uploaded a pre-trained frozen model here. This model is trained on 400 images with epoch 200.

### 5. Run Demo

```
1 python run_webcam.py --source 0 --show 0 --landmark-model  
  shape_predictor_68_face_landmarks.dat --tf-model face2face-reduced-  
  model/frozen_model.pb
```

Input:

- `source` is the device index of the camera (default=0).
- `show` is an option to either display the normal input (0) or the facial landmark (1) alongside the generated image (default=0).
- `landmark-model` is the facial landmark model that is used to detect the landmarks.
- `tf-model` is the frozen model file.

Example:



---

## Requirements

- Anaconda / Python 3.5
- TensorFlow 1.2
- OpenCV 3.0
- Dlib 19.4

## Acknowledgments

Kudos to Christopher Hesse for his amazing pix2pix TensorFlow implementation and Gene Kogan for his inspirational workshop.

## Copyright

See LICENSE for details. Copyright (c) 2017 Dat Tran.