Segmentation can be light!

1st Skapetis Christos
Deepartment of Electrical Engineering
Eindhoven University of Technology
Eindhoven, the Netherlands
c.skapetis@student.tue.nl

***Abstract*—This paper explores a variety of deep learning architectures and training strategies for semantic segmentation, with a focus on optimizing performance while maintaining efficiency. Starting from a baseline U-Net architecture, several enhancements were investigated, including attention mechanisms, pretrained encoders, alternative batch sizes and image dimensions and adaptive learning rate schedules. Lightweight models such as BowlNet and AFFormer were also evaluated to assess their potential for mobile and edge deployment. Experiments demonstrate that the AFFormer architecture consistently outperforms the U-Net in every metric, despite its low parameter count. In contrast, BowlNet underperformed, suggesting limitations in generalizability. The study provides insights into balancing model complexity, training strategy, and resource constraints in semantic segmentation tasks in resource-constrained environments.**

***Index Terms*—semantic segmentation, deep learning, U-Net, attention mechanisms, pretrained models, EfficientNet, transfer learning, lightweight architecture, AFFormer, BowlNet**

I. Introduction

Semantic segmentation plays a crucial role in numerous computer vision applications such as autonomous driving, medical image processing, satellite image analysis, security. It assigns a class label to pixels using a deep learning algorithm. This task tends to be complex and computationally intensive task. To address these challenges, various deep learning architectures have been proposed, with U-Net being one of the most well-known due to its strong performance in medical image segmentation.

This work aims to evaluate and enhance the performance of some baseline models by integrating attention mechanisms, pretrained encoders, and training tricks such as learning rate scheduling and gradual unfreezing. Additionally, the study investigates lightweight architectures like BowlNet and AFFormer to assess their effectiveness and feasibility for real-time or resource-constrained scenarios.

By comparing different models in terms of accuracy, IoU, and parameter count, this research highlights the trade-offs involved in designing deep learning solutions for segmentation and suggests promising directions for future improvement.

The dataset that it is used for the evaluation and the testing is the Cityscapes Dataset [1]. This is a large, diverse set of stereo video sequences recorded in streets from 50 different cities [1].

II. Methods

In order to find the optimal model for this task, the tests started with a simple architecture, and progressive enhancements were applied to build on it, even if it performed worse than the original U-Net architecture.

The first architecture selected was a U-Net architecture with some attention mechanism added. The attention mechanism was the Attention Gate mechanism [2]. There is an attention module called Three-Level Attention (TLA) composed of an Attention Gate (AG), channel attention, and spatial normalization mechanism. The AG preserves structural information, whereas channel attention helps to model the interdependencies between the three different RGB channels.

The next step is to add a pre-trained encoder. The pre-trained encoder is ResNet34 [3]. The ResNet model is pre-trained on ImageNet-1k at resolution 224x224 [4]. Subsequent enhancements involved adding a fully connected neural network at the end. Adding a fully connected neural network (FCNN) at the end of a convolutional architecture can significantly enhance performance. This integration allows for improved generalization and accuracy. Fully connected layers augment convolutional neural networks (CNNs) to enhance performance while research indicates that incorporating FC layers can lead to a relative improvement in validation accuracy without increasing parameters during inference [5]. In tasks like semantic segmentation of high-resolution images, the addition of FC layers has been shown to improve classification accuracy and recognition of small objects [6].

Subsequently, using a scheduler, the learning rate is changed after a specific number of epochs. For the first 14 epochs (steps), a learning rate of $1e-4$ is used, but for the rest of the training process, it is diminished to $1e-5$.

In order to further improve model efficiency, three additional modifications were made. First of all, the pre-trained encoder has its layers' weights frozen for the first 9 epochs. The layers 4, 3, 2, 1 and input are getting unfrozen in the epochs 9, 11, 13, 15, 17, respectively [7]. As far as the batch size is concerned, there was an attempt to increase it from 64 to 128, but there were memory issues in the current setup. Last but not least, the size of the images used is increased from 256×256 to 512×512.

Then there was an attempt at some light models, targeting high efficiency with simple architectures. The first architecture used was a variation of BowlNet [8]. BowlNet is an Ultra Lightweight Architecture for Real-Time Semantic Segmentation. In this case it was adapted for the needs of the experiment [8]. Different variations of BowlNet have been used with varying batch sizes (64, 100, 128) and image resolutions (256, 512, 1024), but most of them encountered memory problems.

Finally, another light model was utilized. AFFormer is a head-free lightweight semantic segmentation model using a linear transformer [9]. The architecture used was an adaptation of the small version of the AFFormer with a pre-trained encoder. The encoder weights were initialized with a pretrained EfficientNet-B0 [10] model with the IMAGENET1K_V1 [11] weights. This backbone was unfreezed at later stages of the training phase [7]. That took place in the epochs 9, 11, 13, 15, and 17.

Different gradient-based techniques were considered for the optimizer choice. It is decided that an adaptive learning rate method is the best option for training a deep or complex neural network [12]. The torch libraries have only 3 implementations of optimization algorithms that are supported for CUDA, Adam, AdamW and SGD [13]. SGD is in its beta version [13]. AdamW tends to have better performance [14], while SGD tends to converge faster [14]. At the end, AdamW was selected due to the fact that SGD requires selective learning rate scaling at initialization or similar techniques to perform equally [15].

The implementation can be found here.

III. Results

The results for some models were more favourable than others and can be seen in Table III. The AFFormer architecture was outperforming against the original unet [16] both in accuracy and Intersection over Union (IoU) / Jaccard Index [17]. At the same time, it was extremely light and mobile, probably allowing running on the edge. On the other hand the BowlNet's performance was extremely poor. Its mobility wasn't combined with high accuracy or high performance in the Intersection of Union (IoU) metric. Finally, the attention mechanism added to the original unet is underperforming in size comparing with the original architecture. The increase in accuracy and IoU is substantial, but especially compared with the afformer, it does not justify the extreme increase of the parameters. The model listed below are the AFFormer twice (with 32 and 64 batch size in the training phase), the U-net architecture three times (one without attention, one with attention and one with attention and a pretrained encoder) and the BowlNet once. Despite the fact higher batch size would create expectations for higher metrics, the limited resources used allowed training for less epochs, leading to worse results.

Figure 1 shows three segmentation examples for the selected models. This qualitative comparison shows where the weakness of the models are. The Attention U-net fails to capture the background correctly focusing more in the tiny objects such as cars!

Model	**IoU**	**Accuracy**	**Params (M)**	**FLOPs (G)**
afformer-batch32	0.5034	0.5865	2.98	2.62
afformer-batch64	0.4371	0.5059	2.98	2.62
attention U-net	0.4162	0.4897	39.56	9.63
attention U-net pre	0.4890	0.5731	39.56	38.53
BowlNet	0.1944	0.2307	3.05	17.16
U-net	0.4418	0.5105	17.26	40.15

TABLE I
Performance comparison of different models

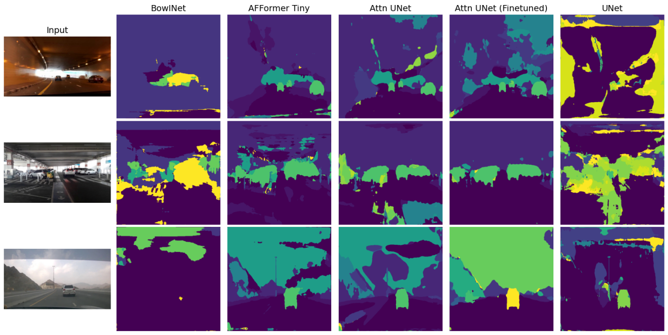

Fig. 1. Segmentation examples

IV. Discussion

The implementation of tests and evaluation experiments has led to some further conclusions. First of all, it is clear that the initial architecture was inefficient for this kind of task. Since it performs worse than the original unet architecture [16], it can be safely assumed that the modelling choices were suboptimal. In addition the BowlNet was extremely unreliable at its results, especially considering the high scores in the original research paper where it was introduced. Some modifications and different training parameters should give better results. Moreover, it is believed that the afformer architecture has great potential and room for improvement. Making it even lighter using a smaller encoder can make this model even lighter without losing much of its accuracy. Last but not least, some other models, techniques have been implemented without success. A transfer learning technique was implemented, but it was never successfully trained. Using the segment anything model [18], there were efforts to train a lighter model with transfer learning and a student-teacher relationship.

Some other interestingt concepts that are worth researching would be augmenting the data to increase the robustness of the model. Modern libraries such as imgaug have special augmentation techniques that allow for augmenting different weather or environments.

In addition, spending more time and resources would allow to explore more architectures and models, fine tuning the hyperparameters more and utilize more commonly used metrics such as Pixel Accuracy (PA) [19], Mean Pixel Accuracy (MPA), Dice Coefficient [20] [21], Frequency Weighted IoU [19], Precision and Recall [22] (per class) or even confusion matrices [23] per class or per group of classes.

LIST OF FIGURES

LIST OF TABLES

REFERENCES

[1] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[2] A. AL Qurri and M. Almekkawy, "Improved unet with attention for medical image segmentation," *Sensors*, vol. 23, no. 20, 2023. [Online]. Available: https://www.mdpi.com/1424-8220/23/20/8589

[3] Microsoft and H. Face, "Resnet-34," 2021, accessed: 2025-04-16. [Online]. Available: https://huggingface.co/microsoft/resnet-34

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015. [Online]. Available: https://arxiv.org/abs/1512.03385

[5] P. Kocsis, P. Súkeník, G. Brasó, M. Nießner, L. Leal-Taixé, and I. Elezi, "The unreasonable effectiveness of fully-connected layers for low-data regimes," 2022. [Online]. Available: https://arxiv.org/abs/2210.05657

[6] G. Chen, C. Li, W. Wei, W. Jing, M. Woźniak, T. Blažauskas, and R. Damaševičius, "Fully convolutional neural network with augmented atrous spatial pyramid pool and fully connected fusion path for high resolution remote sensing image segmentation," *Applied Sciences*, vol. 9, no. 9, 2019. [Online]. Available: https://www.mdpi.com/2076-3417/9/9/1816

[7] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," 2018. [Online]. Available: https://arxiv.org/abs/1801.06146

[8] A. S. Nadig, A. N. Preethi Tammana, A. S. V Amogh, A. Pradeepan, and M. J, "Bowlnet: An ultra lightweight architecture for real-time semantic segmentation on resource constrained environments," in *2024 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)*, 2024, pp. 719–726.

[9] B. Dong, P. Wang, and F. Wang, "Head-free lightweight semantic segmentation with linear transformer," 2023. [Online]. Available: https://arxiv.org/abs/2301.04648

[10] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," 2020. [Online]. Available: https://arxiv.org/abs/1905.11946

[11] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[12] S. Ruder, "An overview of gradient descent optimization algorithms," 2017. [Online]. Available: https://arxiv.org/abs/1609.04747

[13] PyTorch, "Pytorch optimization documentation," 2025, accessed: 2025-03-27. [Online]. Available: https://pytorch.org/docs/stable/optim.html

[14] H. Zhu, Z. Zhang, W. Cong, X. Liu, S. Park, V. Chandra, B. Long, D. Z. Pan, Z. Wang, and J. Lee, "Apollo: Sgd-like memory, adamw-level performance," 2025. [Online]. Available: https://arxiv.org/abs/2412.05270

[15] M. Xu, L. Xiang, X. Cai, and H. Wen, "No more adam: Learning rate scaling at initialization is all you need," 2024. [Online]. Available: https://arxiv.org/abs/2412.11768

[16] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," 2015. [Online]. Available: https://arxiv.org/abs/1505.04597

[17] P. Jaccard, "Étude comparative de la distribution florale dans une portion des alpes et des jura," *Bulletin de la Société Vaudoise des Sciences Naturelles*, vol. 37, p. 547–579, 1901.

[18] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," 2023. [Online]. Available: https://arxiv.org/abs/2304.02643

[19] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," in *International Journal of Computer Vision*, vol. 88, 2010, p. 303–338.

[20] L. R. Dice, "Measures of the amount of ecologic association between species," *Ecology*, vol. 26, no. 3, p. 297–302, 1945.

[21] T. Sørensen, "A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on danish commons," *Kongelige Danske Videnskabernes Selskab*, vol. 5, p. 1–34, 1948.

[22] C. J. V. Rijsbergen, "Information retrieval," *Butterworth-Heinemann*, 1979.

[23] T. Fawcett, *An introduction to ROC analysis*, 2006, vol. 27, no. 8.