For the pretraining task we selected the extracted backbone of an image segmentation model that was already pretrained on complex tasks. The idea was to start pretraining from a better initialization than just a standard COCO-pretrained resnet50.

model\_backbone\_pretrained = torchvision.models.segmentation.deeplabv3\_resnet50(pretrained=True)

This backbone already has a higher spatial awareness than standard resnet50 with COCO pretraining. After a 12 epoch pretraining on the cityscapes train-extra we have an even better resnet50 for backbone to masked rcnn as we tuned it more to create good feature maps from cityscapes images that include lots of small and detailed objects.

Backbone in total has 281 weights 265 of them being from resnet50 and 16 of them being from Feature Pyramid Network.

One caveat is that deeplabv3 implementation doesn’t include a Feature Pyramid Network (FPN) that is a critical component in extracting resnet50 features in a way that is useful for RPN or Masked RCNN. As a result, after transferring the 265 backbone weights to masked rcnn we unfreezed all of the parameters in the network (including all backbone parameters and FPN). This is crucial so that FPN adapts to the transferred weights. In short the the methodology was:

1. **Pretraining DeepLabV3:** We start by loading a pretrained DeepLabV3 implementation to obtain better parameter initialization. We then pretrain DeepLabV3 on the Cityscapes semantic segmentation dataset. At this stage, the FPN is not involved; the ResNet-50 backbone learns features that are more useful for Cityscapes images.
2. **Transferring the Backbone to Mask R-CNN:** Next, we plug the pretrained ResNet-50 backbone into Mask R-CNN. We train the full network on instance segmentation tasks with all parameters unfrozen. This allows the FPN to adapt to the newly learned backbone features.
3. **Freezing Backbone and FPN:** Finally, we freeze the backbone, FPN, and RPN weights and train only the ROI heads. The ROI heads are initialized randomly to allow for flexible experimentation with hyperparameters. This approach reduces the number of parameters being trained, speeds up convergence, and makes hyperparameter tuning more efficient.

In the figure the percentual difference between the original deeplabv3 resnet50 and the pretrained weights is shown. As we previously stated there are total of 265 weights that were transferred from deeplabv3 resnet50 backbone that was trained for semantic segmentation task to the backbone of Mask-RCNN. An interesting pattern is that the relative difference is more pronounced at the first few layers of the backbone and the difference gradually decreases as we go deeper.

✅ Common keys: 265

📂 bn1 | keys: 4 | avg % diff: 108.66%

📂 conv1 | keys: 1 | avg % diff: 300.24%

📂 layer1 | keys: 50 | avg % diff: 97.88%

📂 layer2 | keys: 65 | avg % diff: 50.85%

📂 layer3 | keys: 95 | avg % diff: 37.57%

📂 layer4 | keys: 50 | avg % diff: 44.02%

