REPRODUCTION OF FASTERSEG: SEARCHING FOR FASTER REAL-TIME SEMANTIC SEGMENTATION

Adomas Lebedys, Ahmad Shabir Galili & Tan Jun Xian
School of Electronics and Computer Science
University of Southampton
Southampton, UK
{al2u18,asg1g18,jxt1f17}@soton.ac.uk

ABSTRACT

In this report, a small-scale partial reproduction of the paper by Chen et al. (2019) was conducted. We focused on proving the use of knowledge distillation between the 'teacher' and the 'student' network, which, along with the neural architecture search (NAS), was one of the two major concepts mentioned in the paper. Experiments conducted include: training a teacher network and distilling it onto a student network, training the student network as well as distillation of the pre-trained teacher network. The results indicated the test accuracy of the trained student was on par with that of the distilled student of the same number of epochs and required less training time. For closer comparison to the published paper, the student was trained a further 200 epochs which achieved a test accuracy of 67.6%, 2% less than the published results despite having half the epochs. Overall, the paper was reproducible and the results gathered in this project are similar to those published.

1 INTRODUCTION

As the development of robotic systems accelerates, new applications involving real-time processing of real-world environments emerge. One topic of increasing importance is image segmentation, where every pixel within a frame is individually labelled so the scene can be partitioned into distinct segments.

In this paper, we describe attempts to reproduce the results obtained in the work by Chen et al. (2019), which proposes a novel approach to real-time image segmentation. The authors develop an original framework for neural architecture search (NAS) that exploits multi-resolution branches to achieve low latency image processing. While preserving high inference speed, the resultant model is able to achieve accuracy competitive with alternative state-of-the-art methods. A notable addition to the NAS framework is collaborative search, enabling concurrent search for two networks: the 'teacher' network, and the less complex 'student' network, which is suitable for deployment on resource-constrained hardware. The teacher conducts knowledge distillation (the transfer of knowledge) to the student for further accuracy increase.

This paper reproduces the findings of the FasterSeg paper across its various experiments. There was however a large computational cost which the original paper did not specify, and therefore the experiments were scaled down.

2 METHODOLOGY

The paper produced by Chen et al. (2019) involved training a NAS to be used to generate the teacher and student network architectures. Having produced the teacher network, supervised training on the Cityscapes dataset (Cordts et al., 2016) is used to update the weights of the network. Finally, the trained teacher network is distilled into the student network; the performance of this final network, designated **'FasterSeg'**, is evaluated.

In this paper we attempt to reproduce the work from Chen et al. (2019). Due to limited time and computational resources, an exact reproduction of the results was not feasible. Instead, replications of the experiments were constructed on a smaller scale. These experiments were as follows:

- Distillation of pre-trained teacher network into student network
- Training of the teacher network
- Distillation of trained teacher network into student network
- Training a neural network with the given student architecture
- Latency evaluation of the models

Limited access to computational resources prevented training of the NAS 'supernet' as the estimated time for this process approached a month. Moreover, while the original work trained the teacher network for 600 epochs, this was unrealistic within the time-frame. Instead, training time was decreased to 100 epochs.

Our trained teacher network was then distilled into the student architecture for 100 epochs; again, this was in contrast to 600 epochs of distillation in the original paper. To provide additional comparison, the distillation process was repeated for 100 epochs using the pre-trained teacher weights (Chen et al., 2021).

One extension to the original paper was direct training of the student network architecture using the same dataset as the teacher network. The purpose of this particular experiment was to evaluate the efficacy of the knowledge distillation process and examine its impact on the student model performance. Two instances of this model were produced: one trained for 100 epochs and another for 300. The results from the 300-epoch training served to offer a closer comparison to the FasterSeg student network in the original paper.

Latencies for inference of both the teacher and student model were computed using the same GPU as the original paper (NVIDIA 1080Ti 12GB) with the aim of achieving a fair reproduction.

3 Implementation details

The size of the project's codebase discouraged reimplementation of the model architecture, so the main focus of this project was to design experiments using the code provided (Chen et al., 2021). A large obstacle in this project was the training times of the models as well as the amount of computational power required to run it.

Training the teacher networks took upwards of 40 hours for $n = 100$ epochs. The distillation into the student took surprisingly longer at an average of 52 hours for the same n. The training of the student network architecture for $n = 300$ took approximately 56 hours, significantly less per epoch than the distilled student model. All models were trained using an Nvidia RTX 3080 Ti GPU with 12GB of RAM.

The hyperparameters from the original paper were not altered, with the aforementioned exception of the number of epochs. The optimiser used was a SGD with momentum ($m = 0.9$) working in tandem with a learning rate decay after every epoch starting from an initial value $\lambda_0 = 0.01$. Batch size was maintained at 12 for both the student and teacher network training. For reproducibility, the seed 12345 was used throughout all the networks.

Despite the large memory available, the validation subroutine in the original code triggered out-of-memory errors. A solution was found by limiting the processing to a single thread, which severely slowed down the validation process but had no effects on training or the ultimate performance of the models.

4 Results & Analysis

From Table 1, we observe that the distilled student has a slightly higher validation accuracy compared to the teacher, highlighting the positive impact of distillation. However, the student architecture that was trained without distillation managed to outperform both the teacher and distilled

Table 1: Table showing the validation accuracies of the various models trained during this project. The results for 600 epochs were obtained using the pre-trained weights provided.

	100 epochs			600 epochs (Pre-trained paper)		300 epochs
Labels	**Teacher**	**Distilled Student**	**Trained Student**	**Teacher**	**Distilled Student**	**Trained Student**
Mean IU no back	56.4%	60.2%	60.9%	73.6%	72.3%	69.4%
Mean pixel acc	92.4%	93.4%	93.3%	95.2%	95.2%	94.8%
Validation mIoU	56.7%	60.4%	61.1%	73.5%	72.3%	69.4%

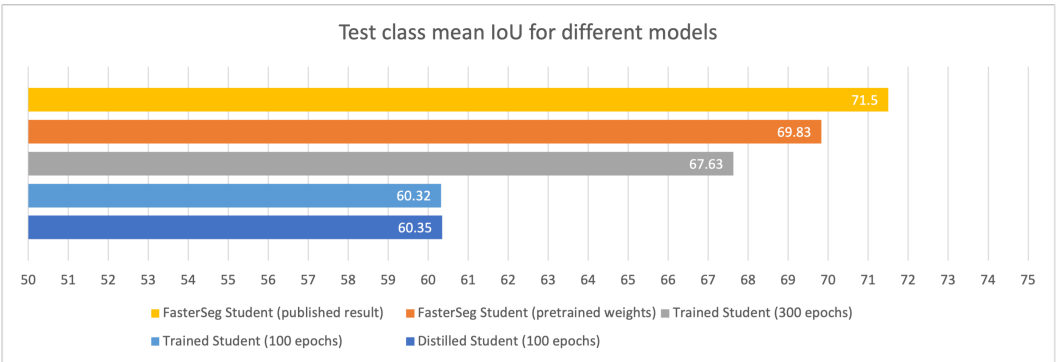

Figure 1: Test class mean IoU for different models

student in terms of validation accuracy by 4.5% and 0.7% respectively. This finding drove the idea of training the student network further to draw comparisons with the published results. After training for a further 200 epochs, the trained student was able to perform almost on par with the accuracy from the pre-trained weights with half the epochs and significantly less training time per epoch than distillation would incur. This is exacerbated by the distilled student requiring a teacher be trained whereas the trained student did not.

These models were then evaluated on an unseen test dataset and the predictions were submitted to Cityscapes for assessment. The result of the assessments are then plotted in Figure 1. One unexpected finding was a discrepancy between the published FasterSeg accuracy results and those of the same network using pre-trained weights. The margin of difference was approximately 2%. A possible explanation for this mismatch could be changes to the test dataset since the paper was published. However, no evidence of this was found and no archive of previous dataset versions were accessible.

As for the models trained as part of this project, the results are somewhat in line with those reflected by the validation accuracies. The test accuracy of both the trained and distilled students are identical to 3 significant figures at 60.3%. The student architecture that was trained for 300 epochs achieved a test accuracy of 67.7%, 2% lower than the distilled student published in the paper despite training for a small fraction of the time and half the number of epochs. Further experimentation could speak to the trained models performance in contrast to the distilled model, as this slight margin could indicate the distillation processes additional training time is unwarranted.

The disparity between the validation and test accuracy is more apparent for the trained student models, but the margins are not a cause for concern.

The values of the student models as shown in Table 2, show a vast variance in performance. The number of parameters as well as FLOPs are virtually identical for all student models however, there was a significant discrepancy in frame rate. The reason for this is the hardware on which the model was run. The hardware details of the FasterSeg paper were not included in their entirety outside of the GPU used. Hardware *B* has significantly more threads and as such, produced the highest frame rate. Hardware *A* used the same GPU as the original researchers, but the CPU must account for the 40% difference in frame rate. The frame rate of the teacher was not included in the original paper nor here, but it achieved 79.2 FPS.

Table 2: Latency results and architecture details. FasterSeg refers to the published results from Chen et al. (2019). The results were reproduced on two different set of hardware:
(**A**) Intel(R) Xeon(R) CPU E5-2623 v4 @ 2.60GHz + NVIDIA 1080 Ti
(**B**) AMD Ryzen Threadripper PRO 3975WX 32-Cores + NVIDIA 3080 Ti

	Student		
	FasterSeg	**Reproduced (A)**	**Reproduced (B)**
Params (MB)	4.4	4.4	4.4
FLOPs (GB)	28.2	27.9	27.9
FPS	163.9	96.1	195.9

5 CONCLUSION

The paper was mostly reproducible. The 'README' file on the github repository contained detailed instructions for deploying the provided code. The code was also very readable making it easier to debug. However, the code given had few comments on the vast majority of the files with exception to the configuration files. This meant it was difficult to address the errors as they arose. Furthermore, the 'requirements.txt' file containing the dependencies was outdated and incomplete, an updated version can be found in the repository Tan et al. (2022).

The paper did not give any estimate with respect to the time required to train the models, although the complexity of NAS could imply as such. Each model proposed by the paper would take upwards of a week to train, which was not realistic in our time-frame. An explicit note of this in the paper would have helped curious parties.

The results gathered for this project have all been put into the repository (Tan et al., 2022). The experiments performed largely support the findings of the paper. We were able to distill the pre-trained teacher network into a student network. The accuracy of this network was lower than the published network's, but this is due to only running for a sixth of the number of epochs. This is true for all training and distillation models. The slight discrepancy in test accuracy of the pre-trained student network was cause for concern, but it is possible the dataset may have been changed. The experiments conducted as part of this report question the need for the teacher-student distillation approach as the results indicate similar performance with less time could be achieved, but without running for 600 epochs, this can not be said definitively.

REFERENCES

Wuyang Chen, Xinyu Gong, Xianming Liu, Qian Zhang, Yuan Li, and Zhangyang Wang. Fasterseg: Searching for faster real-time semantic segmentation. *arXiv preprint arXiv:1912.10917*, 2019.

Wuyang Chen, Xinyu Gong, Xianming Liu, Qian Zhang, Yuan Li, and Zhangyang Wang. Fasterseg. `https://github.com/VITA-Group/FasterSeg`, 2021. Github repository (commit 478b026).

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 *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Jun Xian Tan, Ahmad Shabir Galili, and Adomas Lebedys. Fasterseg reproducibility results. `https://github.com/the-dropouts/fasterseg-results`, 2022. Github repository.

ACKNOWLEDGMENTS

We would like to express our thanks to Dr. Graeme Bragg, who gave us access to powerful hardware and provided invaluable help for setting up our training process.