PROGRESS REVIEW

HUBMAP + HPA - HACKING THE HUMAN BODY

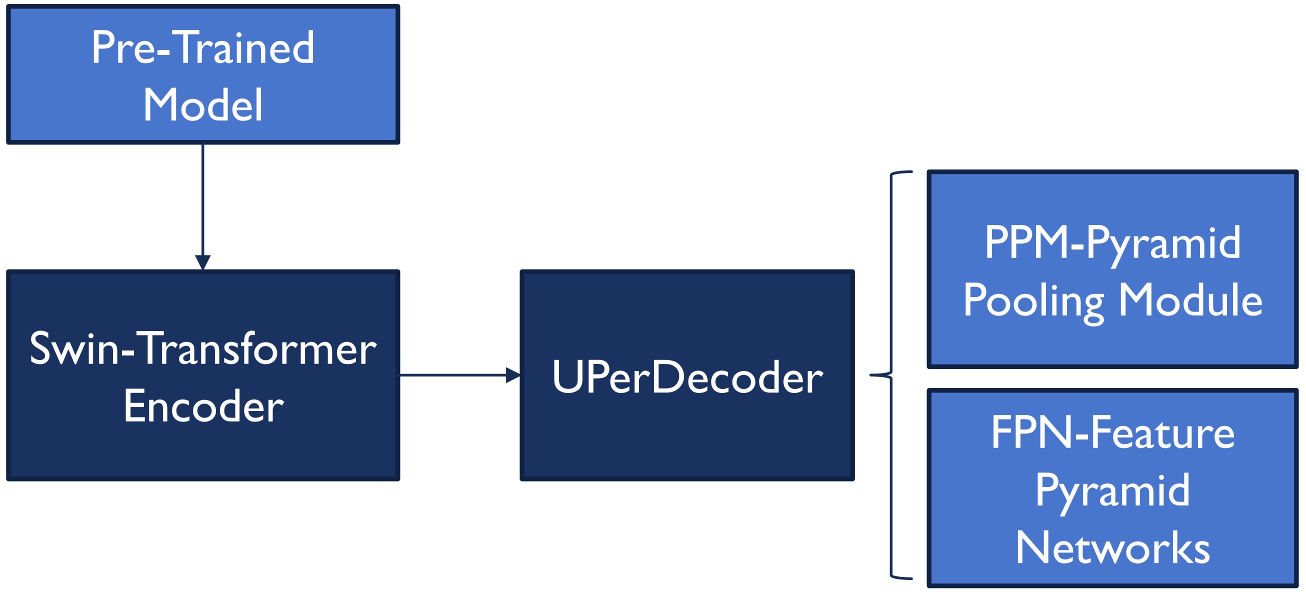
GROUP B

CONTENT

- Currently Model Structure
- Full 768*768 samples

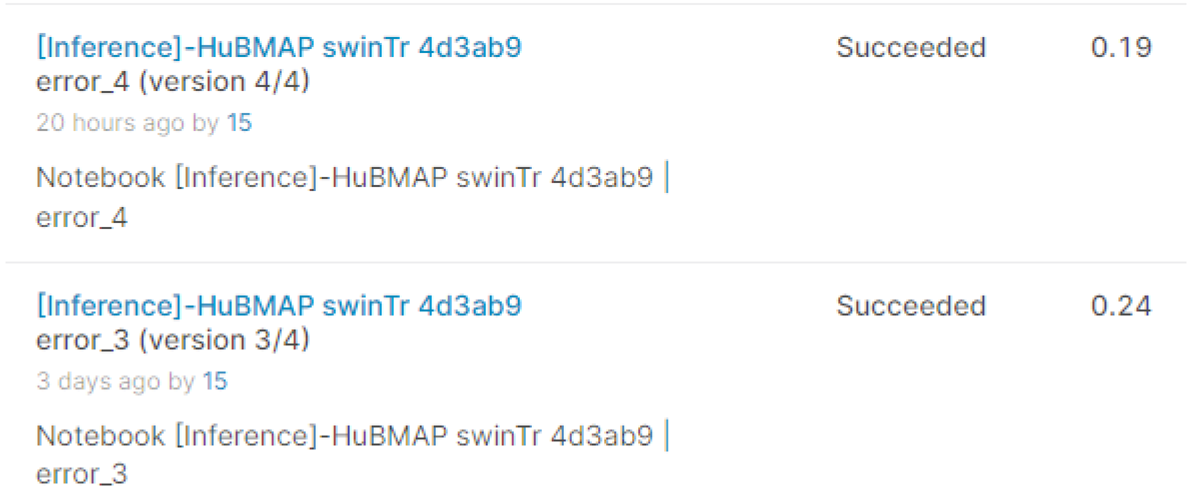
SWIN-TRANSFORMER V1 + UPERNET

- For the sake of novelty, we have designed a new variation of Swin-Transformer, which is a combination of Swin-transformer V1 + UperDecoder.

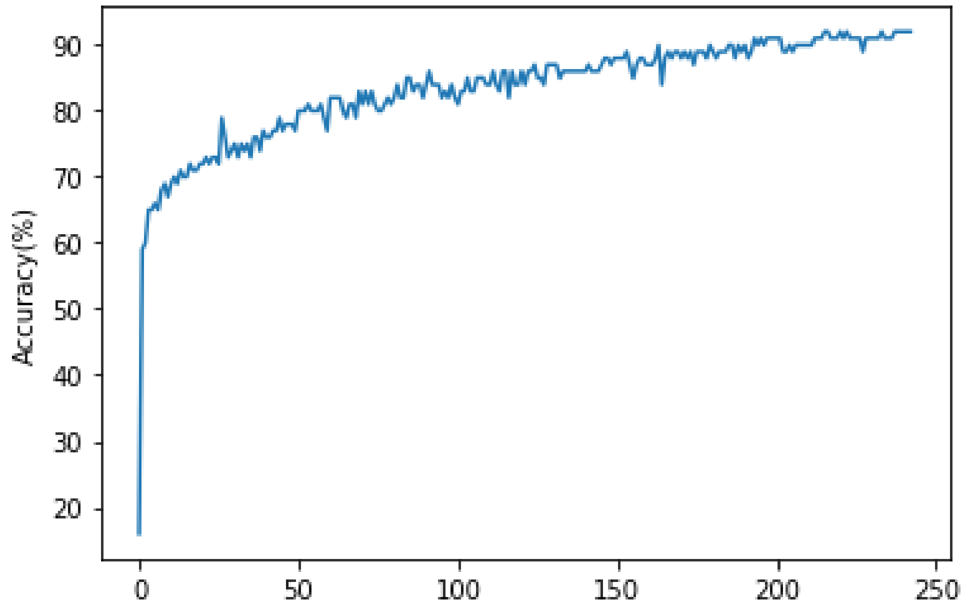


EXPERIMENTS ON IT

- We have get decent local results on it. But when it comes to submission, even though it looks correct in visualization, the scores are quite low. That means our inference notebook may be problematic.


[Inference]-HuBMAP swinTr 4d3ab9 error_4 (version 4/4) Succeeded 0.19
20 hours ago by 15
Notebook [Inference]-HuBMAP swinTr 4d3ab9 | error_4

[Inference]-HuBMAP swinTr 4d3ab9 error_3 (version 3/4) Succeeded 0.24
3 days ago by 15
Notebook [Inference]-HuBMAP swinTr 4d3ab9 | error_3



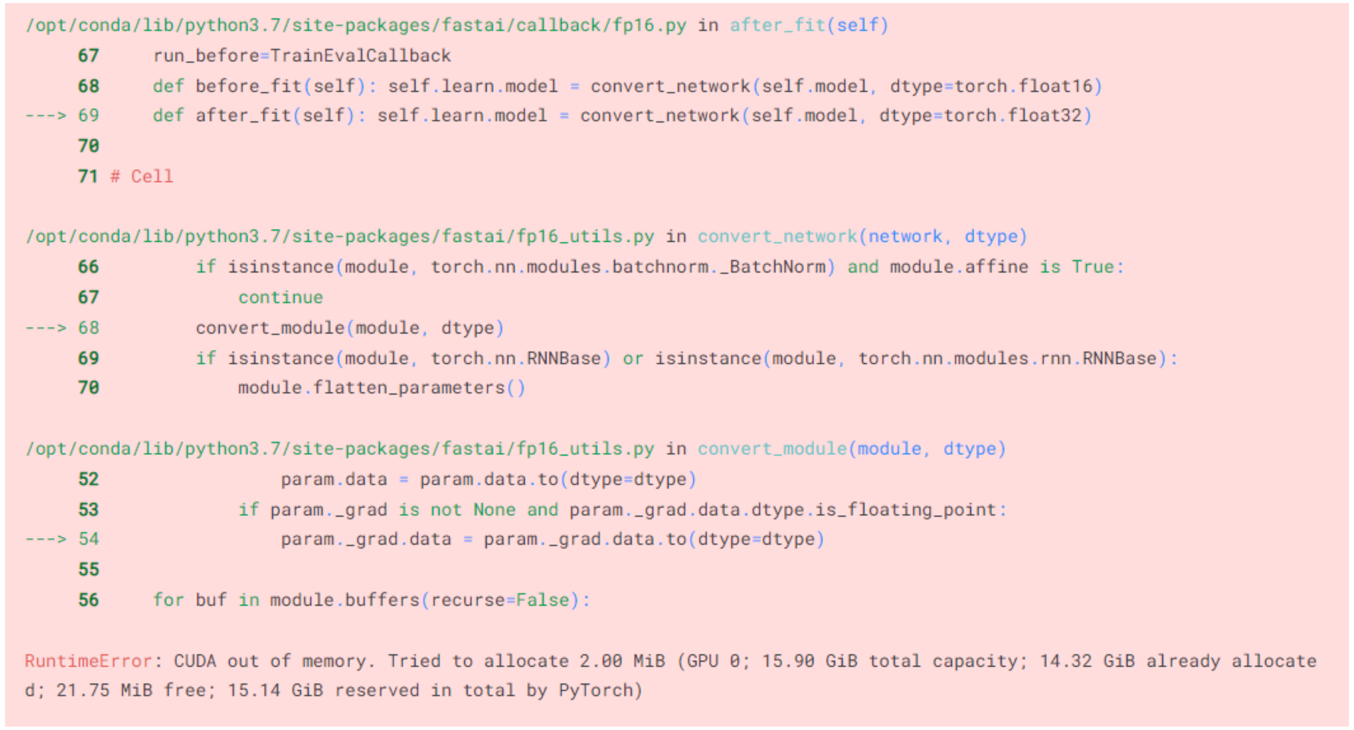
ANOTHER ATTEMPT ON DATASET

- We have found that using 256*256 tiles in training may not be helpful for transformer related networks, therefore we decided to use resized whole image in training to make use of slide window mechanisms which should be of the size divisible to 256, in our implementations: 768*768.

- However, we have found that there is no pre-trained Swin-Transformer model parameters on this size of datasets. Therefore, we decided to make some experiments on EfficientNet based FPN first.

SIZE PROBLEMS

- We can only have extremely small batch size (of 4 or 2) to work with under sample size of 768.



```
/opt/conda/lib/python3.7/site-packages/fastai/callback/fp16.py in after_fit(self)
     67     run_before=TrainEvalCallback
     68     def before_fit(self): self.learn.model = convert_network(self.model, dtype=torch.float16)
---> 69     def after_fit(self): self.learn.model = convert_network(self.model, dtype=torch.float32)
     70 
     71 # Cell

/opt/conda/lib/python3.7/site-packages/fastai/fp16_utils.py in convert_network(network, dtype)
     66         if isinstance(module, torch.nn.modules.batchnorm._BatchNorm) and module.affine is True:
     67             continue
---> 68         convert_module(module, dtype)
     69         if isinstance(module, torch.nn.RNNBase) or isinstance(module, torch.nn.modules.rnn.RNNBase):
     70             module.flatten_parameters()

/opt/conda/lib/python3.7/site-packages/fastai/fp16_utils.py in convert_module(module, dtype)
     52                 param.data = param.data.to(dtype=dtype)
     53             if param._grad is not None and param._grad.data.dtype.is_floating_point:
---> 54                 param._grad.data = param._grad.data.to(dtype=dtype)
     55 
     56     for buf in module.buffers(recurse=False):

RuntimeError: CUDA out of memory. Tried to allocate 2.00 MiB (GPU 0; 15.90 GiB total capacity; 14.32 GiB already allocate
d; 21.75 MiB free; 15.14 GiB reserved in total by PyTorch)
```

- Therefore, we are still coding for better memory allocation strategy for training.

FUTURE PLANS

- Debugging on what we have for now.
- And combining them as a valid result which may improve the benchmark.

Thank you!