Comparison between PSPNet an FPN on solar panel dataset pv01

At first I want to have a brief introduction about each of these architectures of neural networks . One of the important disadvantages of classic CNNs was the missing of detailed information about scene context in deeper layers in these 2 architectures we want to solve this problem.

PSPNET

In neural networks size of receptive fields can show that how much information about the context we have . The empirical receptive field of CNN is much smaller than the theoretical one, especially on high-level layers. This makes many networks not sufficiently incorporate the momentous global scenery prior.

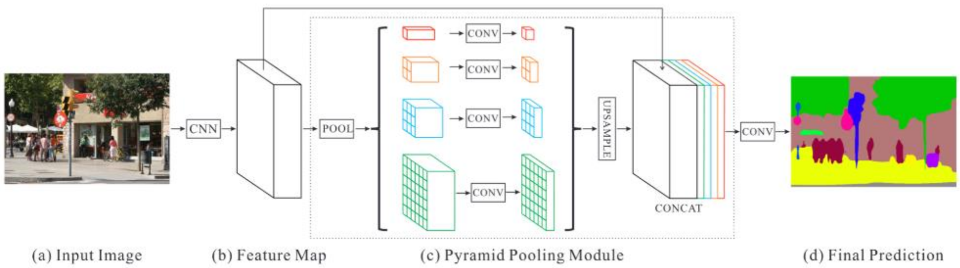

Figure 3. Overview of our proposed PSPNet. Given an input image (a), we first use CNN to get the feature map of the last convolutional layer (b), then a pyramid parsing module is applied to harvest different sub-region representations, followed by upsampling and concatenation layers to form the final feature representation, which carries both local and global context information in (c). Finally, the representation is fed into a convolution layer to get the final per-pixel prediction (d).

The pyramid pooling module fuses features under four different pyramid scales. The coarsest level highlighted in red is global pooling to generate a single bin output. The following pyramid level separates the feature map into different sub-regions and forms pooled representation for different locations.

FPN

In FPN instead of pooling we used a featured map and we use lateral connections to merge these featured maps. This approach's computational cost is much more than the PSPnet .

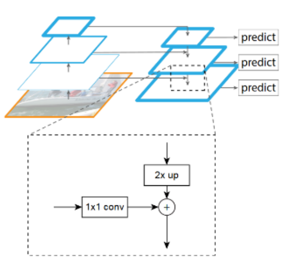

Figure 3. A building block illustrating the lateral connection and the top-down pathway, merged by addition.

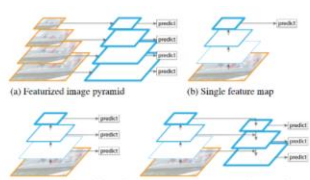

Figure 1. (a) Using an image pyramid to build a feature pyramid. Features are computed on each of the image scales independently, which is slow. (b) Recent detection systems have opted to use only single scale features for faster detection. (c) An alternative is to reuse the pyramidal feature hierarchy computed by a ConvNet as if it were a featurized image pyramid. (d) Our proposed Feature Pyramid Network (FPN) is fast like (b) and (c), but more accurate. In this figure, feature maps are indicate by blue outlines and thicker outlines denote semantically stronger features.

Comparison:

In comparison, we can see that the total performance of FPN is much higher than PSPnet (f1score:0.857,f1score:0.964) on the test set and also we can see that the final loss of the FPN on the test set is less than PSPnet(fig2, fig5). On the other hand if look at the learning curve of these 2 models we can see that the learning curve on the validation set for FPN is much more unstable than PSPnet but it's hard to make a judgment because the training last for only 5 epochs.

Final conclusion: we can say that the FPN is much better but the time of training and time for execution of the model on test sets is worth so if we are working on the problem the execution should be faster and we have a limited amount of resource maybe PSPnet is a better idea.

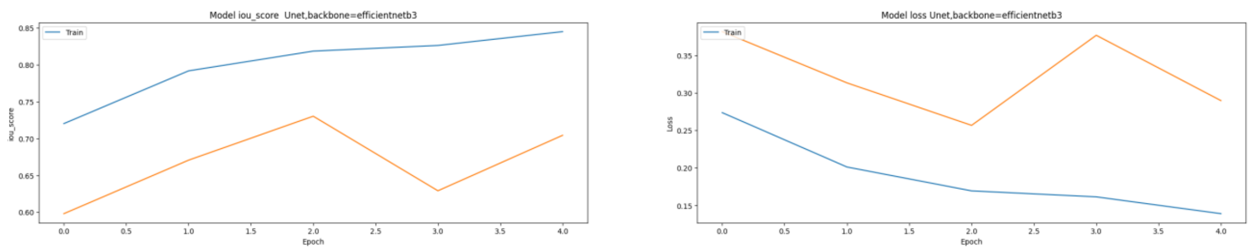

Figure 1 PSPnet learning Curve (blue train, orange validation)

```
91/91 [==============================] - 34s 376ms/step - loss: 0.1981 - iou_score: 0.7780 - f1-score: 0.8578
Loss: 0.19806
mean iou_score: 0.77799
mean f1-score: 0.85776
```

Figure 2Performance pf PSPnet on test set

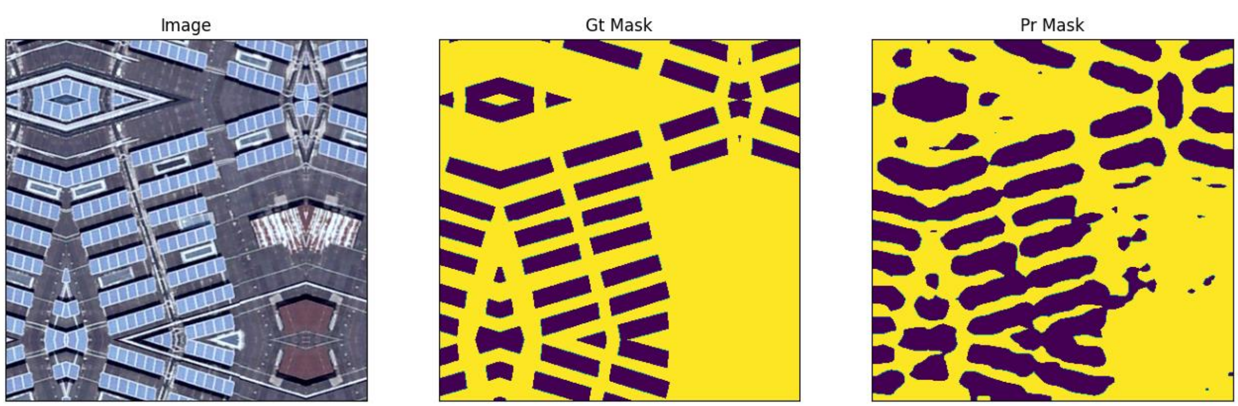

Figure 3 example performance of PSPnet on sample test image

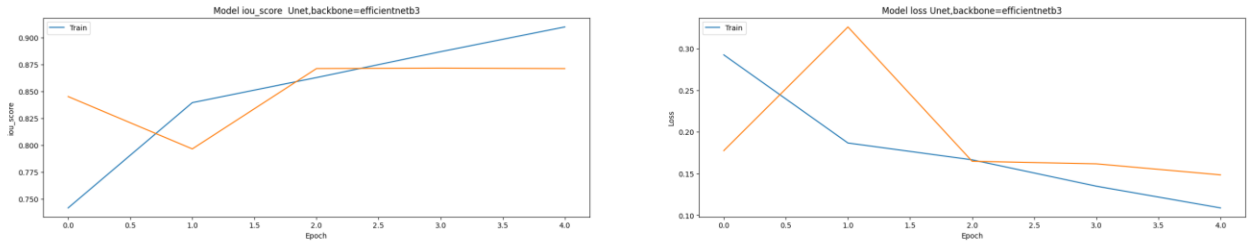

Figure 4 FPN learning Curve (blue train, orange validation)

```
91/91 [==============================] - 102s 1s/step - loss: 0.0782 - iou_score: 0.9333 - f1-score: 0.9643
Loss: 0.078244
mean iou_score: 0.93326
mean f1-score: 0.96433
```

Figure 5 Performance pf FPN on test set

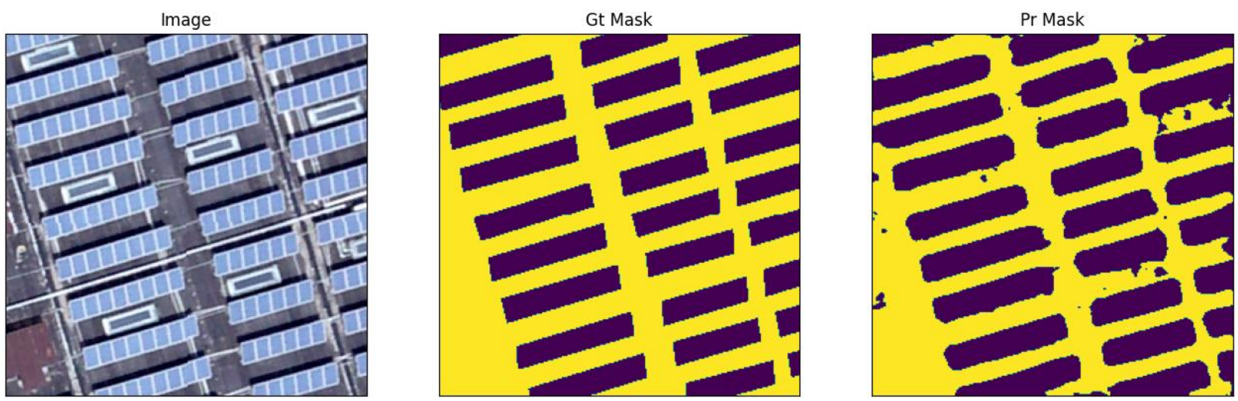

Figure 6 example performance of FPN on sample test image