SKIN ABNORMAL DETECTION

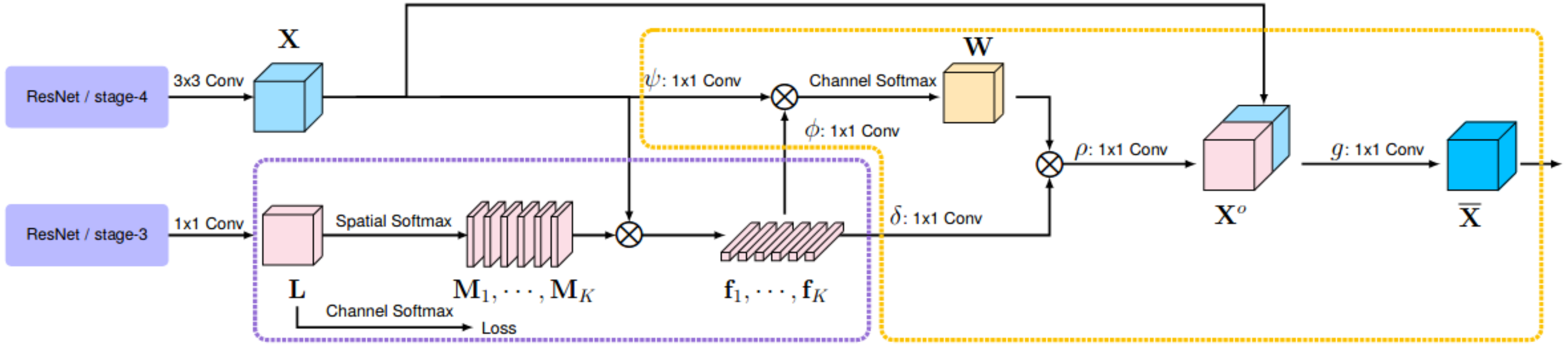
With mmsegmentation

Method used: OCRNET

Config file: ocnfigs/ocrnet/ocrnet_hr18s_4xb2-40k_cityscapes-512x1024.py

Checkpoin file:

https://download.openmmlab.com/mmsegmentation/v0.5/ocrnet/ocrnet_hr18s_4xb2-40k_cityscapes-512x1024/ocrnet_hr18s_4xb2-40k_cityscapes-512x1024_20230227_145026-6c052a14.pth



Config file modified :

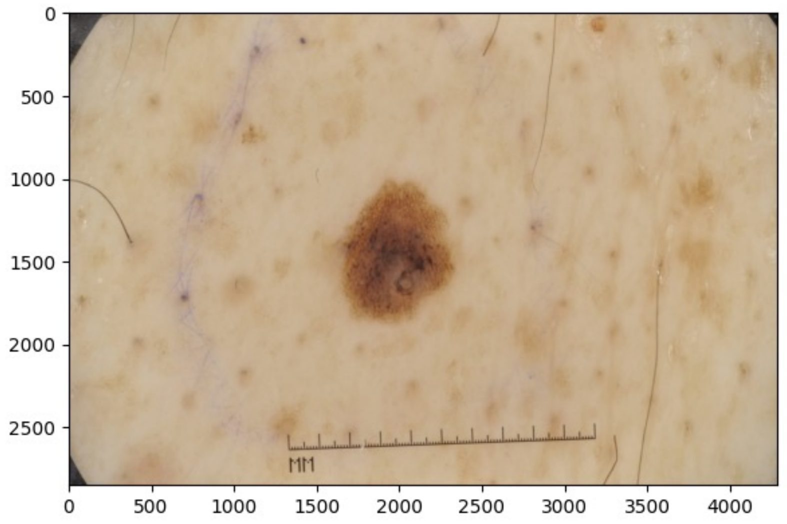
- Change data path, datatype
- Define number of batches and gpu
- Define default hook for logging and saving checkpoint
- Define dataset (type, root, prefix, pipeline)
- Define param_scheduler for auto modify learning rate
- Define the output num for two decoders
- Load check point pretrained
- Define epoch and iteration

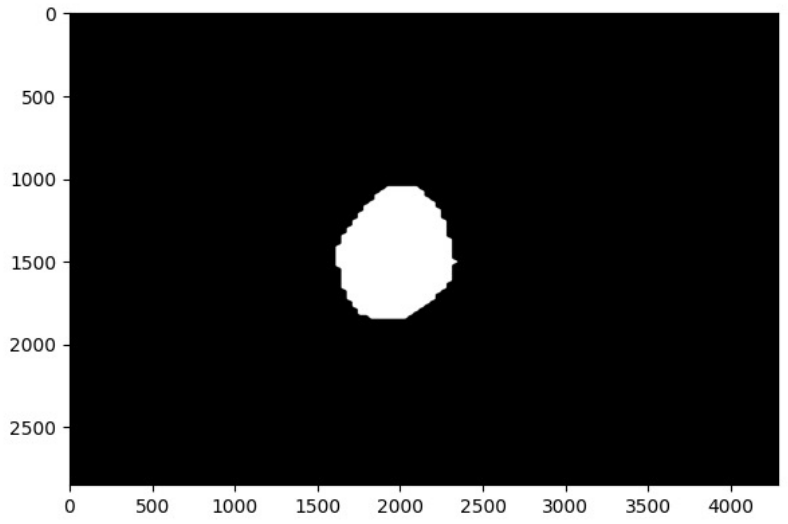
- #Define num of class
- cfg.model.decode_head[0].num_classes = 2
- cfg.model.decode_head[1].num_classes = 2
- #Define batch norm
- cfg.norm_cfg = dict(type='BN', requires_grad=True)
- cfg.model.backbone.norm_cfg = cfg.norm_cfg
- cfg.model.decode_head[0].norm_cfg = cfg.norm_cfg
- cfg.model.decode_head[1].norm_cfg = cfg.norm_cfg
- # Modify dataset type and path
- cfg.dataset_type = 'ISICDATASET_'
- cfg.data_root = '/content/dataset'
- cfg.train_dataloader.dataset.type = 'ISICDATASET_'
- cfg.train_dataloader.dataset.data_root = '/content/dataset'
- cfg.train_dataloader.dataset.data_prefix = dict(img_path='images/ISIC2018_Task1-2_Training_Input', seg_map_path='groundTruth/ISIC2018_Task1_Training_GroundTruth')
- cfg.train_dataloader.dataset.pipeline = cfg.train_pipeline
-
  cfg.val_dataloader.dataset.type = 'ISICDATASET_'
- cfg.val_dataloader.dataset.data_root = '/content/dataset'
- cfg.val_dataloader.dataset.data_prefix = dict(img_path='images/ISIC2018_Task1-2_Validation_Input', seg_map_path='groundTruth/ISIC2018_Task1_Validation_GroundTruth')
- cfg.val_dataloader.dataset.pipeline = cfg.test_pipeline

- cfg.test_dataloader.num_workers = 2
- cfg.test_dataloader.batch_size = 1
- cfg.test_dataloader.dataset.data_root = '/content/dataset'
- cfg.test_dataloader.dataset.data_prefix = dict(img_path='images/ISIC2018_Task1-2_Test_Input', seg_map_path='groundTruth/ISIC2018_Task1_Test_GroundTruth')
- cfg.test_dataloader.dataset.pipeline = cfg.test_pipeline
- cfg.test_evaluator = dict(
- type='IoUMetric',
- iou_metrics=['mIoU'],
- format_only=False,
- output_dir='work_dirs/format_results'
-)
- # the number of samples and workers per GPU
- cfg.train_dataloader.batch_size = 4
- cfg.train_dataloader.num_workers = 1
- cfg.work_dir = './work_dirs/final'
- cfg.train_cfg = dict(
- type='EpochBasedTrainLoop', max_epochs=5, val_begin=1, val_interval=1)

- `cfg.param_scheduler = [`
- `dict(type='LinearLR', by_epoch=False, start_factor=0.1, begin=0, end=200),`
- `dict(`
- `type='PolyLR',`
- `eta_min=0.0001,`
- `power=0.9,`
- `begin=0,`
- `end=160,`
- `by_epoch=False)`
- `]`
- `cfg.default_hooks = dict(`
- `timer=dict(type='IterTimerHook'),`
- `logger=dict(type='LoggerHook', interval=50, log_metric_by_epoch=False),`
- `param_scheduler=dict(type='ParamSchedulerHook'),`
- `checkpoint=dict(type='CheckpointHook', interval = 1000, by_epoch=False),`
- `sampler_seed=dict(type='DistSamplerSeedHook'))`
-
  `cfg.log_processor = dict(by_epoch=True)`
- `cfg['randomness'] = dict(seed=32)`
- `cfg.dump('/content/mmsegmentation/configs/ocrnet/ocrnet_khanh.py')`
-
  `#Load pretrain model`
- `cfg.load_from = "https://download.openmmlab.com/mmsegmentation/v0.5/ocrnet/ocrnet_hr18s_4xb2-40k_cityscapes-512x1024/ocrnet_hr18s_4xb2-40k_cityscapes-512x1024_20230227_145026-6c052a14.pth"`

Dataset





- Add custom palatte for each label
- Modify file images suffix and seg_map_suffix

Train command

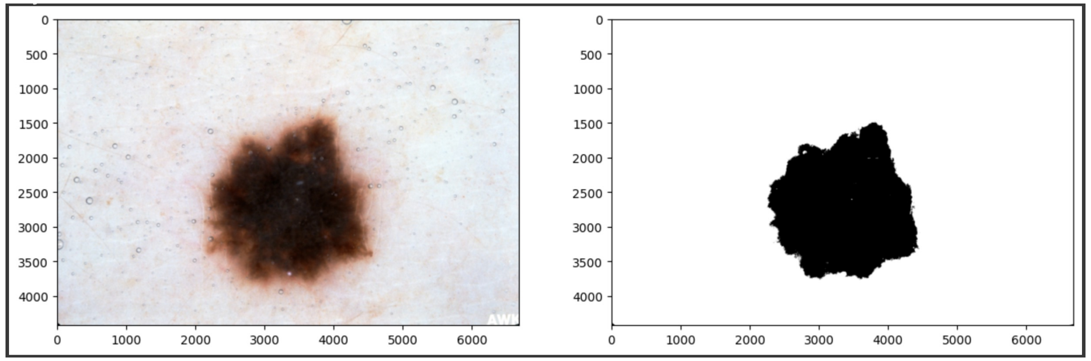
```
from mmengine.runner import Runner

runner = Runner.from_cfg(cfg)

runner.train()
```

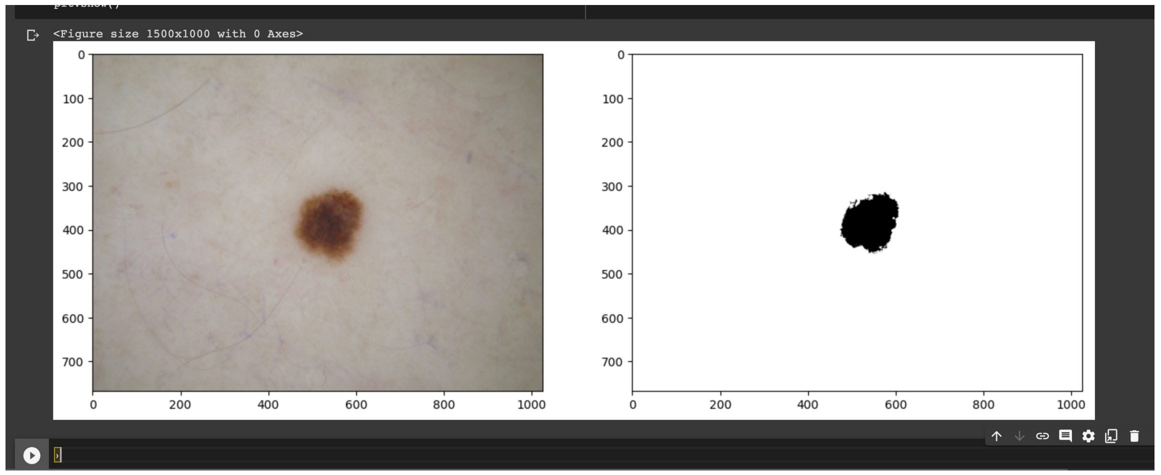
Evaluation command

- `cfg = Config.fromfile('/content/mmsegmentation/configs/ocrnet/ocrnet_khanh.py')`
- `checkpoint_path = '/content/mmsegmentation/work_dirs/tutorial/iter_1000.pth'`
- `cfg.model.pretrained = checkpoint_path`
- `runner_1 = Runner.from_cfg(cfg)`
- `runner.test()`

Review:

The model does extremely good at predict the abnormal region of for all the test images.
At the early stage of trainning, model reach high accuracy, and then quickly to converge.





```
<Figure size 1500x1000 with 0 Axes>
```