

SEGMENTATION CÁC LOẠI BỆNH VỀ DA

THÀNH VIÊN

Phan Ngọc Anh
Nguyễn Trí Ban
Tô Gia Huy
Trần Đức Hiếu
Phạm Thanh Hà
Trần Nguyễn Quang Lâm

GVHD: Nguyễn Mạnh Hùng



FILE CONFIG GỐC

https://github.com/open-mmlab/mmdetection/blob/main/configs/mask2former/mask2former_swin-l-in22k-384x384-pre_8xb2-160k_ade20k-640x640.py

FILE CHECKPOINT GỐC

https://download.openmmlab.com/mmdetection/v0.5/mask2former/mask2former_swin-l-in22k-384x384-pre_8xb2-160k_ade20k-640x640/mask2former_swin-l-in22k-384x384-pre_8xb2-160k_ade20k-640x640_20221203_235933-7120c214.pth

BUILDING DATASET

```
from mmseg.registry import DATASETS
from mmseg.datasets import BaseSegDataset

classes = ('outside', 'inside')
palette = [[0, 0, 0], [255, 255, 255]]

@DATASETS.register_module()
class ISICTask1Dataset(BaseSegDataset):
    METAINFO = dict(classes = classes, palette = palette)
    def __init__(self, **kwargs):
        super().__init__(img_suffix='.jpg', seg_map_suffix='_segmentation.png', **kwargs)
```

Thiết lập khung cho việc tải và xử lý dữ liệu hình ảnh và bản đồ phân đoạn cho tập dữ liệu ISIC Task 1 bằng thư viện MMsegmentation.

CHANGE CONFIG

Thay đổi Num_classes cho toàn bộ File Config

Đổi loại Dataset thành custom từ trước vì loại dataset không được hỗ trợ và thay đổi đường dẫn tới dataset đã thiết lập

Thay đổi Batch_size và cấu hình số lượng Worker để tối ưu quá trình huấn luyện tốt hơn, do model quá nặng

```
cfg.num_classes = 2
# Modify dataset type and batch
cfg.dataset_type = 'ISICTask1Dataset'

cfg.data_root = '/content/dataset'

cfg.train_data_loader.batch_size = 2
cfg.train_data_loader.num_workers = 2
cfg.val_data_loader.num_workers = 1
cfg.test_data_loader.num_workers = 2
```

CHANGE CONFIG

```
cfg.work_dir = './work_dirs/tutorial'

cfg.train_cfg = dict(
    type='EpochBasedTrainLoop', max_epochs=5, val_begin=1, val_interval=1000)
# cfg.train_cfg = dict(
#     type='IterBasedTrainLoop', max_iters=300, val_interval=1)

cfg.param_scheduler = [
    dict(type='LinearLR', by_epoch=False, start_factor=0.1, begin=0, end=100),
    dict(
        type='PolyLR',
        eta_min=0.0001,
        power=0.9,
        begin=0,
        end=160,
        by_epoch=False)
]
```

Thiết lập đường dẫn để lưu quá trình huấn luyện, và thiết lập lại các thông số, lịch trình của mô hình

CHANGE CONFIG

```
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 = 250, by_epoch=False),  
    sampler_seed=dict(type='DistSamplerSeedHook'))  
cfg.dump('/content/mmsegmentation/configs/tiensiha.py')  
cfg.log_processor = dict(by_epoch=True)  
cfg['randomness'] = dict(seed=48)
```

Cấu hình các hooks cho model(checkpoint: Hook này được sử dụng để lưu lại trạng thái của mô hình sau mỗi khoảng thời gian nhất định)

Điều chỉnh cách ghi lại các thông tin về quá trình huấn luyện, ở đây là ghi lại thông tin theo mỗi epoch.

Thiết lập seed, cho quá trình tạo ra các số ngẫu nhiên trong quá trình huấn luyện, đảm bảo kết quả huấn luyện được tái lập lại một cách nhất định.

TRAIN MODEL

```
from mmengine.runner import Runner  
  
runner = Runner.from_cfg(cfg)
```

Tạo một đối tượng Runner từ cấu hình cfg đã được cấu hình trước đó

```
runner.train()
```

Thực thi quá trình huấn luyện mô hình bằng cách sử dụng đối tượng Runner đã được khởi tạo từ cấu hình cfg.

Trong quá trình huấn luyện, Runner sử dụng các thông số được cấu hình trước đó để cập nhật trọng số của mô hình. Ngoài ra, các hooks được cài đặt trong cfg cũng được sử dụng để ghi lại các thông tin về quá trình huấn luyện, bao gồm các thông số đánh giá và thời gian huấn luyện.

TEST MODEL

```
▶ cfg = Config.fromfile('/content/mmdetection/work_dirs/tutorial/mask2former_r50_8xb2-160k_ade20k-512x512.py')  
checkpoint_path = '//content/mmdetection/work_dirs/tutorial/iter_4000.pth'  
cfg.model.pretrained = checkpoint_path # set the path of the checkpoint  
cfg.model.backbone.init_cfg = None
```

```
[ ] runner_1 = Runner.from_cfg(cfg)
```

```
[ ] runner_1.test()
```

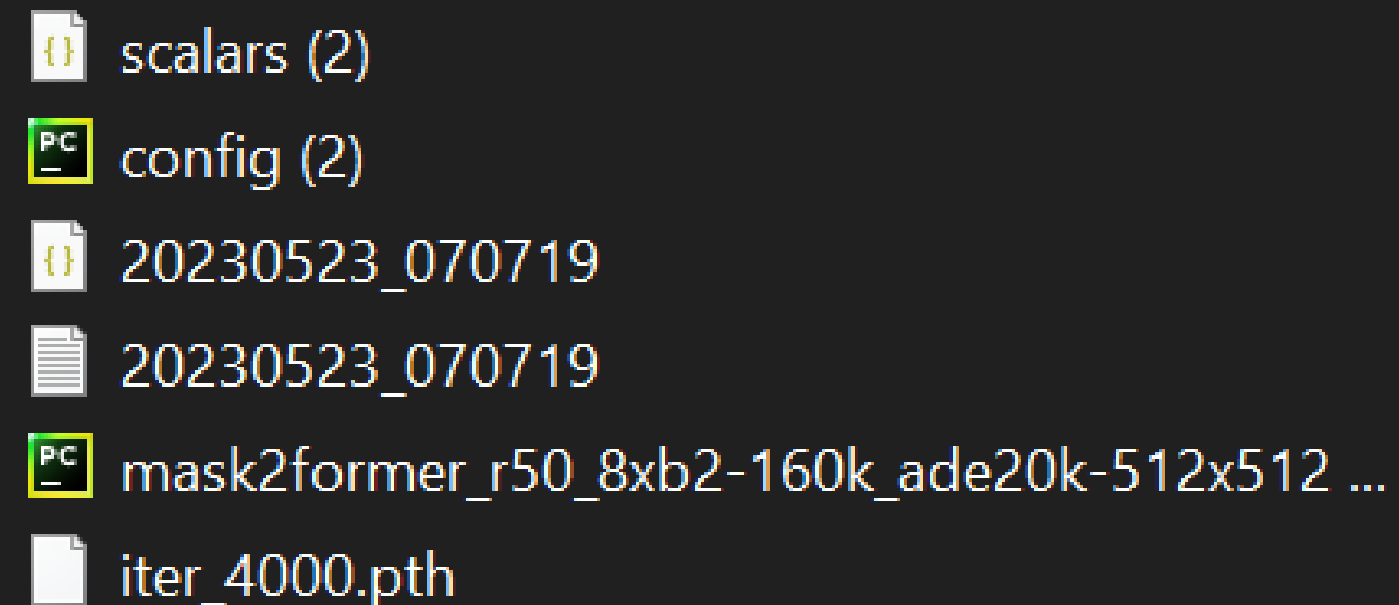
Thực hiện câu lệnh Test tương tự câu lệnh Train, sử dụng phương thức Test bằng đối tượng Runner_1, tuy nhiên bị lỗi CUDA out of Memory, chỉ có thể test trên từng ảnh, không thể test trên toàn tập dataset.

Test Thất bại

CONCLUSION

```
05/23 07:55:05 - mmengine - INFO - Epoch(train) [1][3800/1297] lr: 1.0000e-04 eta: 0:33:20 tim
05/23 07:55:42 - mmengine - INFO - Epoch(train) [1][3850/1297] lr: 1.0000e-04 eta: 0:32:42 tim
05/23 07:56:18 - mmengine - INFO - Epoch(train) [1][3900/1297] lr: 1.0000e-04 eta: 0:32:04 tim
05/23 07:56:56 - mmengine - INFO - Epoch(train) [1][3950/1297] lr: 1.0000e-04 eta: 0:31:27 tim
05/23 07:57:33 - mmengine - INFO - Exp name: mask2former_r50_8xb2-160k_ade20k-512x512_20230523_0
05/23 07:57:33 - mmengine - INFO - Epoch(train) [1][4000/1297] lr: 1.0000e-04 eta: 0:30:50 tim
05/23 07:57:33 - mmengine - INFO - Saving checkpoint at 4000 iterations
```

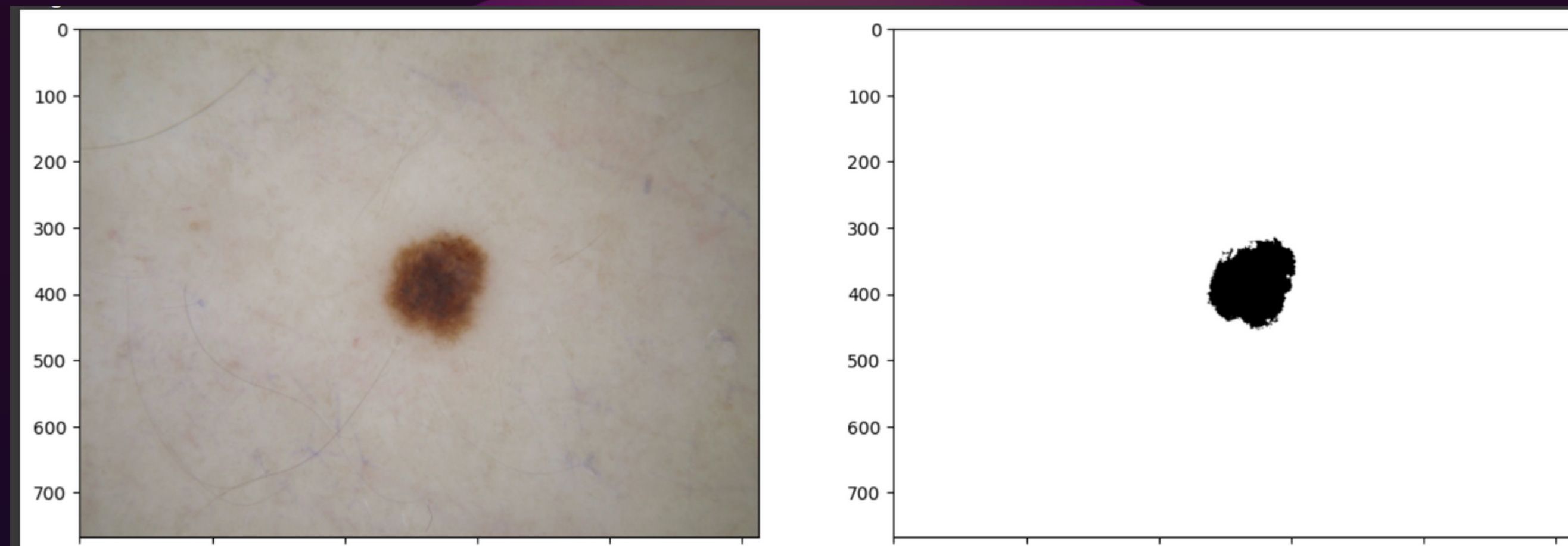
Kết thúc quá trình huấn luyện, thu File check point ở iter 4000



- scalars (2)
- config (2)
- 20230523_070719
- 20230523_070719
- mask2former_r50_8xb2-160k_ade20k-512x512 ...
- iter_4000.pth

Các File Json, checkpoint, config phát sinh thu được

CONCLUSION



Kết quả khi Test trên 1 hình ảnh trong Dataset

CONCLUSION

Nhận Xét về Quá trình Huấn luyện

Việc sử dụng mô hình Mask2Former có thể không phù hợp với bài toán, hoặc không phù hợp với phần cứng.
Model khá nặng, tập Dataset quá lớn, làm quá trình huấn luyện khá khó khăn,

MMSegmentation là một thư viện mã nguồn mở mạnh mẽ và đa năng được sử dụng để giải quyết các bài toán phân loại ảnh, đặc biệt là trong lĩnh vực xử lý ảnh y tế. Sau khi làm quen với thư viện, sẽ thấy rằng MMSegmentation rất mạnh mẽ và có thể xử lý các tác vụ phân loại ảnh nhanh chóng và hiệu quả.

Các methods và backbone trong MMSegmentation thường có dung lượng lớn, đòi hỏi phải có phần cứng đáp ứng để thực hiện. Nếu không có phần cứng đủ mạnh, việc sử dụng MMSegmentation có thể gặp phải các vấn đề về hiệu suất và thời gian xử lý.

