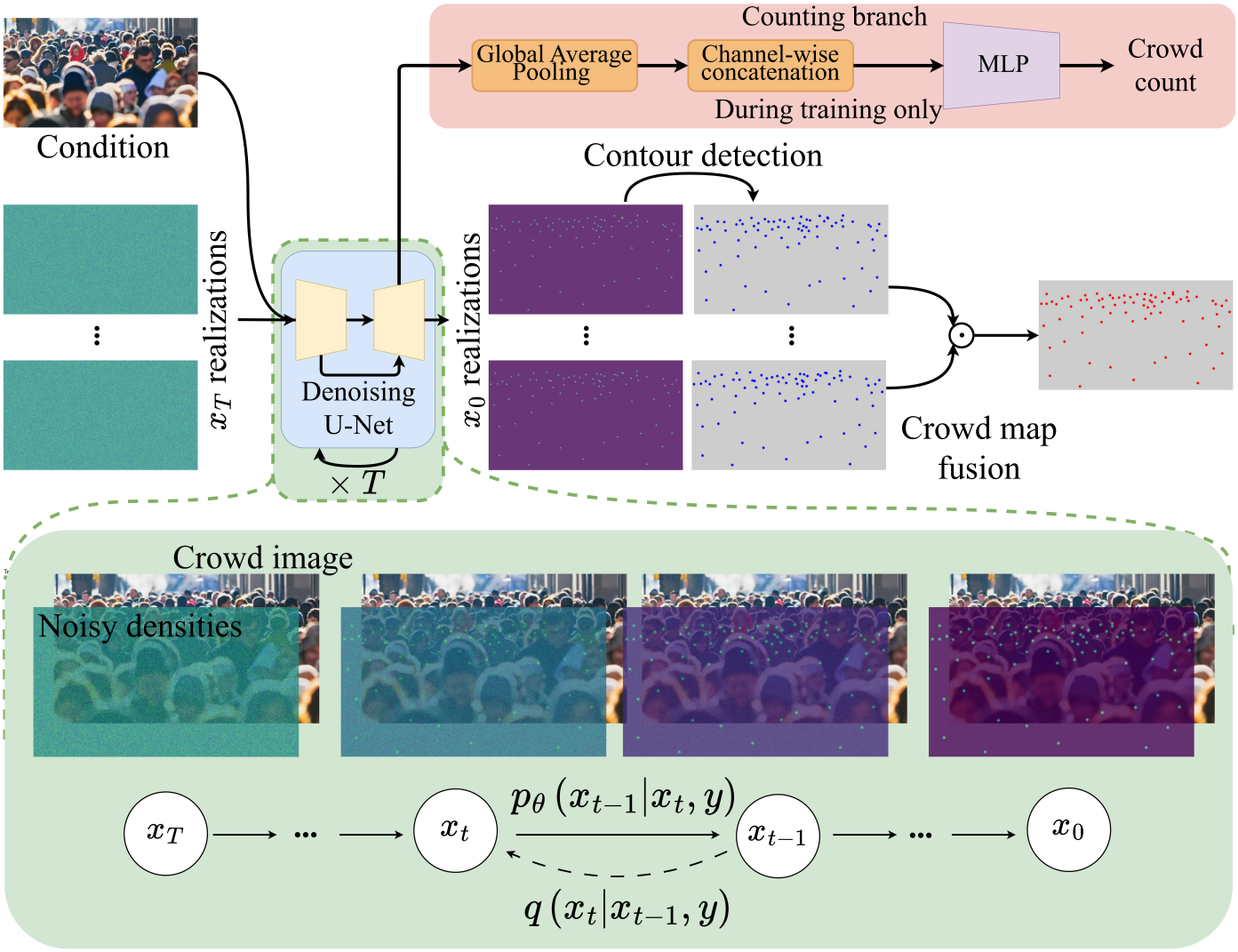
- CrowdDiff: Multi-hypothesis Crowd Density Estimation using Diffusion Models
  - Method
  - Visualized demos for density maps
  - Visualized demos for crowd maps and stochastic generation
  - Installing
  - Dataset preparation
  - Training
  - Testing
  - Acknowledgement:

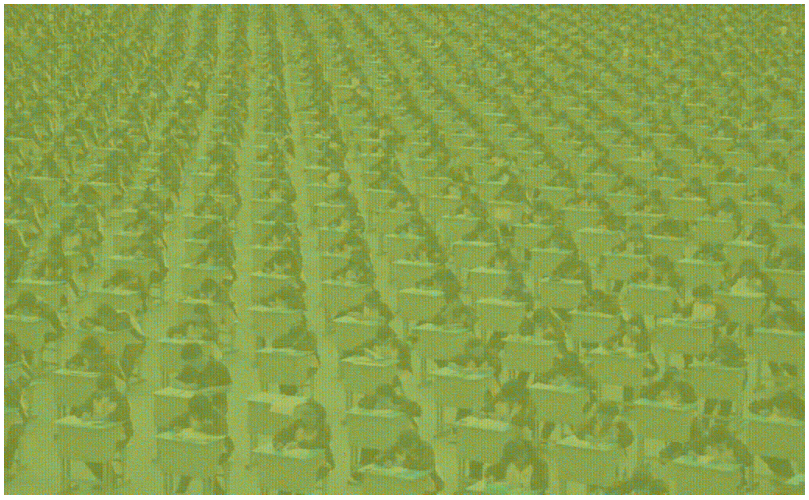
CrowdDiff: Multi-hypothesis Crowd Density Estimation using Diffusion Models

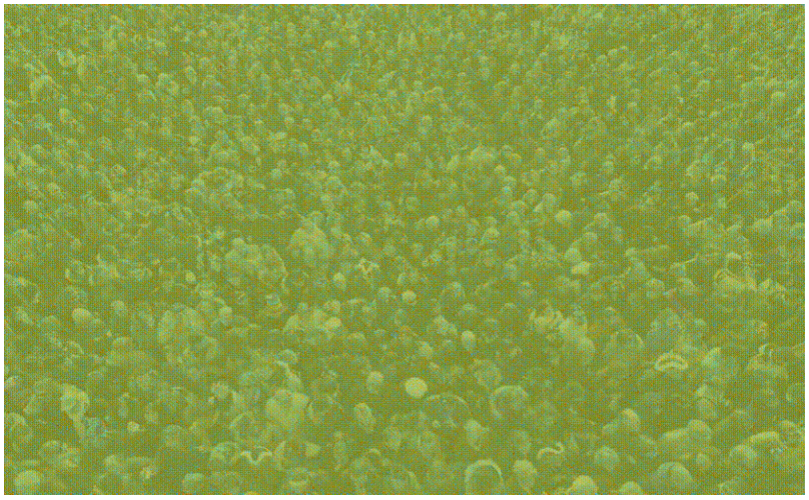
This repository contains the codes for the PyTorch implementation of the paper [Diffuse-Denoise-Count: Accurate Crowd Counting with Diffusion Models]

Method



Visualized demos for density maps





Visualized demos for crowd maps and stochastic generation

Ground Truth: 361

Trial 1: 349

Trial 2: 351

Final Prediction: 359

Trial 3: 356

Trial 4: 360

Installing

- Install python dependencies. We use python 3.9.7 and PyTorch 1.13.1.

```
pip install -r requirements.txt
```

Dataset preparation

- Run the preprocessing script.

```
python cc_utils/preprocess_shtech.py \
    --data_dir path/to/data \
    --output_dir path/to/save \
    --dataset dataset \
    --mode test \
    --image_size 256 \
    --ndevices 1 \
    --sigma '0.5' \
    --kernel_size '3' \
```

Training

- Download the pre-trained weights.
- Run the training script.

```
DATA_DIR="--data_dir path/to/train/data --val_samples_dir path/to/val/data"
LOG_DIR="--log_dir path/to/results --resume_checkpoint path/to/pre-
trained/weights"
TRAIN_FLAGS="--normalizer 0.8 --pred_channels 1 --batch_size 8 --
save_interval 10000 --lr 1e-4"
MODEL_FLAGS="--attention_resolutions 32,16,8 --class_cond False --
diffusion_steps 1000 --large_size 256  --small_size 256 --learn_sigma True -
-noise_schedule linear --num_channels 192 --num_head_channels 64 --
num_res_blocks 2 --resblock_updown True --use_fp16 True --
use_scale_shift_norm True"

CUDA_VISIBLE_DEVICES=0 python scripts/super_res_train.py $DATA_DIR $LOG_DIR
$TRAIN_FLAGS $MODEL_FLAGS
```

Testing

- Download the pre-trained weights.
- Run the testing script.

```
DATA_DIR="--data_dir path/to/test/data"
LOG_DIR="--log_dir path/to/results --model_path path/to/model"
TRAIN_FLAGS="--normalizer 0.8 --pred_channels 1 --batch_size 1 --per_samples
1"
MODEL_FLAGS="--attention_resolutions 32,16,8 --class_cond False --
diffusion_steps 1000 --large_size 256  --small_size 256 --learn_sigma True -
-noise_schedule linear --num_channels 192 --num_head_channels 64 --
num_res_blocks 2 --resblock_updown True --use_fp16 True --
use_scale_shift_norm True"

CUDA_VISIBLE_DEVICES=0 python scripts/super_res_sample.py $DATA_DIR $LOG_DIR
$TRAIN_FLAGS $MODEL_FLAGS
```

Acknowledgement:

Part of the codes are borrowed from guided-diffusion codebase.