

Data Directory

/media/preethamam/Utilities-SSD/Xtreme_Programming/Z_Data/DLCrack/Liu+Xincong+DS3+CrackSegNet

Proposed Method

Code Location

/media/preethamam/Utilities-SSD/Xtreme_Programming/Liu/project-dlcrack-2d

How to Train

Under /media/preethamam/Utilities-SSD/Xtreme_Programming/Liu/project-dlcrack-2d

Run `sh ./scripts/train.sh Linknet densenet169 8 400 ./ 1`

Model | backbone | batch_size | epoch | useless | useless

You can check SMP library for more available models and backbones

Sample Training Verbose:

```
Epoch: 0
train: 100% | 276/276 [01:11<00:00, 3.85it/s, DiceFocalloss - 0.8171, iou_score - 0.1916, fscore - 0.3081, precision - 0.2011, recall - 0.7166]
valid: 100% | 155/155 [00:06<00:00, 22.82it/s, DiceFocalloss - 0.6854, iou_score - 0.3645, fscore - 0.5142, precision - 0.3692, recall - 0.981]
valid: 100% | 237/237 [00:10<00:00, 22.83it/s, DiceFocalloss - 0.7836, iou_score - 0.266, fscore - 0.406, precision - 0.2686, recall - 0.9729]
valid: 100% | 184/184 [00:07<00:00, 23.85it/s, DiceFocalloss - 0.9212, iou_score - 0.1218, fscore - 0.2132, precision - 0.1261, recall - 0.7955]
valid: 100% | 154/154 [00:06<00:00, 23.83it/s, DiceFocalloss - 0.7373, iou_score - 0.2824, fscore - 0.4241, precision - 0.2884, recall - 0.9738]
valid: 100% | 250/250 [00:10<00:00, 23.82it/s, DiceFocalloss - 0.6128, iou_score - 0.452, fscore - 0.6115, precision - 0.4586, recall - 0.982]
cuda - Sat, 11 Sep 2021 07:20:18 torch_utils.py[line:293] INFO Validation score increased (inf --> 0.514209). Saving model ...
```

First row represents training performance, second row represents validation performance, the rest represent test performance(in case you wanna cheat)

Data Directory(in case you need more data):

Under /media/preethamam/Utilities-SSD/Xtreme_Programming/Liu/project-dlcrack-2d/tail_train.py

```
152 x_train_dir = os.path.join(DATA_DIR, 'TrainingCracks')
153 y_train_dir = os.path.join(DATA_DIR, 'TrainingCracksGroundtruth')
```

Change GPU

Under /media/preethamam/Utilities-SSD/Xtreme_Programming/Liu/project-dlcrack-2d/tail_train.py

```
249 model = create_model(args, n_classes, activation)
250 model = nn.DataParallel(model, device_ids=[0,1])
```

Two GPUs at least for 8 batch size.

How to Test

Under /media/preethamam/Utilities-SSD/Xtreme_Programming/Liu/project-dlcrack-2d

Run

```
sh ./scripts/performance_analysis.sh all 1
```

Or for a single dataset:

```
sh ./scripts/performance_analysis.sh Liu 1
```

Dataset Name(one of Liu Xincong CrackSegNet DS3)

The script will output scores and predictions.

The output of predictions and groundtruth is under /media/preethamam/Utilities-SSD/Xtreme Programming/Liu/project-dlcrack-2d/pred/tail test.py

If you want to change threshold, go /media/preethamam/Utilities-SSD/Xtreme Programming/Liu/project-dlcrack-2d/performance_analysis.py

```
287     for threshold in range(253,254):
```

With a little modification, you can also generate scores for all different threshold

Loads specific weights weights for testing

under /media/preethamam/Utilities-SSD/Xtreme Programming/Liu/project-dlcrack-2d/pred/tail test.py

```
129     PATH = os.path.join('./logs/8/', 'best_model_iou.pth_311')
```

Parameters:

Epoch: see above

Batch_size: see above

Learning Rate:

In /media/preethamam/Utilities-SSD/Xtreme Programming/Liu/project-dlcrack-2d/tail train.py

```
252     # define optimizer
253     lr = 0.00001
254     optimizer = torch.optim.Adam([dict(params=model.parameters(), lr=lr)]) #SOTA
```

Optimizer:

In /media/preethamam/Utilities-SSD/Xtreme Programming/Liu/project-dlcrack-2d/tail train.py

```
252     # define optimizer
253     lr = 0.00001
254     optimizer = torch.optim.Adam([dict(params=model.parameters(), lr=lr)]) #SOTA
```

Data augmentation

In /media/preethamam/Utilities-SSD/Xtreme Programming/Liu/project-dlcrack-2d/torch_utils.py

```

305 # define heavy augmentations
306 def get_training_augmentation():
307     train_transform = [
308
309         A.HorizontalFlip(p=0.5),
310
311         A.ShiftScaleRotate(scale_limit=0.5, rotate_limit=45, shift_limit=0.2, p=0.5,
312
313         A.VerticalFlip(p=0.4),
314
315         #A.RandomCrop(height=250, width=250, p=0.2),
316
317         A.PadIfNeeded(min_height=img_dims_height, min_width=img_dims_height, p=1),
318         A.IAAGaussianNoise(p=0.3),
319         A.IAAPerspective(p=0.5),
320
321         A.OneOf(
322             [
323                 A.CLAHE(p=1),
324                 A.RandomBrightness(p=1),
325                 A.RandomGamma(p=1),
326             ],
327             p=0.2,
328         ),
329
330         A.OneOf(
331             [
332                 A.IAASharp(p=1),
333                 A.Blur(blur_limit=3, p=1),
334                 A.MotionBlur(blur_limit=3, p=1),
335             ],
336             p=0.2,
337         ),
338
339         A.OneOf(
340             [

```

Loss Function:

You can find a lot in [/media/preethamam/Utilities-SSD/Xtreme_Programming/Liu/project-dlcrack-2d/torch_utils.py](#)

One wired thing: the horizontalFlip and VerticalFlip will have a huge impact on CrackSegNet performance, WHICH SHOULD NOT HAPPEN.

Logs:

```

best_model_iou.pth_0 best_model_iou.pth_13 best_model_iou.pth_173 best_model_iou.pth_24 best_model_iou.pth_41 best_model_iou.pth_6 best_model_iou.pth_9 Xincong.json
best_model_iou.pth_1 best_model_iou.pth_130 best_model_iou.pth_18 best_model_iou.pth_26 best_model_iou.pth_43 best_model_iou.pth_67 CrackSegNet.json
best_model_iou.pth_10 best_model_iou.pth_132 best_model_iou.pth_19 best_model_iou.pth_28 best_model_iou.pth_48 best_model_iou.pth_7 DS3.json
best_model_iou.pth_100 best_model_iou.pth_139 best_model_iou.pth_2 best_model_iou.pth_3 best_model_iou.pth_5 best_model_iou.pth_74 Liu.json
best_model_iou.pth_11 best_model_iou.pth_14 best_model_iou.pth_20 best_model_iou.pth_31 best_model_iou.pth_50 best_model_iou.pth_78 temp.pth
best_model_iou.pth_115 best_model_iou.pth_155 best_model_iou.pth_22 best_model_iou.pth_38 best_model_iou.pth_51 best_model_iou.pth_79 train.json
best_model_iou.pth_12 best_model_iou.pth_17 best_model_iou.pth_23 best_model_iou.pth_4 best_model_iou.pth_56 best_model_iou.pth_83 valid.json

```

Model checkpoints with epochs

```

best_model_iou.pth_0 best_model_iou.pth_13 best_model_iou.pth_173 best_model_iou.pth_24 best_model_iou.pth_41 best_model_iou.pth_6 best_model_iou.pth_9 Xincong.json
best_model_iou.pth_1 best_model_iou.pth_130 best_model_iou.pth_18 best_model_iou.pth_26 best_model_iou.pth_43 best_model_iou.pth_67 CrackSegNet.json
best_model_iou.pth_10 best_model_iou.pth_132 best_model_iou.pth_19 best_model_iou.pth_28 best_model_iou.pth_48 best_model_iou.pth_7 DS3.json
best_model_iou.pth_100 best_model_iou.pth_139 best_model_iou.pth_2 best_model_iou.pth_3 best_model_iou.pth_5 best_model_iou.pth_74 Liu.json
best_model_iou.pth_11 best_model_iou.pth_14 best_model_iou.pth_20 best_model_iou.pth_31 best_model_iou.pth_50 best_model_iou.pth_78 temp.pth
best_model_iou.pth_115 best_model_iou.pth_155 best_model_iou.pth_22 best_model_iou.pth_38 best_model_iou.pth_51 best_model_iou.pth_79 train.json
best_model_iou.pth_12 best_model_iou.pth_17 best_model_iou.pth_23 best_model_iou.pth_4 best_model_iou.pth_56 best_model_iou.pth_83 valid.json

```

Performance log in json format, key value: {precision, recall, f1, iou_score, epochs}

Plot

Code:

[/media/preethamam/Utilities-SSD/Xtreme_Programming/Liu/project-dlcrack-2d/analysis/plot.py](#)

Input the path of .json file, the program will read train.json and valid.json. Then it will produce corresponding plot. (Axis might be wrong)

If you want to output precision, recall or other scores wrt epochs, change following code:

```
76 data_dir = "/media/preethamam/Utilities-SSD/Xtreme_Programming/Liu/project-dlcrack-2d/logs/1/30Data/Linknet/densenet169/8/DiceFocalLoss/0.0001/NoneWeights/metrics"
77 data_lists = [data_dir + "/train.json", data_dir + "/valid.json"]
```

The plot graph will be outputted to [/media/preethamam/Utilities-SSD/Xtreme_Programming/Liu/project-dlcrack-2d/plot](#)

All Files:

```
drwxrwxrwx 1 root root 4096 Sep 11 07:38 analysis
drwxrwxrwx 1 root root 0 Feb 3 2021 conda_venv
-rwxrwxrwx 1 root root 0 Mar 16 09:07 cu42dlh_
-rwxrwxrwx 1 root root 180 Jun 5 18:24 data.json
-rwxrwxrwx 1 root root 22096 Aug 31 16:13 history_for_print.txt
drwxrwxrwx 1 root root 4096 Sep 9 15:01 logs
-rwxrwxrwx 1 root root 10085 Feb 3 2021 map_extract.py
-rwxrwxrwx 1 root root 0 Aug 31 16:12 output.txt
drwxrwxrwx 1 root root 0 Jun 9 00:32 past_data
-rwxrwxrwx 1 root root 9915 Aug 31 06:29 performance_analysis.py
drwxrwxrwx 1 root root 4096 Sep 11 07:15 plot
-rwxrwxrwx 1 root root 3128208 Mar 14 10:18 'plot_SOTA!!!!!!!.mat'
drwxrwxrwx 1 root root 4096 Sep 11 07:12 pred
drwxrwxrwx 1 root root 4096 Sep 5 16:03 pycache
-rwxrwxrwx 1 root root 3739 Feb 3 2021 Readme.md
drwxrwxrwx 1 root root 4096 Sep 11 07:03 scripts
-rwxrwxrwx 1 root root 8264 Sep 7 06:55 tail_test.py
-rwxrwxrwx 1 root root 14754 Sep 11 07:07 tail_train.py
-rwxrwxrwx 1 root root 21394 Sep 5 16:03 torch_utils.py
```

Torch_utils.py include data preprocessing, resizing, data reading, loss function and log class

Tail_test.py is responsible for inference

Tail_train.py is responsible for training

Scripts: the entrance for train, test and analysis

Plot is the folder storing the plot graphs

Logs: logs

Analysis: mostly used for plot graph

Pred: the prediction images and groundtruth images(I don't know why the first author want to output groundtruth images again but it is useless in my version)

Performance_analysis.py read from pred folder and generate all score you need