Sapienza Università di Roma

Advanced Machine Learning

Criss Cross Segformer

Students

Riccardo Agabiti
2027220
Andrei Caraman
1664744
Simone Piperno
1792917

Professor

Prof. Fabio Galasso

Accademic Year 2021/2022

1 Abstract

We introduce here the Criss-Cross Segformer, a variant to the state of the art for semantic segmentation, the Segformer. We tried to replace the Efficient Self Attention (one of the main features of the Segformer) with the Criss-Cross Attention but the results achieved show that this variant wasn't able to achieve better performances than the state of the art model. The code is available in the GitHub repository[1].

2 Introduction

The aim of the project is to replace the Segformer Efficient Self-Attention with the Criss Cross Attention and see if it improves performances on semantic segmentation tasks. As far as we know, this approach has not been tried yet.

In order to accomplish this we made use of the Segformer model in the HuggingFace repository[2], which implements the Segformer described in Wang et al. [2021], and the Criss Cross attention implemented in the MMCV repository[3], which implements the attention proposed in Huang et al. [2020].

3 Related work

3.1 Segformer

SegFormer is a semantic segmentation framework which unifies Transformers with lightweight multilayer perceptron (MLP) decoders. Its two main features are: 1) a hierarchically structured Transformer encoder which outputs multiscale features. It does not need positional encoding, thereby avoiding the interpolation of positional codes which leads to decreased performance when the testing resolution differs from training. 2) an MLP decoder that aggregates information from different layers, and thus combining both local attention and global attention. The approach is scaled to obtain a series of models from SegFormer-B0 to SegFormer-B5, reaching significantly better performance and efficiency than previous counterparts.

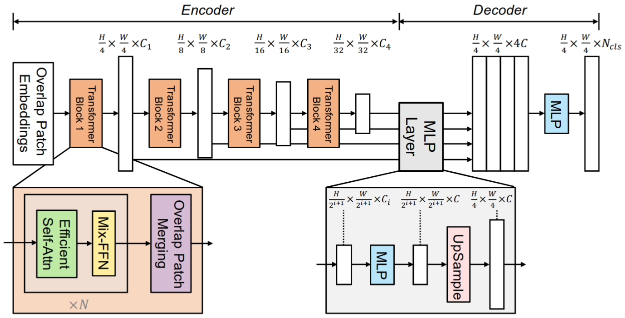

Encoder
Decoder
$\frac{H}{4}\times\frac{W}{4}\times C_1$
$\frac{H}{8}\times\frac{W}{8}\times C_2$
$\frac{H}{16}\times\frac{W}{16}\times C_3$
$\frac{H}{32}\times\frac{W}{32}\times C_4$
$\frac{H}{4}\times\frac{W}{4}\times 4C$
$\frac{H}{4}\times\frac{W}{4}\times N_{cls}$
Overlap Patch Embeddings
Transformer Block 1
Transformer Block 2
Transformer Block 3
Transformer Block 4
MLP Layer
MLP
Efficient Self-Attn
Mix-FFN
Overlap Patch Merging
$\times N$
$\frac{H}{2^{i+1}}\times\frac{W}{2^{i+1}}\times C_i$
$\frac{H}{2^{i+1}}\times\frac{W}{2^{i+1}}\times C$
$\frac{H}{4}\times\frac{W}{4}\times C$
MLP
UpSample

Figure 1: SegFormer framework.

[1] https://github.com/RiccardoAgabiti/Advanced-Machine-Learning-Final-Project
[2] https://huggingface.co/docs/transformers/model_doc/segformer
[3] https://mmcv.readthedocs.io/en/latest/_modules/mmcv/ops/cc_attention.html

3.2 Efficient Self Attention

In the original multi-head self-attention, given an image of dimensions H x W x C, each of the heads Q, K, V have the same dimensions N × C, where N = H × W is the length of the sequence, the self-attention is estimated as:

$$Attention(Q, K, V) = Softmax\left(\frac{QK^t}{\sqrt{d_{head}}}\right)V$$

The Efficient Self-Attention instead uses a sequence reduction process, dependent to a ratio R, that works as follows, where K is the sequence to be reduced:

$$\hat{K} = Reshape\,(N/R, C * R)\,(K), \quad K = Linear\,(C * R, C)\,(\hat{K}).$$

Where $Linear(C_{in}, C_{out})(*)$ refers to a linear layer taking a C_{in} dimensional tensor as input and generating a C_{out} dimensional tensor as output.

3.3 Criss Cross Attention

The Criss-Cross Attention module harvests, for each pixel, the contextual information of all the pixels on its criss-cross path, i.e. all the pixels belonging to the given x,y axis. This approach captures long-range dependencies useful for contextual information to benefit visual understanding problems. By taking a further recurrent operation, each pixel can finally capture the full-image dependencies. Overall, comparing Criss Cross attention to the non-local block results in less memory usage, an increased computational efficiency and gives state of the art performance.

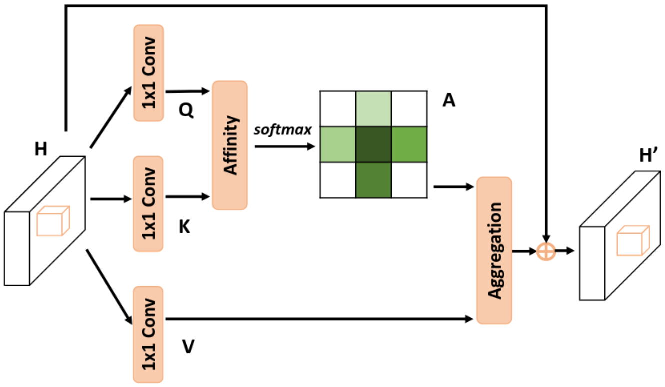


Figure 2: Criss Cross attention module details.

4 Dataset and Benchmark

4.1 Dataset

MIT Scene Parsing Benchmark (SceneParse150[4]) data comes from ADE20K Dataset from Zhou et al. [2017] which contains more than 20K scene-centric images exhaustively annotated with objects and object parts. The dataset is divided into 20K images for training, 2K images for validation, and another batch of held-out images for testing. There are in total 150 semantic categories included for evaluation.

[4] https://huggingface.co/datasets/scene_parse_150

4.2 Data Augmentation

In order to increase the diversity of our training set we applied some random transformations, each of which would be applied with a probability of 0.25.

The transformations we made are the following:

- Horizontal Flip
- Inversion
- Conversion to grayscale
- Gaussian Blurring
- Random resize with ratio 0.5-2.0
- Random cropping to 512 × 512, 1024×1024

5 Proposed method explained

5.1 Modeling choices

We first loaded the B0 Segformer model from HuggingFace[5], whose encoder is fine-tuned on *Imagenet-1k*, and then replaced the efficient self-attention layers with Criss Cross attention layers we implemented starting from the baseline code in the MMCV repository[6]. All the model layers were at this point frozen, except those with the new attention module, which were randomly initialized, and those in the decoder. The such obtained architecture looks like the following:

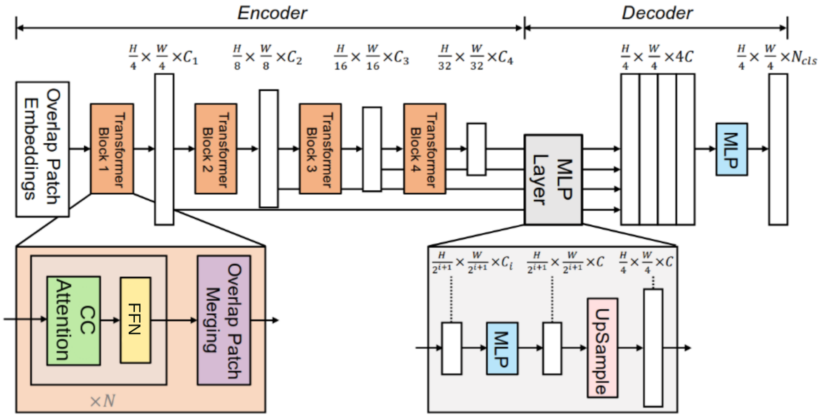

Encoder
Decoder
$\frac{H}{4}\times\frac{W}{4}\times C_1$
$\frac{H}{8}\times\frac{W}{8}\times C_2$
$\frac{H}{16}\times\frac{W}{16}\times C_3$
$\frac{H}{32}\times\frac{W}{32}\times C_4$
$\frac{H}{4}\times\frac{W}{4}\times 4C$
$\frac{H}{4}\times\frac{W}{4}\times N_{cls}$
Overlap Patch Embeddings
Transformer Block 1
Transformer Block 2
Transformer Block 3
Transformer Block 4
MLP Layer
MLP
CC Attention
FFN
Overlap Patch Merging
$\times N$
$\frac{H}{2^{i+1}}\times\frac{W}{2^{i+1}}\times C_i$
$\frac{H}{2^{i+1}}\times\frac{W}{2^{i+1}}\times C$
$\frac{H}{4}\times\frac{W}{4}\times C$
MLP
UpSample

Figure 3: Criss Cross Segformer.

[5] https://huggingface.co/nvidia/mit-b0

[6] https://mmcv.readthedocs.io/en/latest/_modules/mmcv/ops/cc_attention.html

5.2 Training of the model

Many of the training choices we made were guided mainly by a major constraint: Colab's GPU memory. Segformer indeed isn't such a lightweight model, hence in order to train it we had to use batch sizes of only 1 image, which would be enhanced to a batch size of 4 by using gradient accumulation[7]. Always in the spirit of not overloading the GPU we made use of the Adafactor optimizer, which was first introduced in Shazeer and Stern [2018], and it's well known for its sublinear memory cost.

The training was then carried on using a Multi-Class Cross Entropy loss for 16 epochs with a learning rate of $6 \cdot 10^{-5}$ in order to tune the model. We report here the training loss:

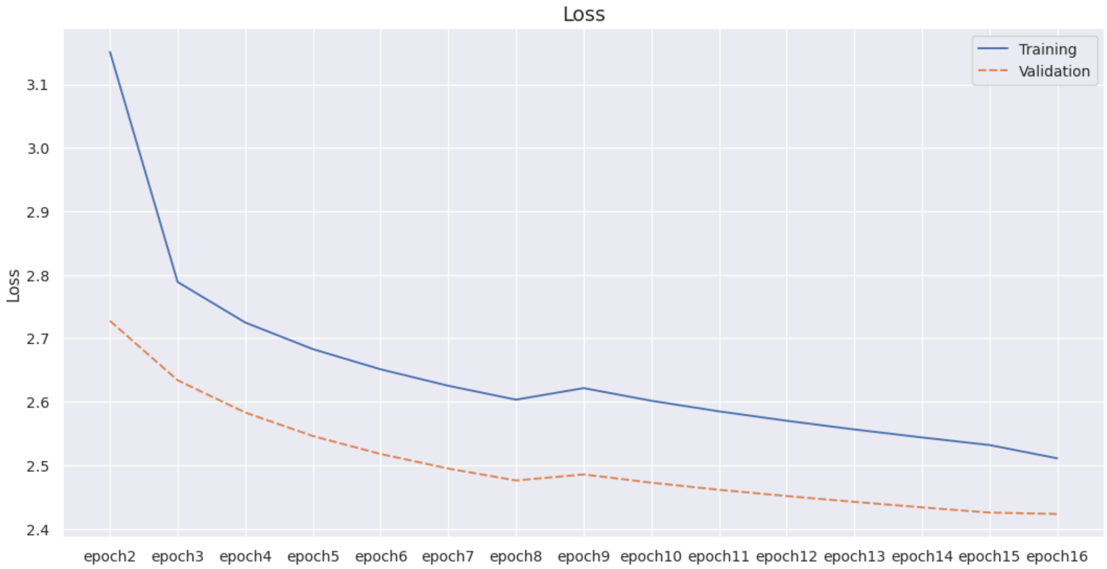
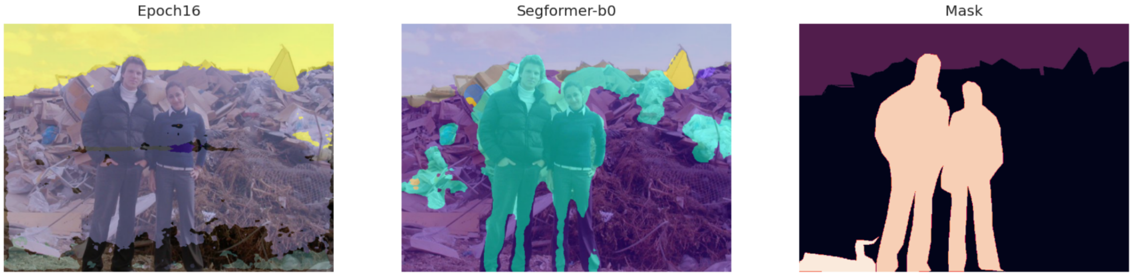


Figure 4: Criss Cross Segformer training loss

6 Experimental results

We report here some examples of application of the model to the validation set images:



[7] citing one of the HuggingFace guides at https://huggingface.co/docs/accelerate/usage_guides/gradient_accumulation 'Gradient accumulation is a technique where you can train on bigger batch sizes than your machine would normally be able to fit into memory. This is done by accumulating gradients over several batches, and only stepping the optimizer after a certain number of batches have been performed.'

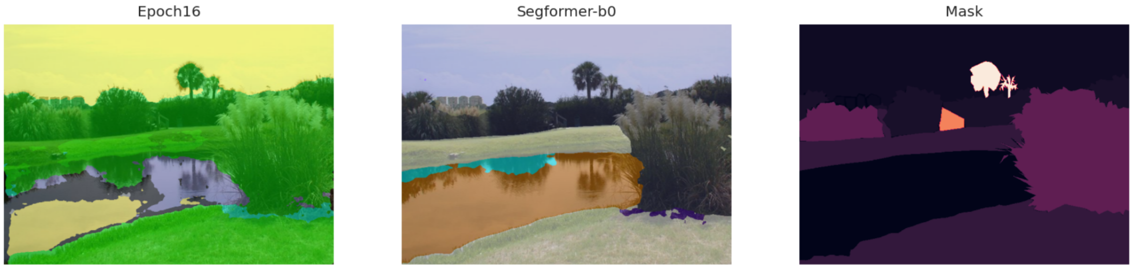


Figure 5: Example images

We can notice that for this dataset our Criss-Cross Segformer model is overall outperformed by the state of the art version of it. Although it has his flaws we can see that this kind of attention is able to grasp the overall structure of the image, being able to distinguish semantic areas of it. While it might not be able to distinguish all of the single objects in a picture (performing expecially bad when people are present) it is great at separating 'big' semantic areas, expecially in open space environments. We believe that, with the proper fine-tuning on the right dataset, this model could make his way into some specific applications like semantic segmentation for satellite images.

We report semantic segmentation performance using mean Intersection over Union (mIoU):

	MiT-B0	Criss-Cross Segformer
MIOU	0.37	0.10

Table 1: Summary table of the peformance of the models.

7 Conclusions and Future work

Our model did not result as performing as the original Segformer and that is due to the different type of attention we used. We believe that the training of the model can definitely be improved for example by using a better optimizer and bigger batches for the images, which we weren't able to do because of GPU's memory constraint. It's hard to tell if with a longer and improved train the Criss-Cross Segformer can outperform the MiT B0 but we suppose that it could work well in the context of semantic segmentation for satellite images. In future it could be retrained, fine-tuned and tested on this kind of task.

References

Z. Huang, X. Wang, Y. Wei, L. Huang, H. Shi, W. Liu, and T. S. Huang. Ccnet: Criss-cross attention for semantic segmentation. https://arxiv.org/abs/1811.11721, 2020.

N. Shazeer and M. Stern. Adafactor: Adaptive learning rates with sublinear memory cost. https://arxiv.org/abs/1804.04235, 2018.

E. X. W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. https://arxiv.org/abs/2105.15203, 2021.

B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Scene parsing through ade20k dataset. https://people.csail.mit.edu/bzhou/publication/scene-parse-camera-ready.pdf, 2017.