Temporal Context Aggregation with FANet for Real-Time Image Semantic Segmentation on Video Streams

Andrew Torgesen [*1] **David Elatov** [*2] **Dayne Howard** [*2]

Abstract

This project seeks to improve the trade-off between run-time and accuracy of FANet, a state-of-the-art algorithm aimed at real-time image segmentation on video streams. The trade-off is characterized before and after augmenting FANet to include temporal context aggregation: an extension of self-attention that considers multiple consecutive image frames during both training and inference. To this end, the original FANet algorithm is re-produced and trained using the CityScapes dataset (20 classes) for validation of the base implementation. Subsequently, the temporal context aggregation augmentation to FANet is presented. The agumented FANet implementation is then trained and tested on both single-frame images and video streams from a segmentation dataset created using AirSim (11 classes). Results are presented for each scenario, comparing the segmentation speed (FPS) and performance (in mean-intersection-over-union, or mIoU %) of the re-implemented FANet with the results presented in the original paper, as well as a leader board for image segmentation on the CityScapes dataset. The presented results and analysis suggest that temporal context aggregation is not expressive enough to consistently provide performance improvements.

1. Introduction

1.1. Project Contribution

The contribution of this project is to both implement and augment a state-of-the-art image semantic segmentation algorithm to attempt to improve its trade-off between speed and accuracy on video streams, as opposed to single-frame (or disjoint) images. The augmentation entails taking a learning algorithm that is optimized for fast and accurate inference on single-frame images and expanding its capabilities to take advantage of a set of temporally consecutive images, as with video streams from an autonomous vehicle. This concept can be viewed as an extension of self-attention, and is referred to as temporal context aggregation (TCA) in this work. This project provides an analysis of the effects of including TCA, and prescribes research avenues for further investigating its usefulness for real-time segmentation on video streams.

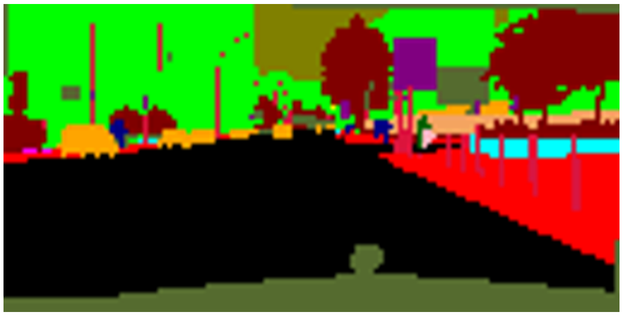

Figure 1: Rendered semantic target labels for a selected training image from the CityScapes dataset. Used to visually validate the results presented in Fig. 4.

1.2. The Speed-Accuracy Tradeoff in Image Semantic Segmentation

Image Semantic Segmentation, which allows for object-level reasoning and analysis from either disjoint images or video streams, is highly useful in the field of autonomous vehicles that rely on discrete object awareness for planning and navigation. Of the many deep learning-based image segmentation algorithms developed to-date, even the more modern ones exhibit a fundamental tradeoff between accuracy and computational efficiency.

The field of image semantic segmentation, as with many applications in image processing, was initially dominated by classical techniques like image thresholding (Sezgin & Sankur, 2004) and later learning algorithms of low compu-

[*] Equal contribution [1] Department of Aeronautics and Astronautics [2] Department of Mechanical Engineering. Correspondence to: David Elatov <elatov@mit.edu>, Dayne Howard <dayneh@mit.edu>, Andrew Torgesen <torgesen@mit.edu>.

Proceedings of the 38th International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

tational complexity such as random forest that could train on a CPU (Schroff et al., 2008). However, the segmentation accuracy gains from works employing deep neural convolutional networks such as (Simonyan & Zisserman, 2015) have led to a proliferation of deep learning-based methods for segmenting images (Hao et al., 2020). Since then, many of the now myriad of deep convolutional network models are devised for segmenting single-frame images (Chen et al., 2018; 2016; Takikawa et al., 2019; Wu et al., 2019; He et al., 2017; Ronneberger et al., 2015; Minaee et al., 2020). Moreover, the majority of segmentation methods have no performance criterion for either temporal consistency across multiple frames depicting the same scene or real-time prediction speeds (Li et al., 2018)–both of which are desirable for usage onboard an autonomous vehicle processing video streams.

Of the methods that address both of the issues stated above, there is a class that focuses on adding additional components such as LSTMs to analyze image sequences (Pfeuffer et al., 2019) at the expense of higher computational costs. The remaining class of methods attempts to recycle features across frames to reduce computational complexity in the feature encoder part of the learning algorithm (Zhu et al., 2017) without addressing overall latency arising from feature decoding. Recent works such as (Li et al., 2018) have illustrated the insufficiency of these two classes of algorithms to handle applications where overall latency and segmentation accuracy are equally important. Thus, at the frontier of image segmentation techniques for autonomous vehicles are methods that try to overcome this trade-off between accuracy and efficiency, while also reducing the overall latency (Hafiz & Bhat, 2020).

1.3. Related Work: FANet

One of the most recent works attempting to overcome the accuracy-efficiency trade-off in segmentation is called the Fast Attention Network (FANet) for real-time semantic segmentation (Hu et al., 2021). FANet reportedly improves the trade-off by attempting to capture crucial high- and low-resolution contextual information in successive images without needing the same network depth generally required for segmentation. This method differs from a previously developed method, self-attention, in its calculation of the “attention module," whose function is to capture this aggregate spatial context in a memory-efficient way. The authors of FANet modify the activation functions and matrix multiplication strategies of the attention module to create the novel “fast attention module," which is able to capture spatial context while also requiring less computation time.

The method boasts a 50% speed increase over the current state-of-the-art (Siam et al., 2018) while retaining the same level of accuracy as measured by mean Intersection-over-Union (mIoU), which is a popular performance metric for semantic segmentation (Jadon, 2020). Performance is demonstrated on multiple datasets, including a 75.5% mIoU at 58 FPS on the the CityScapes dataset (Cordts et al., 2016), which provides real-world video streams and segmentation information for 20 distinct classes in urban street scenes and is popular for testing autonomous driving-oriented applications (Minaee et al., 2020).

The authors of FANet claim that the addition of TCA has the potential to further boost segmentation accuracy by 0.5% without compromising inference speed.

2. Methods

2.1. FANet Implementation and Augmentation with Temporal Context Aggregation

FANet is chosen as a representative state-of-the-art image segmentation algorithm to augment with TCA because of its favorable accuracy-speed trade-off and utilization of self-attention, albeit in a slightly modified form.

Figure 2 provides a visual representation of the FANet image segmentation pipeline, as well as the proposed modifications to include temporal context aggregation within the fast attention module. As is shown in Fig. 2a, the input image is encoded into features by a pre-trained instance of ResNet (He et al., 2015), and those features are extracted at four different resolutions before being fed into four corresponding fast attention modules (Fig. 2b) and subsequently combined through up-sampling (Fig. 2c) into the segmented image output. For this project, ResNet-18 is selected as the feature encoder due to the FANet authors' insight that it provides a desirable balance between encoding capability and computation speed.

The proposed modification to FANet is illustrated in Fig. 2b–specifically in the stacked matrix multiplication blocks, which represent the capturing of the fast attention module's calculated *key* (K) and *value* (V) matrices for previously processed frames at times $T-1, \cdots, T-N$ in memory. The stored key and value matrices for the previous N frames are then summed together and used to help classify the current image frame at time T through multiplication with the current frame's *query* (Q) matrix.

Thus, just as the fast attention module's use of K, V, and Q for each image allows for *spatial* context aggregation to assist inference at various levels of resolution, the storage and sliding window sum of past K, V matrices affords *temporal* context aggregation. Assuming that the training, validation, and testing set images are fed into the network in chronological order (e.g., with a video stream), the inclusion of the proposed temporal context aggregation scheme is designed to improve segmentation performance without significant

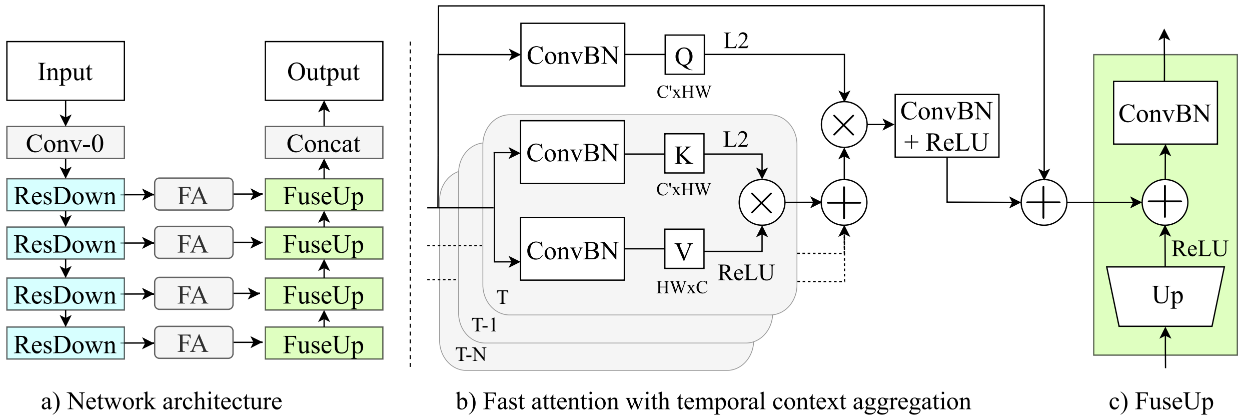

Figure 2: Augmented FANet learning algorithm with temporal context aggregation. Figure modified from (Hu et al., 2021) to show how temporal context aggregation can be introduced into the fast attention module implementation.

increase to the required computation time.

2.2. Video Stream Dataset Generation

Whereas the authors of FANet demonstrate the network's performance on disjoint images from CityScapes, an effective demonstration of TCA's performance-enhancing capabilities requires training and testing on high-rate video streams, where the difference between consecutive images is small. Moreover, each frame in the video stream should have corresponding truth labels for both training and testing. To fulfill these needs, a custom segmented video stream dataset is generated using the AirSim (Shah et al., 2017) photorealistic simulation environment.

AirSim allows for camera image dataset generation by maneuvering a simulated autonomous vehicle, such as a car or a UAV, with an attached camera and logging images from the camera's video stream. An API is also provided for logging segmentation labels; however, these labels correspond to very granular building blocks used to create each AirSim environment, resulting in a large number of classes (250+) that aren't guaranteed to be consistent across different environments.

To amend the AirSim segmentation labels issue, the collected image and segmentation data are manually processed to consolidate all labels into a consistent and concise set of 11 classes: road, vehicle, vegetation, tree, traffic fixture, sky, fence, stone, house, pool, and roof.

An example image with processed true segmentation labels is provided by Fig. 3. In total, the generated AirSim dataset provides 18 separate videos, each a few minutes in length, with 640 × 480 images at 25 Hz.

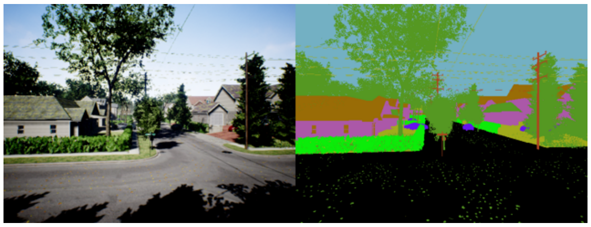

Figure 3: Sample image and segmentation labels from the generated AirSim dataset.

2.3. Training and Inference

2.3.1. On Single-Frame Images

For the first step in evaluation, the FANet algorithm is implemented without TCA and trained on the CityScapes dataset, which provides 20 segmented classes in disjoint images, in order to validate the base implementation by comparing testing performance to that of the original authors.

In keeping with the reported training process from the authors of FANet, the image training set consists of $N_b = 186$ batches of $N_{\mathcal{I}} = 16$ images each, for a total training set size of 2,976 images with corresponding true classification labels for the $N_C = 20$ CityScapes classes. The network is trained for 30 epochs.

An additional validation set with $N_b = 32$, $N_{\mathcal{I}} = 16$ images is set aside for performance and generalization testing.

For each (normalized) input RGB image of size $3 \times w \times h$, the network outputs a down-sampled output classification tensor of size $N_C \times w/8 \times h/8$, representing the classification probabilities among the $N_C = 20$ CityScapes classes for each pixel. To evaluate the disparity between the target and output classifications for each training image, a cross entropy loss metric is computed across all pixels and classes between the output image classification tensor $\mathcal{I}_{\text{out}}$ and the target image classification tensor $\mathcal{I}_{\text{targ}}$. To enable one-to-one cross entropy comparison, the target image is down-sampled to one-eighth the original resolution, matching the resolution of the network output.

The total mIoU accuracy metric $\%_{\text{mIoU,tot}}$ is calculated for each epoch as

$$\%_{\text{mIoU,tot}} = \frac{1}{N_b} \sum_{i=1}^{N_b} \frac{1}{N_{\mathcal{I},i} N_C} \sum_{j=1}^{N_{\mathcal{I},i}} \sum_{k=1}^{N_C} \%_{\text{mIoU},j}\,, \qquad (1)$$

$$\%_{\text{mIoU},j} = \frac{\sum_{p\in\mathcal{I}_j} \mathcal{I}_{\text{out},j} \odot \mathcal{I}_{\text{targ},j}}{\sum_{p\in\mathcal{I}_j} \mathcal{I}_{\text{out},j} + \mathcal{I}_{\text{targ},j} - (\mathcal{I}_{\text{out},j} \odot \mathcal{I}_{\text{targ},j})}, \qquad (2)$$

where $\sum_{p\in\mathcal{I}}$ denotes the sum over all pixels p in image $\mathcal{I}$. Effectively, $\%_{\text{mIoU,tot}}$ represents the average mIoU over all images, classes, and batches in the training iteration. Equation 1 is consistent with the definition of mIoU most widely used in computer vision applications, including the FANet work.

2.3.2. On Video Streams

To evaluate FANet's performance with the TCA augmentation, FANet is trained on the video streams from the AirSim dataset, with and without TCA. The training process on video streams is largely the same as what is described in Sec. 2.3.1. That said, the inclusion of TCA necessitates an important modification of the training process due to how gradients are calculated for backpropagation.

Without TCA, the segmentation maps of each batch are calculated simultaneously and independently. With TCA, the fast attention modules must include the proper K, V history for $T, \cdots, T-N$ (see Fig. 2). To accomplish this, each batch is first ensured to have temporally sequential input frames. During the forward propagation of each fast attention module, the key and value matrices for every frame in the batch are calculated independently of previous time frames. Then, all frames after the first N frames are aggregated with their respective, previous K, V matrices. Importantly, the gradients for backward propagation are calculated excluding the effects of the loss value from the first N frames of the batch, since they did not have the proper K, V history available during computation. In effect, this preserves the influence of the first N key and value matrices on future fast attention modules for back propagation. The advantage of training in this way is that no additional computer memory is required and the additional computation time is minimal, since all other batch computations can be done simultaneously.

Table 1: Accuracy and speed metrics for select state-of-the-art semantic segmentation algorithms on the CityScapes validation image set. For consistent comparison, the validation set is derived from a publicly available CityScapes evaluation pipeline. †Computed on a different GPU.

Method	mIoU %	FPS
FANet (ours)	73.4	50.0
FANet (Hu et al., 2021)	75.0†	72.0†
ENet (Siam et al., 2018)	58.3	62.6
ShuffleSeg (Gamal et al., 2018)	58.3	120.0
Netwarp (Gadde et al., 2017)	80.6	0.33

Furthermore, there are multiple ways to formulate the training batches when seeking to specialize performance for video streams. One way is to have each batch consist entirely of consecutive images from the same video sequence to ensure that the temporal adjacency assumption between gradient calculations holds. This simplistic batch formulation will be referred to as homogeneous batching. An alternative approach is to have each batch consist of smaller "mini-batches," where each mini-batch consists of consecutive frames from the same video. By including different mini-batches within each training batch, the degree of variety in the training samples is increased. This batch formulation will be referred to as mixed batching. Mixed batching requires that care be taken to ensure that gradients are only retained through TCA within mini-batches, and not across them, as distinct mini-batches correspond to distinct video streams. TCA-related results are presented featuring both batch formulation methods.

3. Results

3.1. CityScapes Dataset

Figures 4-5 give the mIoU training curves the FANet implementation with accompanying image segmentations for qualitative analysis. From the plots, it is shown that by 30 training epochs, the network achieves a comparable mIoU to the original authors' reported validation mIoU of 75.0%. The training curves and reference images instill confidence that the baseline implementation of FANet is correct, and thus suitable as a benchmark comparison for the temporal context aggregation addendum's performance.

Table 1 presents the measured speed of our FANet implementation alongside the final validation set mIoU accuracy. The table also places these results in a larger context, comparing with FANet's original reported performance numbers (obtained on a more powerful GPU) as well as various other (pre-trained) high-performing, state-of-the-art image segmentation algorithms. This larger comparison illustrates FANet's positioning as a favorable trade off between accuracy and speed in the current image segmentation landscape. It remains to be shown that FANet's accuracy can be further improved through temporal context aggregation *without* taking a considerable hit to inference speed.

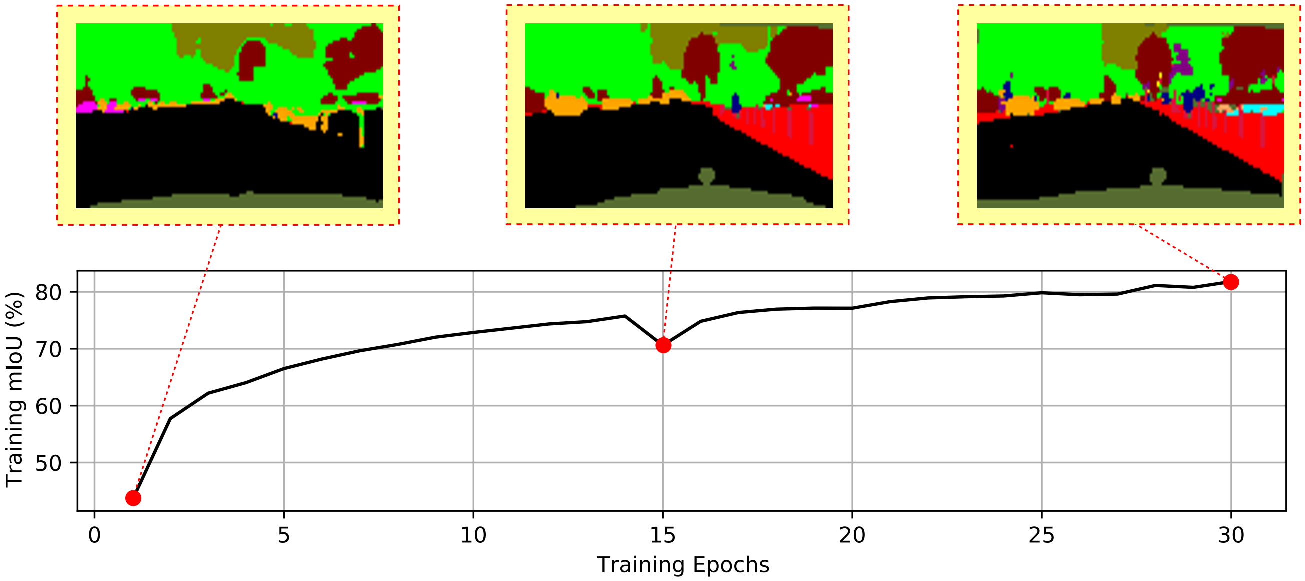

Figure 4: mIoU performance of FANet on the formulated training set as a function of training epoch. Visual samples of FANet's segmentation label outputs are given at various epochs to demonstrate increasing performance. The corresponding true segmentation labels are given in Fig. 1.

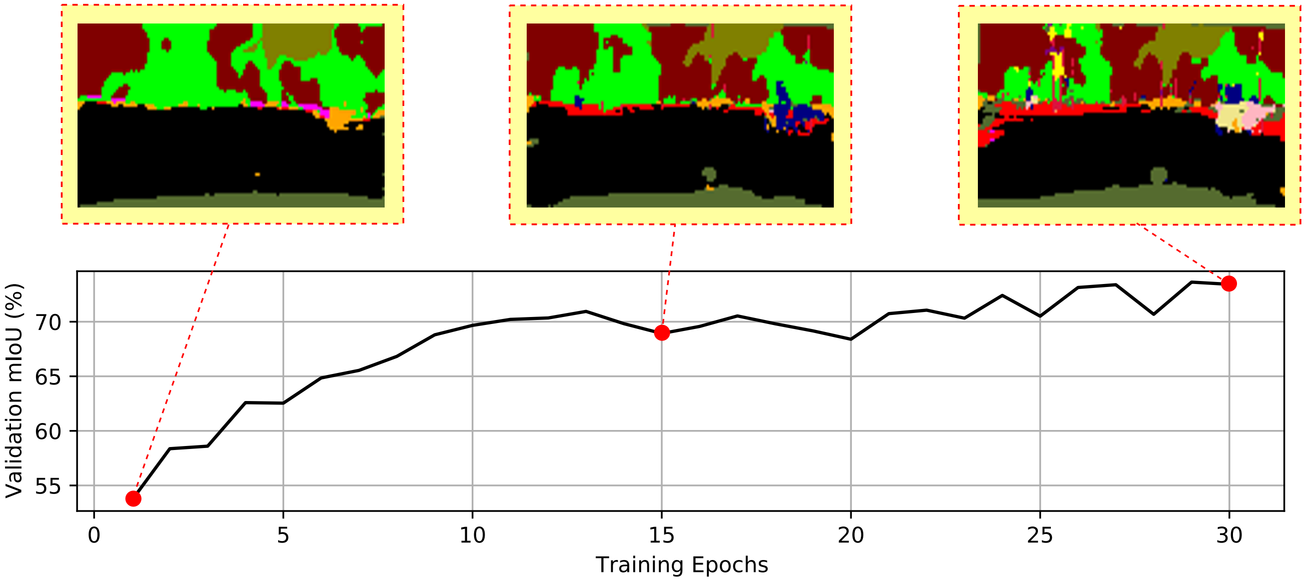

Figure 5: mIoU performance of FANet on the formulated validation set as a function of training epoch. Examination of the provided sample outputs at select epochs reveals a similar level of fidelity improvement across time compared to that of the training set outputs shown in Fig. 4.

3.2. AirSim Dataset

To ensure that neither the structure of the simulated dataset nor the TCA augmentation has not broken FANet's functionality and performance, experiments on AirSim dataset images are presented with a TCA depth of $N = 0$, effectively replicating the training scenario from Sec. 3.1 but using video streams and homogeneous/mixed batching. Fig. 7 shows the training curves for homogeneous versus mixed batching. From the figure, it is immediately apparent that homogeneous batching leads to unstable training performance. This is attributable to the relatively small amount of variation across consecutive video frames, which conceptually is similar to drastically reducing the effective batch size. Mixed batching appears to amend this problem nicely, while still preserving the temporal adjacency of image frames re-

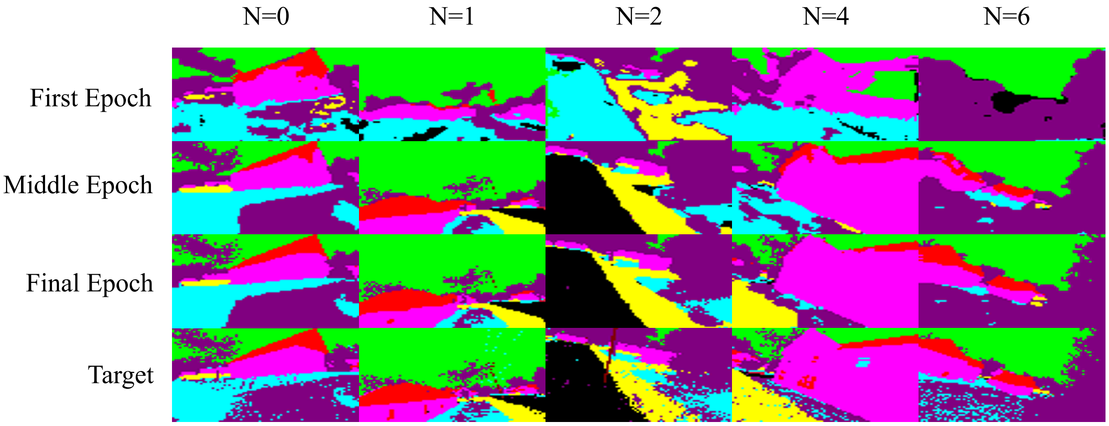

Figure 6: Visualization of the segmentation performance across different depths N of TCA. mIoU accuracy shown to improve stably as the training process progresses.

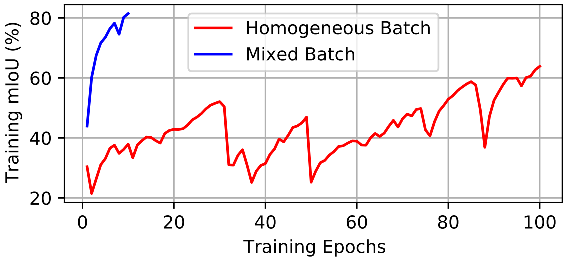

Figure 7: mIoU performance comparison of FANet with TCA depth $N = 0$ as a function of training epoch with homogeneous versus mixed batches. Mixed training batches incorporate images across different video streams while still isolating the TCA-related gradient terms.

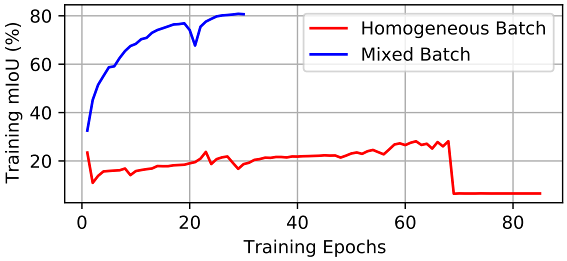

Figure 8: mIoU performance comparison of FANet with TCA depth $N = 4$ as a function of training epoch with homogeneous versus mixed batches.

Table 2: Generalizability of the mIoU performance for different depths N of TCA. Inference speed found to be consistent across all configurations.

	$N = 0$	$N = 2$	$N = 4$	$N = 6$
Train mIoU (%)	81.5	79.6	80.7	78.7
Val. mIoU (%)	80.2	78.0	77.3	72.2
Test mIoU (%)	80.4	78.2	76.2	72.9
Speed (FPS)	50.0	50.0	50.0	50.0

quired for the gradient calculations with TCA. Additionally, with mixed batching, the mIoU performance exceeds that of the CityScapes training process by several percentage points and over less epochs because of the idealized rendering conditions from a simulated–albeit realistic–environment.

The homogeneous versus mixed batching comparison experiment is replicated with a non-zero ($N = 4$) TCA depth, and the results are shown in Fig. 8. The same phenomenon of homogeneous batching instability is observed, but this time it is more drastic. The sudden mIoU dropoff at ≈ 70 epochs is interesting to consider as a symptom of training sample impoverishment leading to an over-fitted model. At $N = 4$, mixed batching still provides a stable training curve, though this time requiring more epochs to achieve the same level of performance.

Figure 9 compares the training curves for increasing TCA depths. The trend uniformly suggests that larger TCA depths prolong the training process without offering significant mIoU performance gains in the limit. Figure 6 gives a visual representation of segmentation performance across TCA depths. Visually, it is difficult to distinguish between performance levels by the time training is complete for each formulation. Further context is provided by Table 2, which suggests that while mIoU performance generalizes well and requires no significant computational increase across all TCA depths, mIoU performance steadily drops, rather than increases, with increasing depth.

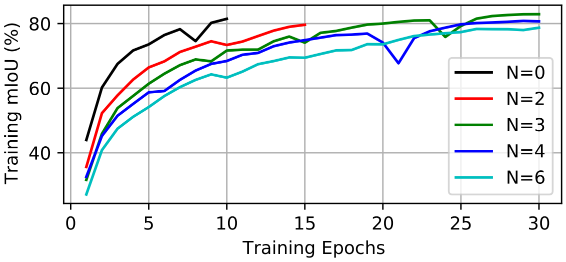

Figure 9: mIoU performance comparison of FANet with different TCA depths as a function of training epoch with mixed batches.

4. Conclusions

The training performance comparison between homogeneous and mixed batching suggests a potential hypothesis for why including TCA in FANet has either negligible, or slightly detrimental, effects. Videos taken from a car or drone moving through an environment produce images that evolve at various rates due to the speed and rotation of the camera, depth to objects, frame rate, etc. With no more than simple addition of past frames' key and value matrices, it is likely difficult for the network to learn weights which will produce matrices that are robust to all situations.

For a more exhaustive analysis into the potential benefits of TCA to a segmentation algorithm like FANet, additional experiments can be performed that attempt to isolate performance benefits or detriments between the training and inference processes, respectively.

Given the high frame rate at which FANet can operate, inclusion of just one previous frame's key and value matrices (i.e., TCA depth of $N = 1$), as the original FANet authors report in their results, may be nearly equivalent to feeding the network two of the same frame. The FANet authors report a potential 0.5% point increase in mIoU performance by incorporating TCA. The statistics of producing this increased performance are not presented in their work, however. The results in this study suggest that the reported performance increase may have been the result of a carefully curated dataset, or even, possibly, the result of chance.

Working GitHub repository with PyTorch implementation:

https://github.com/goromal/FANet_Evaluation

References

Chen, L., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *CoRR*, abs/1606.00915, 2016. URL http://arxiv.org/abs/1606.00915.

Chen, Y., Li, W., Chen, X., and Gool, L. V. Learning semantic segmentation from synthetic data: A geometrically guided input-output adaptation approach. *CoRR*, abs/1812.05040, 2018. URL http://arxiv.org/abs/1812.05040.

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

Gadde, R., Jampani, V., and Gehler, P. V. Semantic video cnns through representation warping. *CoRR*, abs/1708.03088, 2017. URL http://arxiv.org/abs/1708.03088.

Gamal, M., Siam, M., and Abdel-Razek, M. Shuffleseg: Real-time semantic segmentation network. *CoRR*, abs/1803.03816, 2018. URL http://arxiv.org/abs/1803.03816.

Hafiz, A. M. and Bhat, G. M. A survey on instance segmentation: state of the art. *International Journal of Multimedia Information Retrieval*, 9(3):171–189, July 2020. doi: 10.1007/s13735-020-00195-x. URL https://doi.org/10.1007/s13735-020-00195-x.

Hao, S., Zhou, Y., and Guo, Y. A brief survey on semantic segmentation with deep learning. *Neurocomputing*, 406:302–321, 2020. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2019.11.118. URL https://www.sciencedirect.com/science/article/pii/S0925231220305476.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL http://arxiv.org/abs/1512.03385.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. B. Mask R-CNN. *CoRR*, abs/1703.06870, 2017. URL http://arxiv.org/abs/1703.06870.

Hu, P., Perazzi, F., Heilbron, F. C., Wang, O., Lin, Z., Saenko, K., and Sclaroff, S. Real-time semantic segmentation with fast attention. *IEEE Robotics and Automation Letters*, 6(1):263–270, 2021. doi: 10.1109/LRA.2020.3039744.

Jadon, S. A survey of loss functions for semantic segmentation. *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, Oct 2020. doi: 10.1109/cibcb48159.2020.9277638. URL http://dx.doi.org/10.1109/CIBCB48159.2020.9277638.

Li, Y., Shi, J., and Lin, D. Low-latency video semantic segmentation. *CoRR*, abs/1804.00389, 2018. URL http://arxiv.org/abs/1804.00389.

Minaee, S., Boykov, Y., Porikli, F., Plaza, A., Kehtarnavaz, N., and Terzopoulos, D. Image segmentation using deep learning: A survey, 2020.

Pfeuffer, A., Schulz, K., and Dietmayer, K. Semantic segmentation of video sequences with convolutional lstms. *CoRR*, abs/1905.01058, 2019. URL http://arxiv.org/abs/1905.01058.

Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015. URL http://arxiv.org/abs/1505.04597.

Schroff, F., Criminisi, A., and Zisserman, A. Object class segmentation using random forests. 01 2008. doi: 10.5244/C.22.54.

Sezgin, M. and Sankur, B. Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*, 13(1):146 – 165, 2004. doi: 10.1117/1.1631315. URL https://doi.org/10.1117/1.1631315.

Shah, S., Dey, D., Lovett, C., and Kapoor, A. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and Service Robotics*, 2017. URL https://arxiv.org/abs/1705.05065.

Siam, M., Gamal, M., Abdel-Razek, M., Yogamani, S., Jagersand, M., and Zhang, H. A comparative study of real-time semantic segmentation for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition, 2015.

Takikawa, T., Acuna, D., Jampani, V., and Fidler, S. Gated-scnn: Gated shape cnns for semantic segmentation. *CoRR*, abs/1907.05740, 2019. URL http://arxiv.org/abs/1907.05740.

Wu, H., Zhang, J., Huang, K., Liang, K., and Yu, Y. Fastfcn: Rethinking dilated convolution in the backbone for semantic segmentation. *CoRR*, abs/1903.11816, 2019. URL http://arxiv.org/abs/1903.11816.

Zhu, X., Xiong, Y., Dai, J., Yuan, L., and Wei, Y. Deep feature flow for video recognition, 2017.