Low overhead segmentation with Shallow UNet

Debajyoti Halder
114964049

Jignesh Gutta
114980214

Manavjeet Singh
114949493

December 2023

1 Introduction

It's common knowledge that the latest deep learning models require high computational resources, such as high-end GPUs (Graphics Processing Units), and hundreds of thousands of Multiply-ACcumulate (MAC) operations for one inference. With a rising trend towards edge computing, the models need to be optimized for these devices. The energy and computational overhead are dependent on the number of Multiply–accumulate operations MAC operations for a model inference and the floating point precision of these operations. Recent models like YOLOv8-Nano [7] and MobileNet [3] have been proposed to run on relatively lower compute capacity edge devices. However, these models are limited to detection tasks. To the best of our knowledge, there doesn't exist a less computationally intensive model for segmentation tasks. The ability to do segmentation tasks on lower power edge devices can have applications in autonomous vehicles, agriculture, medical image analysis, etc [5]. In this work, we propose a segmentation model with segmentation performance comparable to Residual U-Nets [6] with less than half MAC operations required and at least 36.5 percent lower inference time.

2 UNet

Before going into the details of this work, it is important to understand residual UNet model architecture (mentioned as UNet hereafter). UNet is a deep learning architecture that is widely used for various image segmentation tasks, particularly in the field of medical image analysis. It was introduced by Olaf Ronneberger, Philipp Fischer, and Thomas Brox in a 2015 paper titled "U-Net: Convolutional Networks for Biomedical Image Segmentation. [6]"

The U-Net architecture is characterized by its U-shaped structure, which is where it gets its name. It consists of an encoder part and a decoder part, connected by a bottleneck or "bridge" in the middle. The encoder downsamples the input image, extracting high-level features, while the decoder then upsamples the feature maps to produce a segmented output. The structure of the U-Net is presented in Figure 1. Key components of the U-Net architecture:

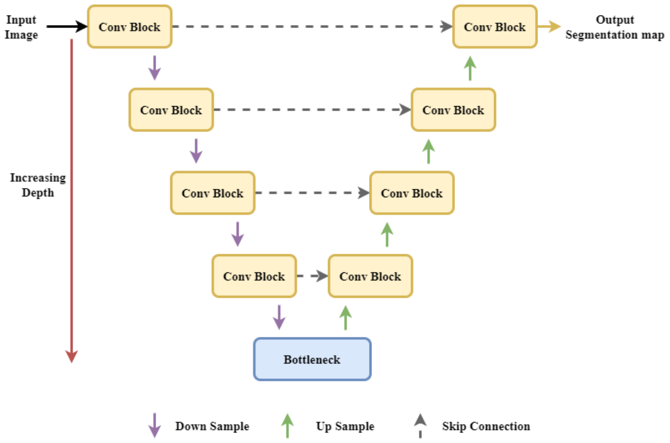


Figure 1: The architecture of the U-Net.

- **Contracting Path (Encoder):** The top half of the U-shape consists of convolutional layers and max-pooling layers. This part is responsible for reducing the spatial dimensions of the input image and capturing important features.
- **Bottleneck:** This is a narrow central part of the U-Net, where the network captures the most critical features of the input image.

- **Expansive Path (Decoder):** The bottom half of the U-shape is the decoder, which consists of up-convolutional layers (also known as transposed convolutions) and concatenation with feature maps from the contracting path. This part helps to upsample the feature maps to the original spatial resolution and generate the segmented output.
- **Skip Connections:** One of the key innovations of U-Net is the use of skip connections that connect the corresponding layers between the contracting path and the expanding path. These connections help the network preserve detailed information and overcome the vanishing gradient problem during training.

3 Models

We evaluated the following four variations of the U-Net architecture.

3.1 Residual U-Net (4 Layers)

This model is the original UNet model from the aforementioned paper, renowned for its effectiveness in biomedical image segmentation. The key feature here is the inclusion of residual connections, which create shortcuts in the network, allowing for easier training of deeper architectures by addressing the vanishing gradient problem. The model comprises four layers, striking a balance between depth for feature extraction and simplicity for computational efficiency.

3.2 2-Layer U-Net

Taking inspiration from Yolov8-nano, we reduced the UNet's model depth from 4 layers to 2 layers (see Figure 2b). This layer reduction results in significantly fewer MAC operations required for inference, reducing computational demand. While it offers benefits in terms of efficiency, the reduction in depth might impact the model's ability to extract and represent complex features, which is a critical aspect of segmentation tasks. However, this was not the case with the UNet, as we show in our evaluations in section 4.

3.3 2-Layer U-Net with Pretrained Residual Inception Unit as the Encoder

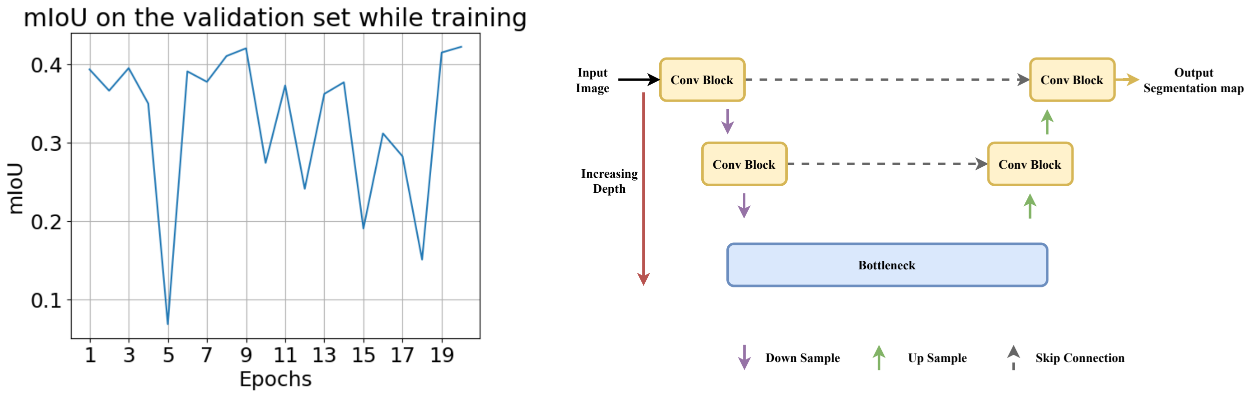
This innovative model the two layers of the encoder with the pretrained residual inception layers (first two layers from Resnet [2]). The residual inception unit combines the strengths of residual connections and inception modules, offering robust feature extraction through filters of various sizes. The pretraining of the encoder on a large dataset equips the model with an enhanced ability to generalize and recognize diverse features, potentially improving performance in segmentation tasks. The combination of pretraining with a 2-layer architecture aims to provide an optimal balance between computational efficiency and effective feature extraction resulting in higher accuracy.

3.4 2-Layer U-Net with Dilated Convolution

Enhancing the 2-layer U-Net, this model incorporates dilated convolutions. **Dilated convolution**, also known as atrous convolution, is an effective technique for increasing the receptive field of filters without increasing the number of parameters. In a standard convolution, where a filter of size $k \times k$ slides over the input feature map, each filter element is applied to consecutive elements of the input. In contrast, in dilated convolution, the filter is applied over an area larger than its actual size by skipping input values at a certain rate. This is achieved by introducing a dilation rate d(=2 in our case), which defines the stride with which we sample the input. The dilated convolution operation can be represented as:

$$\text{Output}[i, j] = \sum_{m,n} \text{Input}[i + m \cdot d, j + n \cdot d] \times \text{Filter}[m, n] \tag{1}$$

where m, n are the spatial dimensions of the filter. By adjusting the dilation rate d, we can control the spacing between the values in a kernel. A dilation rate of 1 means a conventional convolution, while higher rates result in larger receptive fields. A dilation rate of more than 1 also results in smaller dimensions of the output matrix as compared to conventional convolution, reducing the mathematical operations required in the further layers. This is the best model that we propose in this project.



(a) mIoU on validation set v/s epoch for 2 Layer Residual Inception UNet.

(b) 2 layer U-Net

Figure 2

4 Evaluation

4.1 Datasets

We evaluated our models on a subset of images from two datasets, Coco [4] (see fig 3a) and CityScapes [1] (see fig 3b). From Coco, we used permutations of 500 images for training and 50 images for validation; from CityScapes we used 2976 images for training and 500 images for validation. Experiments with the Coco dataset were performed using Nvidia T400 GPU on a local PC and experiments with CityScapes dataset were performed using Nvidia Tesla T4 GPU on Google Colaboratory.

4.2 Evaluation Metrics

We compare our models on two metrics; firstly, mean Intersection over Union (mIoU), 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. It is a common metric representing the accuracy of the model over the validation set, it achieves the value 1 if the predicted segmentation masks that exactly overlaps the ground truth, and anything above 0.35 is considered respectable. Secondly, we report the number of Multiply and Accumulate (MAC) operations required per inference for all the models described in section 3.

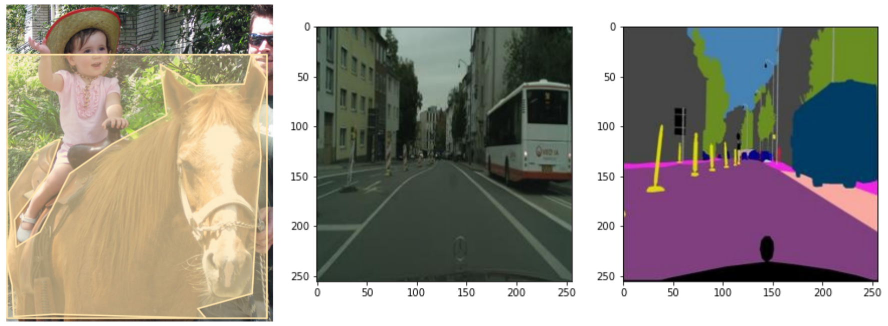
4.3 Model precision

Finally, we quantize the weights of the convolution layer to lower Floating Point Precision (FPP) levels of Float16 and Int8 from the default Float32 and evaluate them for the above-mentioned metrics. Note that the MACs for a model will remain the same, irrespective of the FPP. To make a comparison among different FPPs of the same model, we use the average time taken to perform inference on the validation set.

5 Discussion

Figure 4 and 6 shows the mIoU (left Y-axis) and Giga-MAC (or GMAC) (right Y-axis) for all 4 models tested on Coco and Cityscape dataset respectively. As expected original UNet performs the best out of the bunch but it also has the highest number of operations. By reducing the number of layers from 4 to 2, we reduced GMACs almost to half in UNet 2Layer, and that too with minimal damage to mIoU.

We attempted to increase the accuracy of the 2-layer model by replacing the first two layers of the encoder with pretrained Resnet's [2] first two layers at the cost of increased MAC. However, to our surprise, this made the performance even worse. This is because we trained our models on a very limited dataset and for only 20 and 15



(a) Coco

(b) Cityscapes

Figure 3: Example images from the dataset used.

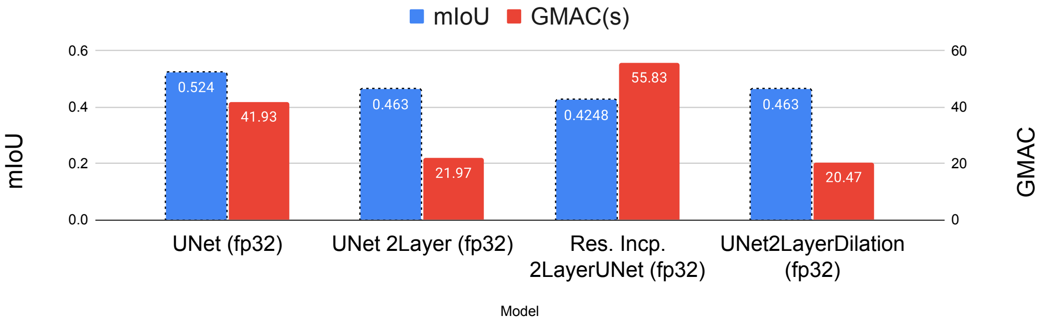


Figure 4: mIoUs and giga-MACs for the given models on the Coco dataset.

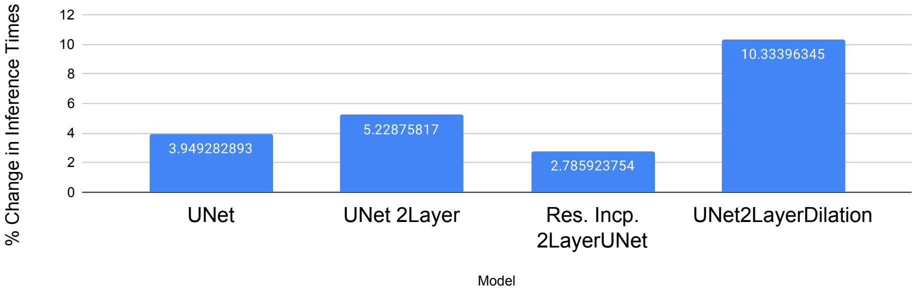


Figure 5: Percentage change in inference times when going from int8 to float 32 precision. A positive value means the time increased. For the Coco dataset.

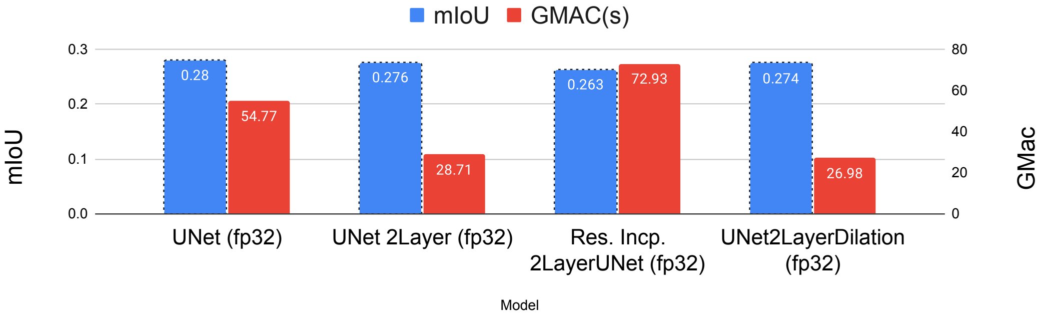


Figure 6: mIoUs and giga-MACs for the given models on the Cityscape dataset.

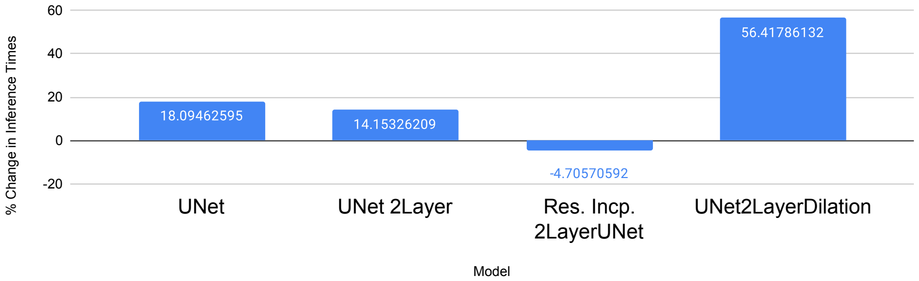


Figure 7: Percentage change in inference times when going from int8 to float 32 precision. A positive value means the time increased. For the Cityscape dataset.

epochs for Coco and CityScapes respectively due to limited GPU resource availability. This can be explained by the fact that we don't see an increasing trend in mIoU for every progressing epoch, shown in figure 2a.

Since using Residual Inception 2 Layer UNet did not increase, we worked on decreasing the number of operations further. We achieved this by using atrous/dilated convolution layers in place of conventional convolution layers. With this addition, the number of operations were reduced by almost 1.5 and 2.2 GMACs on Coco and CityScape respectively. We used Atrous convolutions with dilation factor set to 2 for the experiments, but that can be further increased to reduce the number of MAC operations. This model is shown in figures 4 and 6 as UNet2LayerDilation.

Finally, we evaluated the inference times on validation sets for FPP Float16 and Int8 versions of all the models. The percentage change in the inference times when going from FPP Int8 to Float32 is show in figures 5 and 7. For our proposed model we achieved a speedup of 10.33 and 56.41 percent for Coco and CityScape respectively, which is greater than any other model. This is because our proposed model has the least MACs. We observed that the execution time decreased by around 4 percent in Residual Inception UNet when evaluating on CityScape but we can ignore this since the difference in the values was less than 10 milliseconds.

These graphs show only a limited set of results from our experiments. For complete results please see table 1 in the Appendix.

6 Conclusions and Limitations

We make the following three observations. (i) If there is a limited amount of training data, it is good to use 2-layer UNet with dilation. Otherwise, Residual Inception UNet can be tried. (ii) In our experiments, replacing normal convolution with atrous/dilated convolutions always reduces the required computation costs without degrading the accuracy. (iii) Lowering FPP does not affect the accuracy of any varient of UNet. Low precision int8 model can be used for faster inference with no accuracy drop.

In this work, we were limited by the availability of GPU resources due to which we opted for lesser training data and time. It would be interesting to see, if and how the results change when the models are trained for longer times with much more diverse training data and will the proposed model will perform as close to the original model in terms of mIoU. 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.

In conclusion, we took a step forward towards proposing the very first UNet-based segmentation model: 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.

References

[1] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[3] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[4] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Doll'a r, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014.

[5] Keni Qiu, Nicholas Jao, Mengying Zhao, Cyan Subhra Mishra, Gulsum Gudukbay, Sethu Jose, Jack Sampson, Mahmut Taylan Kandemir, and Vijaykrishnan Narayanan. Resirca: A resilient energy harvesting reram crossbar-based accelerator for intelligent embedded processors. In *2020 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pages 315–327. IEEE, 2020.

[6] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015.

[7] Alexander Wong, Mahmoud Famuori, Mohammad Javad Shafiee, Francis Li, Brendan Chwyl, and Jonathan Chung. Yolo nano: A highly compact you only look once convolutional neural network for object detection. In *2019 Fifth Workshop on Energy Efficient Machine Learning and Cognitive Computing-NeurIPS Edition (EMC2-NIPS)*, pages 22–25. IEEE, 2019.

Appendix

7.1 Contributions

- **Manavjeet Singh:** Proposal, coding different varients of UNet models, training and results with Coco dataset, report and poster.
- **Debajyoti Halder:** Proposal, training, and results with Cityscapes dataset, report and poster.
- **Jignesh Gutta:** Proposal, deciding evaluation methodology and models to use, report and poster.

7.2 All Results

Table 1 contains results from all the experiments conducted. These numbers were used to generate all the plots in the project.

FPP	Dataset	Model	Segmentation Performance		Computational Resources
			mIoU	mDice	GMACs
Float 32	Coco	U-Net	0.524	0.63	41.93
		2 Layer U-Net	0.463	0.585	21.97
		Residual Inception U-Net	0.4248	0.5469	55.83
		2 Layer U-Net with Dilation	0.463	0.586	20.47
	CityScapes	U-Net	0.28	0.427	54.77
		2 Layer U-Net	0.276	0.421	28.71
		Residual Inception U-Net	0.263	0.406	72.93
		2 Layer U-Net with Dilation	0.274	0.419	26.98
Float 16	Coco	U-Net	0.524	0.63	41.93
		2 Layer U-Net	0.463	0.585	21.97
		Residual Inception U-Net	0.4248	0.5469	55.83
		2 Layer U-Net with Dilation	0.463	0.586	20.47
	CityScapes	U-Net	0.26	0.403	54.77
		2 Layer U-Net	0.259	0.401	28.71
		Residual Inception U-Net	0.263	0.406	72.93
		2 Layer U-Net with Dilation	0.262	0.405	26.98
Int 8	Coco	U-Net	0.524	0.63	41.93
		2 Layer U-Net	0.463	0.585	21.97
		Residual Inception U-Net	0.4248	0.5469	55.83
		2 Layer U-Net with Dilation	0.463	0.586	20.47
	CityScapes	U-Net	0.28	0.427	54.77
		2 Layer U-Net	0.276	0.421	28.71
		Residual Inception U-Net	0.263	0.406	72.93
		2 Layer U-Net with Dilation	0.274	0.419	26.98

Table 1: All experimental results