
Everybody Dance Now

[[website](#)] [[paper](#)] [[youtube](#)]



Implementation accompanying paper:

Everybody Dance Now

Caroline Chan, Shiry Ginosar, Tinghui Zhou, Alexei A. Efros

UC Berkeley

hosted on arXiv

Prerequisites

1. PyTorch
2. Python Library Dominate

```
1 pip install dominate
```

3. Clone this repository

```
1 git clone https://github.com/carolineec/EverybodyDanceNow
```

We ran our code on a 12GB NVIDIA GPU. Multi-GPU and CPU setups are currently untested.

Training

Global Stage We follow similar stage training as in pix2pixHD. We first train a “global” stage model at 512x256 resolution

```
1 # train a model at 512x256 resolution
2 python train_fullts.py \
3 --name MY_MODEL_NAME_global \
4 --dataroot MY_TRAINING_DATASET \
5 --checkpoints_dir WHERE_TO_SAVE_CHECKPOINTS \
6 --loadSize 512 \
7 --no_instance \
8 --no_flip \
9 --tf_log \
10 --label_nc 6
```

Local Stage Followed by a “local” stage model with 1024x512 resolution.

```
1 # train a model at 1024x512 resolution
2 python train_fullts.py \
3 --name MY_MODEL_NAME_local \
4 --dataroot MY_TRAINING_DATASET \
5 --checkpoints_dir WHERE_TO_SAVE_CHECKPOINTS \
6 --load_pretrain MY_MODEL_NAME_global \
7 --netG local \
8 --ngf 32 \
9 --num_D 3 \
10 --resize_or_crop none \
11 --no_instance \
12 --no_flip \
13 --tf_log \
14 --label_nc 6
```

Face GAN stage We then can apply another stage with a separate GAN focused on the face region.

```
1 # train a model specialized to the face region
2 python train_fullts.py \
3 --name MY_MODEL_NAME_face \
4 --dataroot MY_TRAINING_DATASET \
5 --load_pretrain MY_MODEL_NAME_local \
6 --checkpoints_dir WHERE_TO_SAVE_CHECKPOINTS \
7 --face_discrim \
8 --face_generator \
9 --faceGtype global \
10 --niter_fix_main 10 \
11 --netG local \
12 --ngf 32 \
13 --num_D 3 \
14 --resize_or_crop none \
15 --no_instance \
16 --no_flip \
17 --tf_log \
```

```
18 --label_nc 6
```

Testing

The full checkpoint will be loaded from `-checkpoints_dir/-name` (i.e. if flags: “`-name foo ... -checkpoints_dir bar`” are included, checkpoints will be loaded from `foo/bar`) Replace `-howmany` flag with an upper bound on how many test examples you have

Global Stage

```
1 # test model at 512x256 resolution
2 python test_fullts.py \
3 --name MY_MODEL_NAME_global \
4 --dataroot MY_TEST_DATASET \
5 --checkpoints_dir CHECKPOINT_FILE_LOCATION \
6 --results_dir WHERE_TO_SAVE_RESULTS \
7 --loadSize 512 \
8 --no_instance \
9 --how_many 10000 \
10 --label_nc 6
```

Local Stage

```
1 # test model at 1024x512 resolution
2 python test_fullts.py \
3 --name MY_MODEL_NAME_local \
4 --dataroot MY_TEST_DATASET \
5 --checkpoints_dir CHECKPOINT_FILE_LOCATION \
6 --results_dir WHERE_TO_SAVE_RESULTS \
7 --netG local \
8 --ngf 32 \
9 --resize_or_crop none \
10 --no_instance \
11 --how_many 10000 \
12 --label_nc 6
```

Face GAN stage

```
1 # test model at 1024x512 resolution with face GAN
2 python test_fullts.py \
3 --name MY_MODEL_NAME_face \
4 --dataroot MY_TEST_DATASET \
5 --checkpoints_dir CHECKPOINT_FILE_LOCATION \
6 --results_dir WHERE_TO_SAVE_RESULTS \
7 --face_generator \
8 --faceGtype global \
9 --netG local \
10 --ngf 32 \
11 --resize_or_crop none \
```

```
12 --no_instance \  
13 --how_many 10000 \  
14 --label_nc 6
```

Dataset preparation

We also provide code for creating both training and testing datasets (including global pose normalization) in the **data_prep** folder. See the **sample_data** folder for examples on how to prepare the code for training. Note the original_img is not necessary at test time and is provided only for reference.

Our dataset preparation code is based on output formats from OpenPose and currently supports the COCO, BODY_23, and BODY_25 pose output format as well as hand and face keypoints. To install and run OpenPose please follow the directions at the OpenPose repository.

graph_train.py

will prepare a train dataset with subfolders - train_label (contains 1024x512 inputs) - train_img (contains 1024x512 targets) - train_facetexts128 (contains face 128x128 bounding box coordinates in .txt files) No smoothing

```
1 python graph_train.py \  
2 --keypoints_dir /data/scratch/caroline/keypoints/jason_keys \  
3 --frames_dir /data/scratch/caroline/frames/jason_frames \  
4 --save_dir /data/scratch/caroline/savefolder \  
5 --spread 4000 25631 1 \  
6 --facetexts
```

graph_avesmooth.py

will prepare a dataset with averaged smoothed keypoints with subfolders (usually for validation) - test_label (contains 1024x512 inputs) - test_img (contains 1024x512 targets) - test_facetexts128 (contains face 128x128 bounding box coordinates in .txt files)

```
1 python graph_avesmooth.py \  
2 --keypoints_dir /data/scratch/caroline/keypoints/wholedance_keys \  
3 --frames_dir /data/scratch/caroline/frames/wholedance \  
4 --save_dir /data/scratch/caroline/savefolder \  
5 --spread 500 29999 4 \  
6 --facetexts
```

graph_posenorm.py

will prepare a dataset with global pose normalization + median smoothing - test_label (contains 1024x512 inputs) - test_img (contains 1024x512 targets) - test_facetexts128 (contains face 128x128 bounding box coordinates in .txt files)

```
1 python graph_posenorm.py \  
2 --target_keypoints /data/scratch/caroline/keypoints/wholedance_keys \  
3 --source_keypoints /data/scratch/caroline/keypoints/  
  dubstep_keypointsF00T \  
4 --target_shape 1080 1920 3 \  
5 --source_shape 1080 1920 3 \  
6 --source_frames /data/scratch/caroline/frames/dubstep_frames \  
7 --results /data/scratch/caroline/savefolder \  
8 --target_spread 30003 178780 \  
9 --source_spread 200 4800 \  
10 --calculate_scale_translation  
11 --facetexts
```

Citation

If you find this work useful please use the following citation:

```
1 @inproceedings{chan2019dance,  
2   title={Everybody Dance Now},  
3   author={Chan, Caroline and Ginosar, Shiry and Zhou, Tinghui and Efros,  
4     Alexei A},  
5   booktitle={IEEE International Conference on Computer Vision (ICCV)},  
6   year={2019}  
7 }
```

Acknowledgements

Model code adapted from pix2pixHD and pytorch-CycleGAN-and-pix2pix

Data Preparation code adapted from Realtime_Multi-Person_Pose_Estimation

Data Preparation code based on outputs from OpenPose