Low overhead segmentation with Shallow UNet

Manavjeet Singh, Jignesh Gutta, Debajyoti Halder

Introduction

- Latest deep learning models require high computational resources (high-end GPUs with hundreds of GFLOPS for one inference).
- With the rising demand of ML on edge computing, models needs to be optimized for use by low-powered edge devices.
- Prior works: YOLOv8-Nano **[1]** and MobileNet **[2]**
- Segmentation tasks have wide applicability: autonomous vehicles, agriculture, medical image analysis, etc **[3]**.
- This work: Modified U-Nets for Segmentation tasks in low-powered edge devices with comparable performance to Residual U-Nets **[4]**.

Evaluation Plan

Models evaluated:

- Residual UNet (4 Layers)
- 2 Layer UNet
- 2 Layer UNet with dilated convolution.
- 2 Layer Unet with pretrained residual inception unit as the encoder.

All the models were trained in float32 FP precision for Coco and CityScapes datasets; and were evaluated in the following precisions:

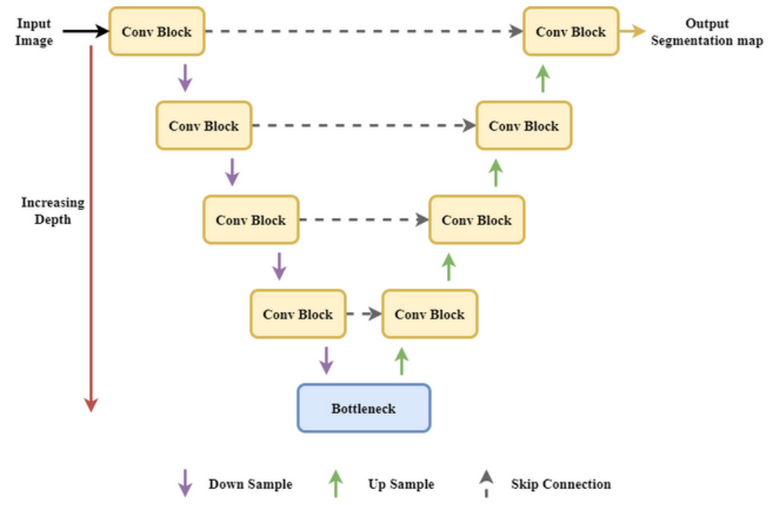
- float32
- float16
- int8

We report the following metrics:

- mIoU (mean Intersection over Union): As the name suggests it is the mean of area of overlap divided by the area of union of predicted segmentation and the ground truth. (Higher is better).
- Number of MAC (Multiply–accumulate) operations: This represents the number of operations required to get an inference from the model. (Lower is better).

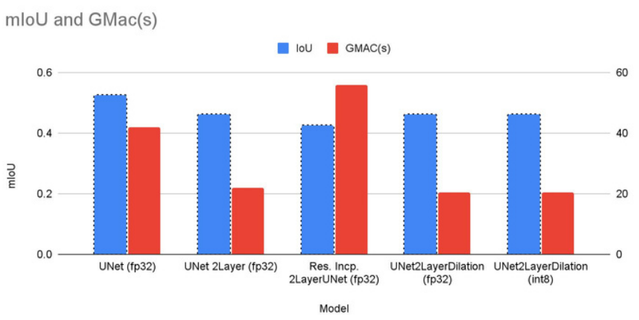
Note: Operations for a model will remain same irrespective of it's floating point accuracy.

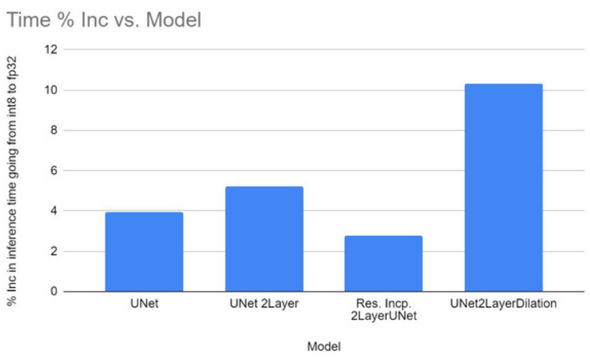
What is a UNet?



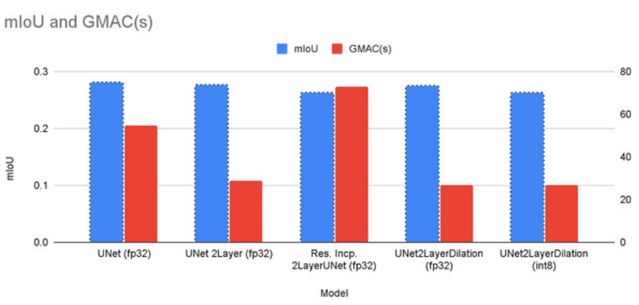
Evaluation Results

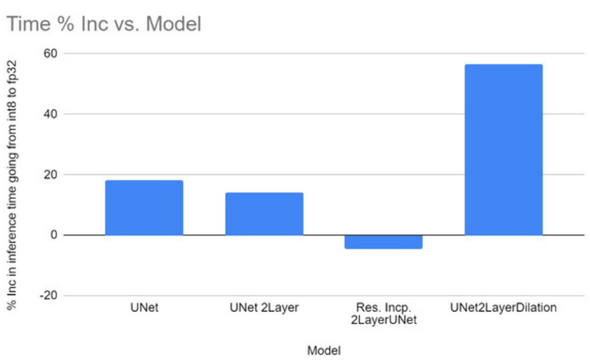
Coco (Nvidia T400)





CityScapes (Google Colab with Nvidia Tesla T4)





Observations

Lower MACs with dilated convolution

$$G[i][j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} H[u][v]F[i-u][j-v]$$

$$G[i][j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} H[u][v]F[i-r.u][j-r.v]$$

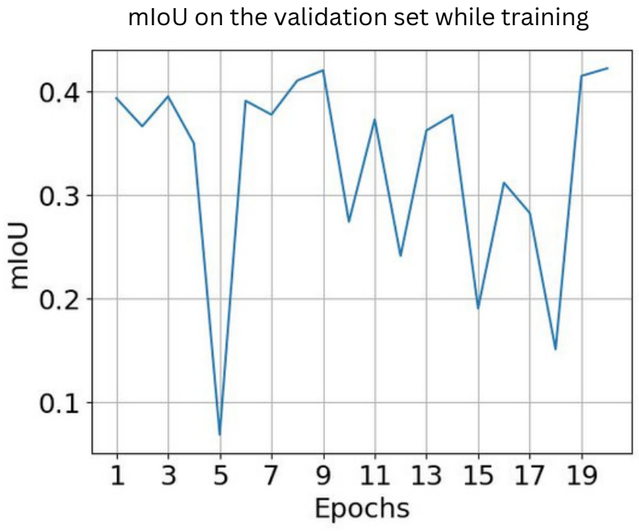
Kernel Size: 2k

Dilation: r

- Increasing the dilation factor results in smaller output matrix (G) which reduces the mathematical operations in lower layers.

Lower performance of Residual Inception U-Net

mIoU on the validation set while training



- mIoU doesn't show an increasing trend with respect to epochs, hinting towards a lack of train data and/or a lower number of epochs.

Conclusions

- If there is a limited amount of training data, it is good to use 2 layer UNet. Otherwise Residual Inception UNet can be tried.
- Replacing normal convolution with atrous/dilated convolutions always reduces the required computation costs without affecting accuracy.
- Lowering FP precision had no effect on accuracy of the model. Low precision int8 model can be used for faster inference with no accuracy drop.
- We did our final testing on desktop class GPUs instead of mobile devices, due to their unavailability; however, we present our results in terms of MAC operations and relative times which are easy to disaggregate from the testing hardware and the accuracy results will remain the same irrespective of the hardware used
- UNet 2 Layer Dilation, which performs with only 11.64 and 2.14 percent lower mIoU than the original UNet but with less than half MAC operations required and 36.53 and 50.17 percent lower inference times on Coco and CityScape datasets respectively

Limitations

- We worked with limited train data and epochs due to the lack of compute resources.
- It would be interesting to see if there is a disparity between more complex models when the
- train data and train time increases.