Semantic Segmentation with Deep Learning

Gautham Gururajan

Parts 4 & 5: mIoU of different models

mIoU after each stage of the model

	Training mIoU	Validation mIoU
Simple Segmentation Net (no pretrained weights)	0.4024	0.4251
+ ImageNet-Pretrained backbone	0.4672	0.4402
+ Data augmentation	0.4603	0.4992
ImageNet-Pretrained PSPNet w/ Data Aug. without PPM	0.5992	0.5804
+ PSPNet with PPM	0.5663	0.5640
+ PSPNet with auxiliary loss	0.6369	0.6031

Parts 4 & 5: Per class IoUs

The model’s IoU for the 11 Camvid classes :

Class Index	Class name	Simple Segmentation Net Class IoU	PSPNet Class IoU
0	Building	0.7849	0.8789
1	Tree	0.7928	0.8638
2	Sky	0.7628	0.9025
3	Car	0.6108	0.8098
4	SignSymbol	0.0000	0.0000
5	Road	0.8557	0.9257
6	Pedestrian	0.0133	0.3239
7	Fence	0.2079	0.5492
8	Column_Pole	0.0000	0.0179
9	Sidewalk	0.6123	0.7730
10	Bicyclist	0.0106	0.5308

Parts 4 & 5: Most difficult classes

Which classes have the lowest mIoU and why they may be the most difficult :

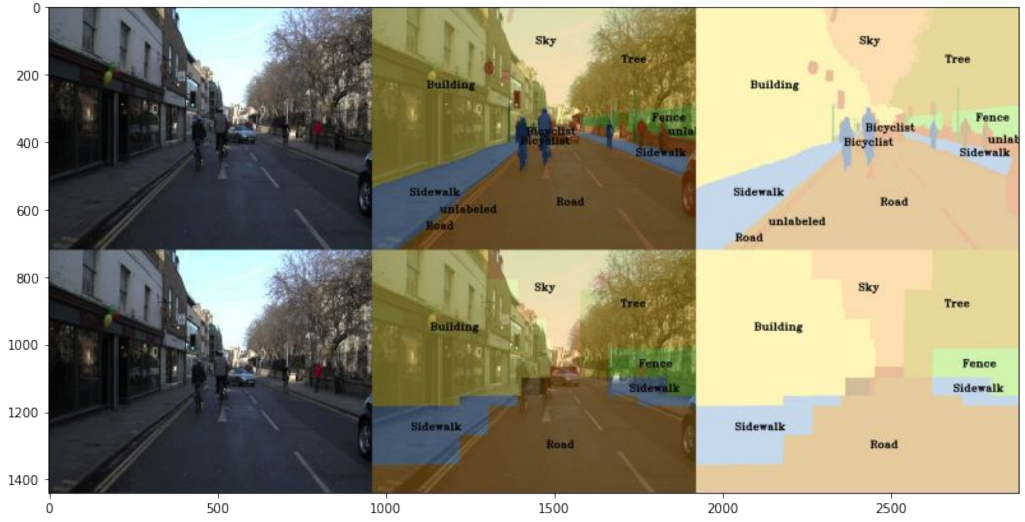
- The classes with the lowest mIoU are ‘SignSymbol’ and ‘Column_pole’, both circled in the RGB image as seen on the right. As you can see, the area of occupancy of both objects is relatively less. It is hard even for the human eye to identify ‘SignSymbol’ here. The PPM aspect of PSP net uses different bin sizes to capture different spatial resolutions, but the objects themselves are very ambiguous in this sense, column poles are similar to building outlines and sign symbols are not visible.

- Another possible reason can be due to class imbalance, the above classes do not have many labels as compared to classes like road or building.

- Even from our PSPNet results on the right, we can see that the model does not even identify it as an object despite there being multiple column poles visible. Again, this has to do with the model not being able to detect at such a high spatial resolution along with the object being hidden due to bad light and low area of occupancy.

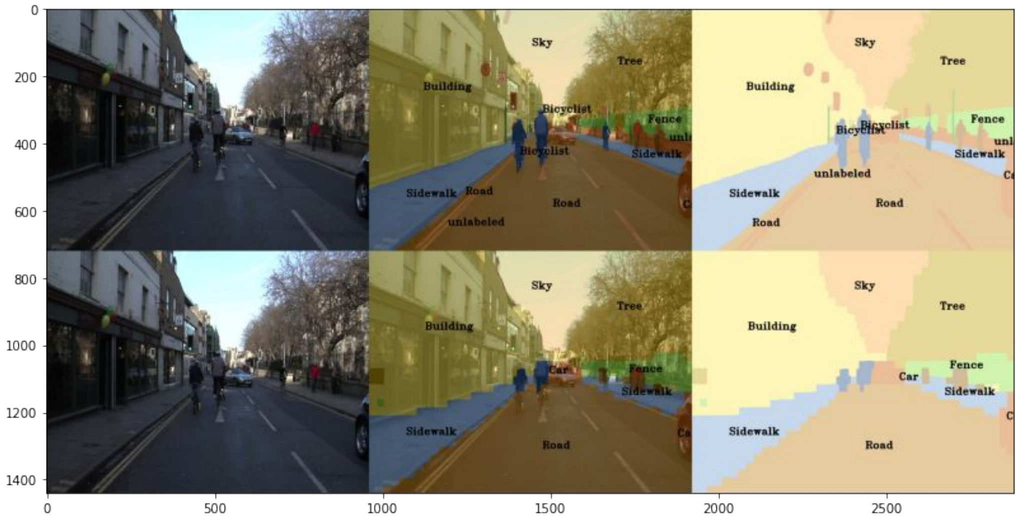
Part 4: Simple segmentation net qualitative results

Ground truth on the top row and SSN predictions on the bottom row



Part 5: PSPNet qualitative results

Ground truth on the top row and PSPNet predictions on the bottom row



Part 6: Transfer Learning

Model's IoU for the Kitti Dataset.

	mIoU	mAcc/	allAcc
Train result	0.9108	0.9521	0.9717
Val result	0.8964	0.9385	0.9674
Class Index	Class name	iou	accuracy
0	Road	0.8315	0.8932
1	Not_Road	0.9612	0.9837

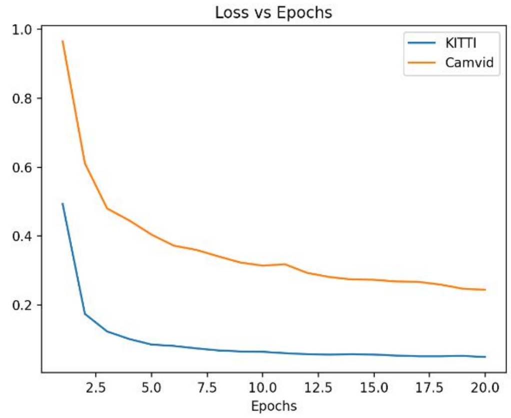
Part 6: Transfer Learning

Comparing the training loss generated when training on Kitti dataset and Camvid dataset :

- The model applied to kitti data-set has training loss decreased at a faster rate. This is the main objective of transfer learning (to achieve faster convergence with pretrained weights).

- For Camvid, the model has it's backbone (Resnet-50) loaded with pretrained weights from Image-Net, whereas Kitti has weights loaded from the model trained with Camvid. Image-net has a much higher number of classes (1000), from which 11 classes are learnt (all not mutually exclusive in Image-net data set), while Kitti learns from Camvid data set (Only 11 classes, all mutually exclusive in Kitti labeling), of which "Road" is already a label.

- Having "Road" already as a label makes the learning of weights comparatively trivial.



As seen above, kitti decreases at a higher rate. Also, Initial accuracy is significantly high as well, as expected from the previous discussion.