Research and Implementation of Video Semantic Segmentation, with Possible Improvements

William Zhang, Muyang Xu, Qianyu Zhu, Star Chen

Oct 2021

1 Introduction

This summer, we mainly focused on learning core concepts of several basic models for semi-supervised video semantic segmentation, and trying new twists based on the pre-existing methods. Our Research can be divided into three stages: in the first stage, we spend a month reading past papers and online tutorials to learn the basic structure of image semantic segmentation and video semantic segmentation, like ResNet structures and fully connected networks.

In the second stage, taking a deeper step in research and implementation of video semantic segmentation, with possible improvements, we referred to the temporal memory attention for video semantic segmentation (TMANet). We intended to increase the accuracy and comparatively improve the computational efficiency of our model based on Camvid and Cityscape—those two datasets. Here are several attempt directions we have accomplished during the DURF program:

- Changing the backbones of the model.
- Adjusting different attention blocks directly at decoding layers.
- Inserting attention block stages among different convolutional layers.
  - non-local attention, pyramid scene parsing strategy, temporal memory attention, crisscross attention, and dual attention
- Reshaping the memory length (the number of consecutive frames we passed into the training model).

We set up our coding environment in the start-up stage and tested our experiments under the GPU acceleration through NYU Shanghai High-Performance Center. To construct an abstract video semantic segmentation model faster, we introduced MMsegmentation toolbox based on Pytorch during our model building. We concentrated our experimental directions on different collocations of model construction blocks in MMsegmentation—backbones, decoding head, and loss functions.

In terms of backbones, we converted TMANet's original ResNet 50 structure into ResNet 18 structure, which intends to reduce the model complexity. In terms of decoding head, we drastically paid attention to enhance our model's training performances. By comparing TMANet's original non-local attention block, we took multiple attempts, including dual attention block, criss-cross attention block, dual attention block, asymmetric attention block, and pyramid scene parsing strategy. The training results show that some improvements in predicting accuracy have been achieved with respect to different types of attention blocks. In terms of the advanced-level attempt, we intend to insert knowledge distillation to optimize our loss functions. We have analyzed the specific factors that boost our predicting accuracy by comparing each attempt with the original training model structure.

2 Asymmetric Non-local Attention

1. Asymmetric Pyramid Non-local Block (APNB)

   In video semantic segmentation, both latencies of response (i.e., computation time) and accuracy are key metrics to measure the performance of a model. TMANet has got state-of-the-art accuracy and is referred to as the benchmark in both Cityscapes and CamVid datasets. However, we identified that, due to the naive temporal memory block and the simple non-local attention in the temporal dimension used in TMANet, large cache and computation power are required to both train and test the model. Therefore, we decided to find and implement replacements for the temporal non-local attention in TMANet, reducing the computation cost and improving the latency issue.

   Asymmetric Pyramid Non-local Block (APNB) is a classic improvement on non-local attention block, which adds a spatial pyramid pooling module after both key and value channel after 1x1 convolution (embedding). The pooling results are then flattened and concatenated to serve as the input to the next layer. Intuitively, such a pyramid pooling module should capture enough spatial information to maintain accuracy (or not drop too much) while reducing the time complexity of the non-local attention.

   So, we expect that by substituting the non-local attention in TMANet with APNB, the high latency of TMANet could be lessened while still keeping the state-of-the-art accuracy. The only issue is to transform APNB, which is initially used for image semantic segmentation, to the video input. In our implementation, the extra temporal layer T is also spatially pooled by changing the original 2D square pooling to 3D cube pooling. Now, matrix multiplication for calculating the attention will be $B \times (H \times W) \times S$, instead of $(B \times H) \times (W \times T) \times (H \times W)$, (S is the result of pyramid spatial pooling and $S << T \times H \times W$).

   Indeed, as our result shows, with every other model and training parameter being equal, the accuracy of the original model (TMANet) is 0.7541.

In contrast, the modified model (APNB - TMANet) has an accuracy of 0.7528, only a 0.17% of accuracy drop. As for the computational time, on the same CamVid dataset, the training time for TMANet is 14:42:33 ($HH : MM : SS$), and the training time for APNB - TMANet is 13:51:51, a 5.744% performance improvement. Therefore, we conclude that by substituting the naive temporal non-local attention with our modified temporal APNB, we successfully reduce the latency of TMANet, while still keeping the state-of-the-art accuracy.

2. Asymmetric Fusion Non-local Block (AFNB), APNB + AFNB

   In the paper "Asymmetric Non-local Neural Networks for Semantic Segmentation," another modification to the non-local attention block aside from APNB is the Asymmetric Fusion Non-local Block (AFNB). In the paper, while APNB already outperforms the non-local attention block in single block efficiency, a combination of AFNB and APNB also shows superior effectiveness in the model level efficiency. AFNB takes two input sources: a high-level feature map (stage 5 output of the ResNet backbone) and a low-level feature map (stage 4 outputs of the ResNet backbone). It uses pyramid pooling before calculating the attention between the feature maps, then concatenates the result with the original stage 5 output. Intuitively, because a standard non-local block only has one input source, the AFNB captures information and pixel correlation in long-range cues from different feature levels.

   In our video semantic segmentation case, attention from different feature levels (AFNB) is implemented on only the current input image (i.e., the current frame). It is worth considering to include AFNB for memory frames, but that implies a repetition of computation and will lead to increased latency. We keep it simple here (also because AFNB on the current frame is already improving enough).

   As our result shows, AFNB+APNB+TMANet (Asym-TMANet) has an accuracy of 0.7618, the best we have yet to possess on the backbone of ResNet18. The accuracy increased by 1.02% compared to TMANet, increasing computational time to 18:20:18, 24.67% increase. We are making a tradeoff between latency and accuracy. We concluded that the cooperation of AFNB and APNB works well since we would expect if only using AFNB, then the computational time would be even higher. It also gives more space for reducing the memory length in later experiments, as we will show in the following sections.

3 Criss-Cross Attention

While attention modules significantly enhanced the performance of the backbone model, a more well-behaved attention algorithm usually requires more intensive computation and storage. (In Dual Attention, Spatio-temporal information is considered concurrently, and in TMANet, each pixel is compared with

every single pixel in a time frame is referenced to compute attention). Aiming at cutting down computation while still preserving decent accuracy, researchers introduced a novel module: criss-cross attention. With its help, each pixel is compared only with the pixels in the horizontal and vertical path. By taking a further recurrent operation, each pixel can finally harvest the contextual information of all the pixels throughout the criss-cross grid, establishing the full-image dependencies.

Our team re-implemented the CCNet with ResNet18 as the backbone and arranged different trials by modifying the number of memory frames considered, the time of criss-cross attention, the dimensions attention is applied, and the way of computing query, key, and value. When fewer memory frames are considered, more attention times yield better results, and comparatively speaking, when more memory frames are included, less attention is needed for the model to achieve the best results. In general, the model's behavior first increased, then decreased with the time of pixel-to-pixel attention. Besides, the original research team also introduced a 3-D cc attention version, adding the time dimension. However, the redundant calculation of this structure is too large (we computed the attention for every pixel, including the current frame and all memory!), and it takes 18 hours for training on HPC. We revised the original code by separating space dimensions from time, putting more weight on the current frame, and using memory only as supporting data. That improved the accuracy slightly to 0.7567 while keeping training time in 9 hours.

There is a flaw of criss-cross attention: diagonal pixels are indirectly obtained, thus not robust enough.

4 Dual Attention

1. Dual Attention (DA)

   Dual Attention Network (DAN), as presented in "Dual Attention Network for Scene Segmentation," incorporates position attention module (PA) channel attention module (CA) and sums the output from each module. PA is more or less the same as the non-local attention block. At the same time, CA makes an innovation to selectively emphasize interdependent channel maps by integrating associated features among all channel maps. It is a relatively computational heavy module as the calculation in attention essentially doubles. Therefore, we would like to implement it to significantly and solely increase the accuracy while not being constrained by latency.

   In our implementation, PA is substituted by the temporal non-local attention in TMANet, and CA is calculated only on the current image, with the code same as the original approach in the DANet paper. This approach of CA is similar to what we did with Asym-TMANet, specifically, AFNB. Similar arguments could also be drawn here, as we could include CA for memory frames. Our current approach avoids the repetition of computa-

tion and even more increased latency. But of course, we have already got rid of the latency constraint, so it would be worth experimenting. Still, we keep it simple here, and as the result shows, incorporating DA in our model does not increase the accuracy.

The modified model (DA-TMANet) on the exact other parameters trained for 16:08:58 and reaches an accuracy of only 0.7528, identical to the APNB-TMANet, which has far less training time (less latency). The result is counter-intuitive because DANet has a significant increase of performance on image semantic segmentation, but in our video case, the performance even drops. According to the empirical results, we conclude that the unique temporal information in video semantic segmentation already suffices to capture the general information in the frames. With the temporal attention already capable, channel-wise attention is unnecessary, and the two attention actually neutralizes the potential accuracy improvement.

2. DA + APNB + AFNB

   With the above DA-TMANet experiment failing in improving the accuracy, we still want to test our conclusion. Thus, with previous success in using APNB + AFNB in TMANet, we try to combine APNB + AFNB and DA:

   (a) use the AFNB output as the input of the DA module implemented above

   (b) substitute the PA in DA with previously implemented temporal APNB)

   The resulting model accuracy is 0.7463, even lower than the previous model. After double-checking the correctness of code, we concluded that the temporal attention and channel-wise attention did indeed neutralize the potential accuracy improvement of each other.

5 Other Strategies

1. Image Model

   Practically, the only difference between image semantic segmentation and video semantic segmentation lies in video's additional temporal dimension. Therefore, to see the effect of improving accuracy using computational heavy temporal memory and temporal attention, we also trained and tested image models with the same parameters. Specifically, because temporal attention is non-local attention, we run ResNet18 with non-local attention as the decoding head to compare the effect with TMANet. The non-local image model yields a 0.7519 accuracy. That indicates an only 0.29% improvement by implementing temporal attention. In the meantime, ResNet18 with Pyramid Scene Parsing module (PSP) yields 0.7496. Therefore, we can see the non-local attention's capability with such a simple calculation. Moreover, though the improvement is mild, TMANet is

still state-of-the-art in video semantic segmentation. And since we are using ResNet18 instead of ResNet50 or, even better, ResNet101, the improvement gap will be expected to be more significant.

2. Memory Length Include Current Frame

   While temporal information is the key for video semantic segmentation, it is worth experimenting on how many past frames to include when calculating temporal attention. As the TMANet approach does not have a current frame in the memory, we tested whether it is necessary to include it. By including the current frame, we compute the attention between the current frame and past frames and calculate the current frame's self-attention.

   Our default approach is using a memory length of 4 without including the current frame. We experiment APNB + TMANet with:

   (a) memory length of 4 and include current frame

   (b) memory length of 3 and include current frame (which maintains the overall frames as four)

   For (a), the accuracy is 0.7541, but the training time is 1-03:20:40 ($D - HH : MM : SS$). The increased accuracy compared to default APNB + TMANet is well-expected, as more spatial and temporal information is captured. Still, the drastic increase in latency indicates an unnecessary adoption of such an approach.

   For (b), the accuracy is 0.7478, and the training time is 18:15:28. Compared to the default APNB + TMANet with accuracy 0.7528 and 13:51:51. Both accuracy and latency drop by including the current frame, though the theoretical memory length is still the same. We conclude that by using temporal information, the self-attention of the current frame is not significantly influential to the outcome because the current frame and past frames will probably have minor changes.

   When testing the memory length, we discovered an interesting thing: when decreasing the memory length of default APNB + TMANet from 4 down to 2, the accuracy increases from 0.7528 to 0.76. While it is clearly stated in the TMANet paper that a temporal memory length of 4 yields the most efficient and accurate result, and our APNB + TMANet modification has nothing changed to the temporal information, we conclude that this is the result of the difference between backbones (ResNet18 and ResNet50). Because ResNet18 cannot capture the deep features of each frame compared to ResNet50, the frames in the increased memory are not contributing enough in-depth information to the prediction of the current frame.

   Moreover, because our Asym-TMANet performs the best in accuracy, we want to try to decrease the memory length to see if we could get less latency and possibly more accuracy (aligning with the previous discovery). Memory length of 3 gives 0.7531 accuracies and train time of 12:52:49;

memory length 0.7539 accuracy and train time of 10:03:19. The train time drops expectedly, but the influence of decreased memory length on accuracy does not align with the previous discovery. Therefore, we draw from the previous conclusion and deduce that the AFNB in Asym-TMANet acts to more affluent the information captured in the backbone, making a ResNet18 gain on ResNet50.

3. Knowledge Distillation

   The models we have implemented are complex in both backbone and attention modules, and even the lightest one(ResNet18 with CC attention) needs HPC support. So far, our team's research has been done remotely by submitting jobs to HPC, but our goal is to achieve instant segmentation with low latency on a laptop. Therefore, we applied knowledge distillation to compress the model.

   We first train an extensive, well-structured, cumbersome model, ResNet101 with non-local attention, as the "teacher model," and transfer useful feature information to a smaller, distilled "student model." The loss function for the distilled model consists of two parts:

   (a) cross-entropy of output data between small model and large model (to ensure that the results of the small model and large model are as consistent as possible)

   (b) cross-entropy of challenging targets and small model output data (to ensure that small model results are as consistent as possible with actual category tags)

   Though the final distillation process has not been finished, we expect the distilled model to be simplified, robust, and strong generalization ability.

References

[1] Huang, Z., Wang, X., Huang, L., Huang, C., Wei, Y., and Liu, W. *CCNet: Criss-cross attention for semantic segmentation* (2019), IEEE/CVF International Conference on Computer Vision (ICCV), https://doi.org/10.1109/iccv.2019.00069.

[2] Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z., Lu, H *Dual attention network for scene segmentationn* (2019), 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), https://doi.org/10.1109/cvpr.2019.00326.

[3] Wang, X., Girshick, R., Gupta, A., He, K *Non-local Neural Networks* (2018), 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, https://doi.org/10.1109/cvpr.2018.00813.

[4] Wang, H., Wang, W., Liu, J *Temporal memory attention for video semantic segmentation* (2021), 2021 IEEE International Conference on Image Processing (ICIP), https://doi.org/10.1109/icip42928.2021.9506731.

[5] Hinton, G. E., Vinyals, O., Dean, J *Distilling the knowledge in a neural network: Semantic scholar. undefined* (1970), Retrieved October 28, 2021, from https://www.semanticscholar.org/paper/Distilling-the-Knowledge-in-a-Neural-Network-Hinton-Vinyals/0c908739fbff75f03469d13d4a1a07de3414ee19.

[6] Gabriel J. Brostow and Jamie Shotton and Julien Fauqueur and Roberto Cipolla *Segmentation and Recognition Using Structure from Motion Point Clouds* (2008) ECCV (1).

[7] 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* (2016) Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, arXiv:1604.01685.