

Edge-LBAM

Pytorch implementation of paper "Image Inpainting with Edge-guided Learnable Bidirectional Attention Maps"

Description

This paper is an extension of our previous work. In comparison to [LBAM](#) we utilize both the mask of holes

and predicted edge map for mask-updating, resulting in our Edge-LBAM method. Moreover, we introduce a multi-scale

edge completion network for effective prediction of coherent edges.

Prerequisites

- Python 3.6
- Pytorch =1.1.0
- CPU or NVIDIA GPU + Cuda + Cudnn

Training

To train the MECNet:

To train the Edge-LBAM model:

```
python train.py --batchSize numOf_batch_size --dataRoot your_image_path \  
--maskRoot your_mask_root --modelsSavePath path_to_save_your_model \  
--logPath path_to_save_tensorboard_log --pretrain(optional)  
pretrained_model_path
```

Testing

To test with random batch with random masks:

```
python test_random_batch.py --dataRoot your_image_path  
--maskRoot your_mask_path --batchSize numOf_batch_size --pretrain  
pretrained_model_path
```

Pretrained Models

The pretrained models can be found at [google drive](#), we will release the models removing bn from Edge-LBAM later which may effect better. You can also train the model by yourself.

Results

Inpainting



Input

Global&Local

PConv

DeepFillv2



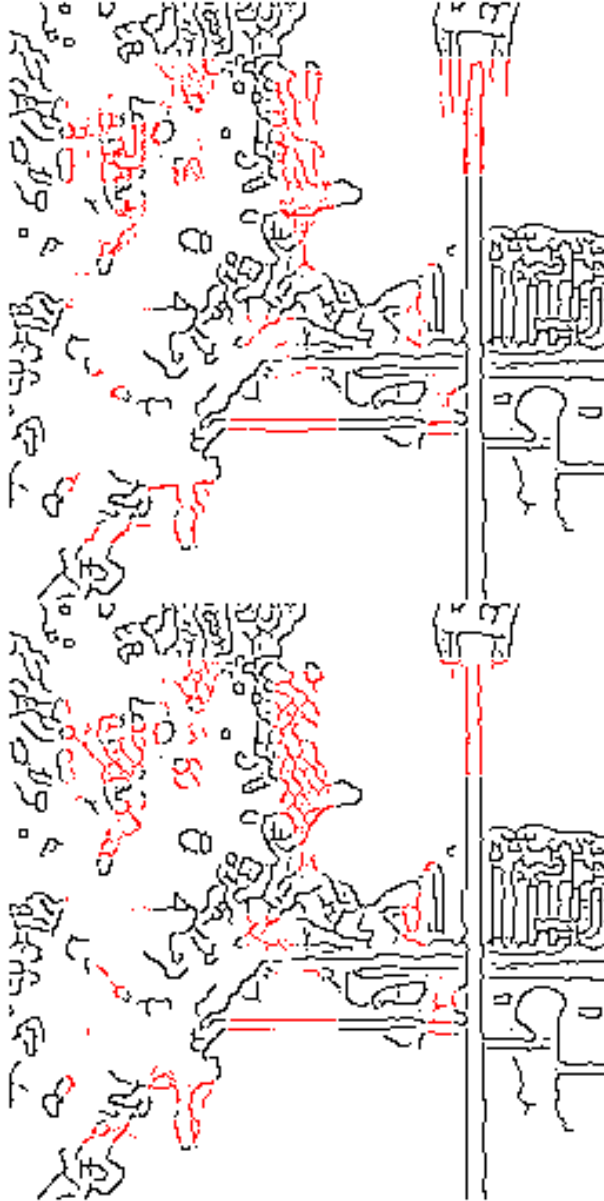
Edge Connect

MEDFE

Ours

GT

MECNet



input

mecnet(single-scale)

mecnet(multi-scale)

