Transformer in Computer Vision

Jianzhe Yu * 20890061
Yufan Chen * 20878362
Yilin Ye * 20922644
Runze Li * 20931982

Abstract

While the transformer has been a powerful model in nature Language processing field original, its application to computer vision also has been proven to be impressive and influential. In this project, we choose two specifical downstream tasks in computer vision, and investigate the performance of Vision Transformer (ViT) and Detection Transformer (DETR) for these two tasks respectively. Moreover, we inspect some features in these two models, which help us better understand how transformer work in computer vision. We achieve excellent results in both two tasks, and prove that transformer can be levaraged in computer vision.

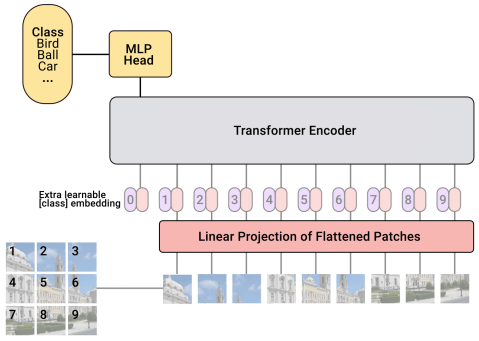

Figure 1. Vision Transformer Model overview. Source: Google Lab

1. Introduction

Transformers, which use a self-attetion multi-head mechanism, have become the most popular and powerful model in nature language processing field[15]. In this model, sequence of data can be fed into the network parallelly, and only need to be processed once. It is nature to use a sequence of data to represent word and sentence in nature language processing. Other than recurrent neuron network (RNN) and Long Short-Term Memory (LSTM)[8], Transformer can process the sequence data parallelly. This high parallelism can save computation time and train the network more effeiently. Following transformers, bidirectional Transformers(BERT) and Generative Pre-Training(GPT) have been used for lanaguage related task, and achieve State of The Art performances in recent year[3, 13].

Before using transformers, convolutional neuron networks (CNN) are the primary tools in computer vision. In image classification, AlexNet was the first convolutional neural network that achieve high precision in ImageNet[9]. Later, Resnet achieve the best performance for image classification, which even beat human's accuracy[7]. For object detection, Region Based Convolutional Neural Networks (RCNN) was the fundamental network in this field[5]. Recently, You Only Look Once (YOLO) has achieve the State of the art result in object detection[14].

The biggest the difference of data in computer vision and nature Language processing is that we always input the images as a whole, while word and sentence can be divided into sequence. How to treat images as a sequence of data is the key point for transformers in computer vision. In the next two section, we will discuss how vision transformer (ViT) and Detection Transformer (DETR) can be used in image classification and object detection[2, 4].

2. ViT in image classification

2.1. ViT

Vision Transformer (ViT) is an expansion of the basic Transformer concept, which was proposed by Vaswani in 2017 [4, 15]. It is merely the implementation of a Transformer in the image domain with a small tweak to handle the various data modalities. To be more precise, a ViT employs different tokenization and embedding techniques. The general architecture, though, does not change. By processing an image as a series of small image patches, Vision Transformers enable interaction between attention processes between all points in the image (i.e., global attention).

The input image, of height, width, and the number of

*All team members have contributed the same effort for this project. For ViT, Jianzhe Yu finish the training part, and Runze Li finish the visualization. For DETR, Yufan Chen finish the training part, and Yilin Ye finish the visualization. All four members contributed to the report and video.

channels (H, W, C), is divided into patches in order to structure the input image data as the way in the NLP task. As a result, we have N patches $N = \frac{HW}{P^2}$ with the resolution of (P, P) pixels. Each patch is then flattened into a single vector with length $P^2 * C$. Then, the flattened image patches are mapped to D dimensions with a trainable linear projection, resulting in a sequence of embedded image patches. The sequence is prefixed with a learnable class embedding. (The classification output, y, is represented by the value of class embedding.) Learnable positional embeddings, or $E_{\{pos\}}$, are added to the patch embeddings for the model to learn about the structure of the image. Following the described processes, the sequence of embedding vectors are:

$$z_0 = [X_{class}; X_P^1 E; X_P^2 E; \ldots; X_P^N E] + E_{pos}$$

Next, z_0 is passed as the input to a standard Transformer encoder, which is made up of a stack of L identical layers, to do classification. After that, the first output, which corresponds to the class embedding, is taken out and fed into a MLP classification head. The Gaussian Error Linear Unit (GELU) non-linearity is implemented by the MLP. Softmax can be used on this output to produce classification labels. Figure 1 is an illustration of the aforementioned operations.

ViT is first pre-trained over a massive dataset (such as the JFT-300M, ImageNet) and fine-tuned and evaluated the model on downstream tasks. For pre-training, an MLP with one hidden layer and GELU non-linearity is used to create the classification head that is connected to the encoder output. For fine-tuning, a single feedforward layer of size $D \times K$ is used in place of the MLP, where K is the number of classes necessary to complete the given task.

2.2. Training

In this part, we use flower102 as our dataset[12]. We set our learning rate as 1e-3, and use adamW optimizer[11]. We fix all the hyperparamters in the pretrained ViT except the head layer, which is a linear leyer output the probability of all classes. We train the ViT under this setting for 30 epoches.

As comparison, we also finetune the ResNet50 under the same setting[7]. The losses are getting closer during the training time as shwon in figure 2. Finally, we achieve good result for ViT, which is comparable for ResNet as shown in Table 1.

	ResNet	ViT
Precision	90.588	87.549

Table 1. Comparison of effects of different backbone on results(On Dataset500)

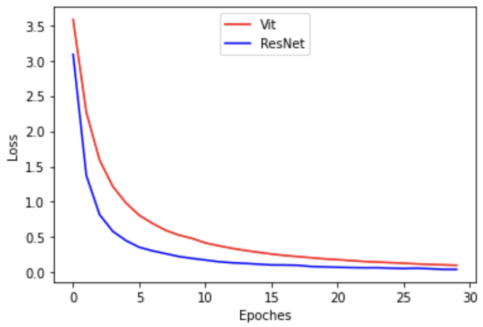

Figure 2. Training process

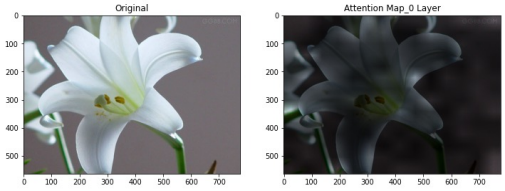

Figure 3.

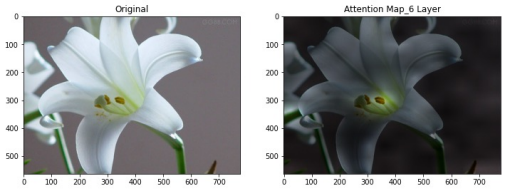

Figure 4.

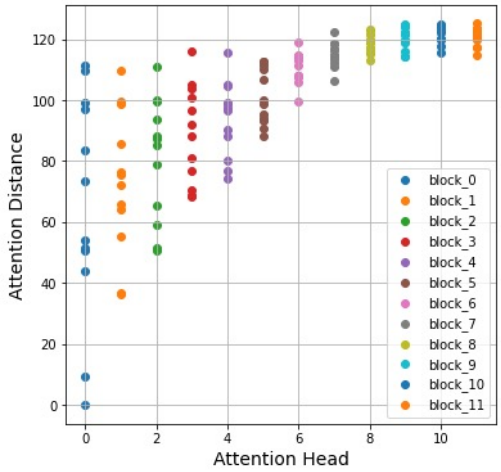

Figure 5.

2.3. visulization

Visualization work is conducted to understand ViT better, including attention map and mean attention distance. The idea of attention map is adapted from attention rollout [1], which quantifies how information flows through self-

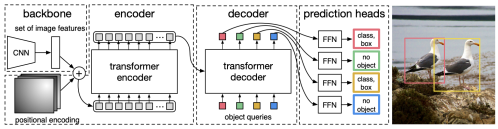

Figure 6. DETR Model overview.

attention layers of Transformer blocks. At each transformer block, we have an attention matrix A_{ij}, which defines how much attention is going through from token j in the previous layer to token i in the next layer. In this experiment, we multiply the matrices between each two attention layers, and re-plot the image by increasing the pixels with high attention weights and discarding those with low weights.

Figure 3 and Figure 4 illustrate the attention map in attention layer 0 and attention layer 6, while the attention maps for each one of the total 12 attention layers are listed in Appendix.A. Pixels with brighter colors and higher contrast indicate that this part has higher attention weight. We can see a significant change of pixels with higher attention weight by comparing the two figures. In attention layer 0, the perimeter of the petals and the entire flower are focused, while in attention layer 6, the focused part transfers to the calyx and stamen.

The mean attention [1] distance provides another method to evaluate attention layers. Defined as the distance between query tokens and the other tokens times attention weights, mean distance scattered map can show how the pixels with attention are concentrated. With this plot 5, we can see that there are heads attending to the whole patch in the early layers, which is shown by points arranged relatively loosely on a vertical axis. While in the deeper attention layers, points are arranged closely, indicating the head is focusing on a more accurate part. Together with attention maps above, we demonstrate the process of attention in ViT in a visual way.

3. DETR in object detection

3.1. DETR

The end-to-end object recognition achieved by the DETR approach eliminates those hand-crafted designs that require prior knowledge of image detection. It constructs an effective prediction framework based on convolution feature maps with transformers and outperforms many models in terms of simplicity and accuracy. DETR solves image detection problems by using CNN backbone as a feature extractor follow by a Transformer encoder-decoder; it converts the image feature map into the result of object detection in a direct manner. Additionally, DETR can be naturally extented to carry out other tasks such as panoptic segmentation by layering a mask head based on the output from

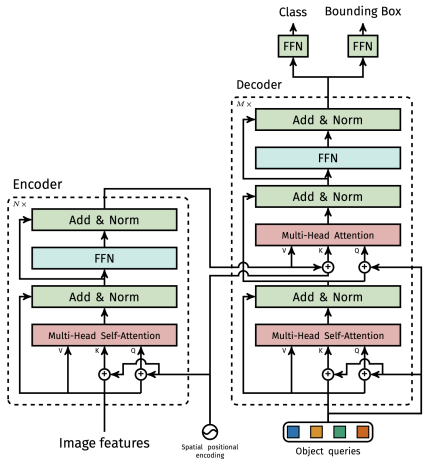

Figure 7. DETR transformer architecture.

the decoder.

Figure 6 illustrates the overall structure of DETR. The overall framework can be split into four parts: Backbone, Encoder, Decoder, and Detection Heads. First, The backbone takes out the feature representation of input tokens with $(H_0 \times W \times C)$ and outputs with dimension $(\frac{H}{32} \times \frac{W}{32} \times 2048)$. Then, positional encoding is added to the backbone outputs before passing them to the transformer-based encoding and decoding pipeline. A set number of learnable tokens (positional embedding) known as "object queries" are also added as additional input to the decoder in order to predict multiple objects. Final predictions for classification and bounding box are then produced by the decoder via output embeddings through FFN (feed-forward network) in the detection heads.

In particular, the encoder layer follows a standard Transformer encoder block, which includes a multi-head self-attention module and a feed-forward network. For the decoder, it also has a standard Transformer architecture using multi-headed attention and self-attention to transform N input embeddings to N output embeddings. The only difference is, at each layer, the DETR decodes the object in parallel. The architecture of the encoder and decoder modules is shown in figure 7.

3.2. Experiment

This part shows that DETR achieves competitive results quantitative evaluation on SSLAD-2D dataset and a data set taken personally in the field and annotated manually, all the datasets is about automatic drive, because now automatic driving is a very promising research direction, it can assist the driver, help the driver cope with dangerous situations more calmly, and even replace the driver in the future, We wanted to explore the potential of DETR for automatic driving. Furthermore, we we extend DETR to achieve semantic

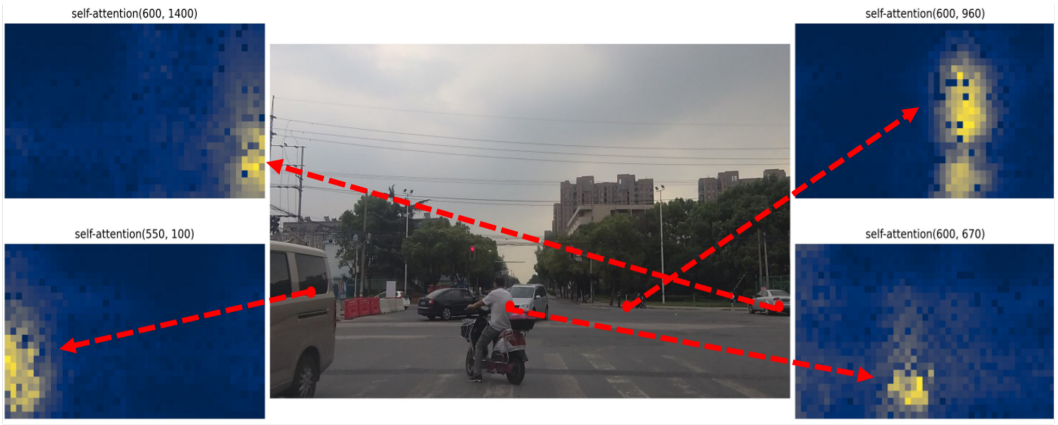

Figure 8. Encoder attention map of SSLAD-2D

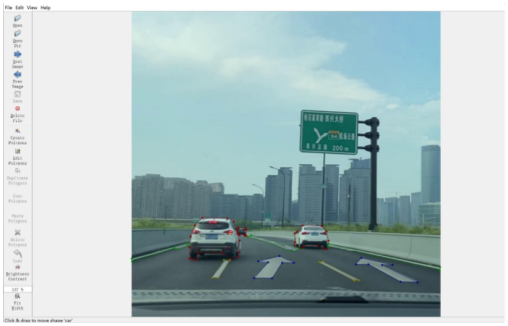

Figure 9. annotation process

```
--|data
    --|coco
        --|annotations
            --|instances_train2017.json
            --|instances_val2017.json
        --|train2017
        --|val2017
```

Figure 10. coco format

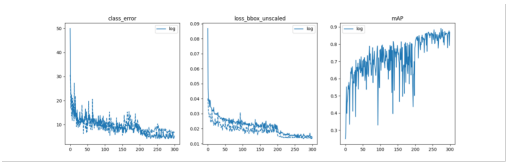

Figure 11. visual training process of class error, box loss and AP50

segmentation, and present a real-time segmentation in the vehicle camera to show the usability of DETR in automatic driving.

3.3. Dataset

SSLAD-2D Is a 2D autonomous driving data set released by Huawei Noah's Ark Laboratory and Sun Yat-sen University. It has three characteristics: large data scale, strong data diversity and strong generalization ability. It contains 5k training images, 5k validation images, and 20k unlabeled images with 7 categories.('BG', 'Pedestrian', 'Cyclist', 'Car', 'Truck','Tram', 'Tricycle')

Individual Manually annotated dataset (Dataset500) This is a personal data set taken personally in the field and annotated manually. Contains 400 training images and 100 test images with 5 categories.('BG' ,'arrow', 'car', 'dashed', 'line')

Data preprocessing:

First, the labelme software was used to mark 400 training images from Dataset500 in detail, including 'background' ,'arrow', 'car', 'dashed', 'line'.The specific process of annotation is shown in Fig.9

All data is then converted to coco format,which in shown in Fig.10

3.4. Training

We train DETR with AdamW setting the initial transformer learning rate to 0.0001,the backbone learning rate to 0.00001, and weight decay to 0.0001[11]. All transformer weights are initialized with Xavier initialization[6], and the backbone is with ImageNet-pretrained ResNet101 model from torchvision with frozen batchnorm layers.We employ CEloss as label loss, L1 loss and giou loss as box loss, employ AP, AP50, AP75, APS, APM, APL as evaluation function.And training the model for 300 epochs on dataset500 takes 8 hours, and on SSLAD-2D takes about 3 days using a single RTX 3080 GPU.

Fig.11 shows some visual training processes for dataset500, including class error, box loss and AP50. We can see that box loss converges at 0.015, when AP reaches about

Datasets	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
SSLAD-2D	43.3	64.4	41.5	24.1	37.1	57.9
Dataset500	54.8	87.0	59.6	32.5	69.1	66.7

Table 2. The final result of training on two datasets

backbones	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
ResNet50	53.2	85.8	57.8	31.4	68.0	65.5
ResNet101	54.8	87.0	59.6	32.5	69.1	66.7

Table 3. Comparison of effects of different backbone on results(On Dataset500)

87%.

3.5. Validation

After training, we shows the result of evaluation functions for the final models for two datasets in Table 2.We can see that although SSLAD-2D has more samples, But the number of validation images and training images is the same, its result is worse than dataset500.Then in Fig.8, we visualize the attention maps of the last encoder layer of the trained model, focusing on a few points in the image. The encoder seems to separate instances already, which likely simplifies object extraction and localization for the decoder.

Then in Fig.12 we visualize the attention maps of the last decoder layer of the trained model, focusing on every predicted object, and in Fig.13 we show the final prediction of two images from different datasets,we can see DETR detects almost all objects, even if they are partially obscured.

We also show more prediction result of two datasets in appendix B, we can see DETR is very robust and can accurately detect different categories targets on the road in complex road environments and in different weather conditions (e.g. night, cloudy, rainy)

3.6. Ablation Study

In this part, we conducted two ablation experiments to explore the effects of the different backbone and number of encoder on the model.First, we change the backbone from ResNet101 to ResNet50, the result is shown in Table 3.With ResNet50, all type of AP drops by 1 to 2 points. We hypothesize that, the selection of back bone is crucial for detection.

Then we evaluate the importance of global imagelevel self-attention by changing the number of encoder layers,the result in shown in Table 4. Without encoder layers, overall AP drops by 4.5 points, with a more significant drop of 6.8 AP on large objects. We hypothesize that, by using global scene reasoning, the encoder is important for detection.

3.7. Extension on semantic segmentation

To make DETR more widely available in autopilot, we extended DETR to semantic segmentation by adding a mask head on top of the decoder outputs. We show the results of the segmentation in Fig.14, And present a real-time seg-

Figure 12. Decoder attention map

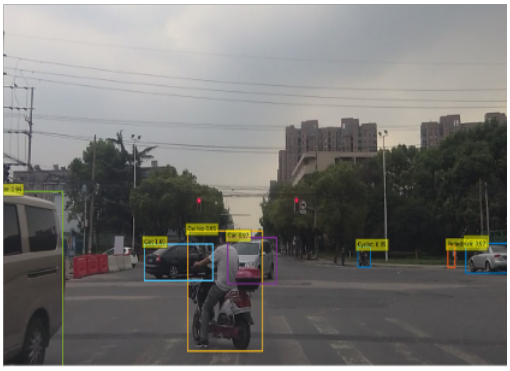

Figure 13. Final prediction

Number	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
0	50.3	82.2	54.5	29.3	64.8	59.9
4	53.9	85.8	58.7	31.6	68.1	65.2
6	54.8	87.0	59.6	32.5	69.1	66.7

Table 4. Comparison of effects of different number of encoder layer a on results(On Dataset500)

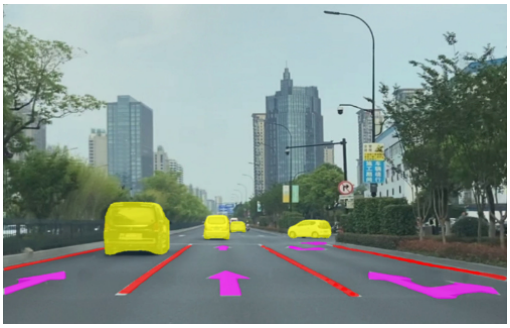

Figure 14. Result of segmentation

mentation in the vehicle camera to show the usability of DETR in automatic driving, which is shown in detail in our powerpoint presentation, we hope that in the future such real-time segmentation can be used in actual driving.

4. Conclusion and further improvement

CNN model have achieved a great success in computer vision field because they utilize two inductive bias: locality and spatial relations. However, both ViT and DETR do not utilize too much spatial information about images, and they can achieve almost the same performance as the cnn model. If ViT and detr utilize more spatial information by manipulating images patch, they can achieve better performance even better than current state of the art CNN model.

We prove that transformer could be a successful model in computer vision by experimenting it in two downstream task: images classification and object detection using two model: vision transformer and detection transformer. By utilizing more spatial information, transformer variant model can be more powerful in the computer vision like deformable detr and swin transformer[10, 16]. We believe transformer, which has powered a recent breakthrough in NLP, can also be leveraged to improve the performance of deep learning for other fields.

References

[1] S. Abnar and W. H. Zuidema. Quantifying attention flow in transformers. *CoRR*, abs/2005.00928, 2020.

[2] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. End-to-end object detection with transformers. *CoRR*, abs/2005.12872, 2020.

[3] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[4] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020.

[5] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR*, abs/1311.2524, 2013.

[6] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In Y. W. Teh and M. Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR.

[7] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[8] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[10] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *CoRR*, abs/2103.14030, 2021.

[11] I. Loshchilov and F. Hutter. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101, 2017.

[12] M.-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing*, Dec 2008.

[13] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. Improving language understanding by generative pre-training. 2018.

[14] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. *CoRR*, abs/1506.02640, 2015.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[16] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai. Deformable DETR: deformable transformers for end-to-end object detection. *CoRR*, abs/2010.04159, 2020.

A. Appendix: Details about Vision Transformer

The attention maps for each one of the total 12 attention layers in the experiment.

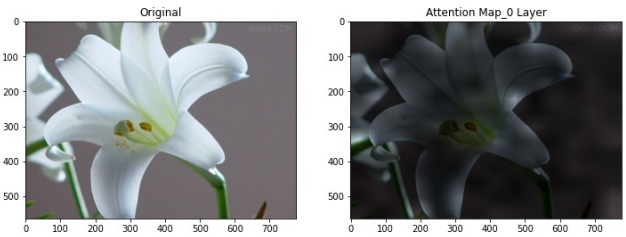

Figure 15.

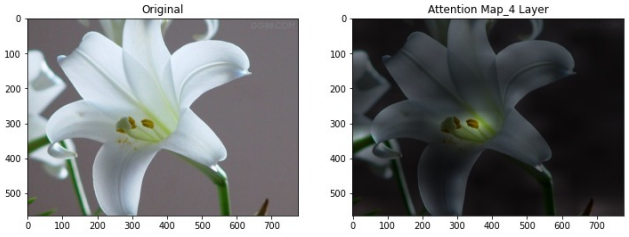

Figure 16.

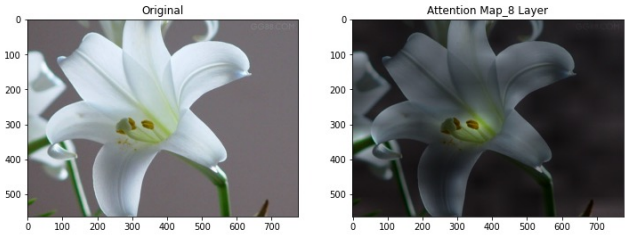

Figure 17.

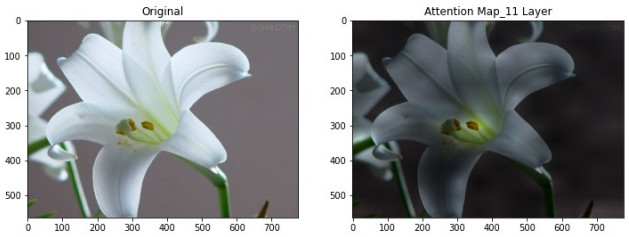

Figure 18.

We do another experiment on a more complex image, while the prediction output turns to be not accurate enough. However, we do find a significant change in the attention maps across 12 attention layers.

Figure 19.

B. Appendix: Details about DETR

In this part, we will show more results of our experiment on DETR.

Figure 20.

Figure 21.

Figure 22.

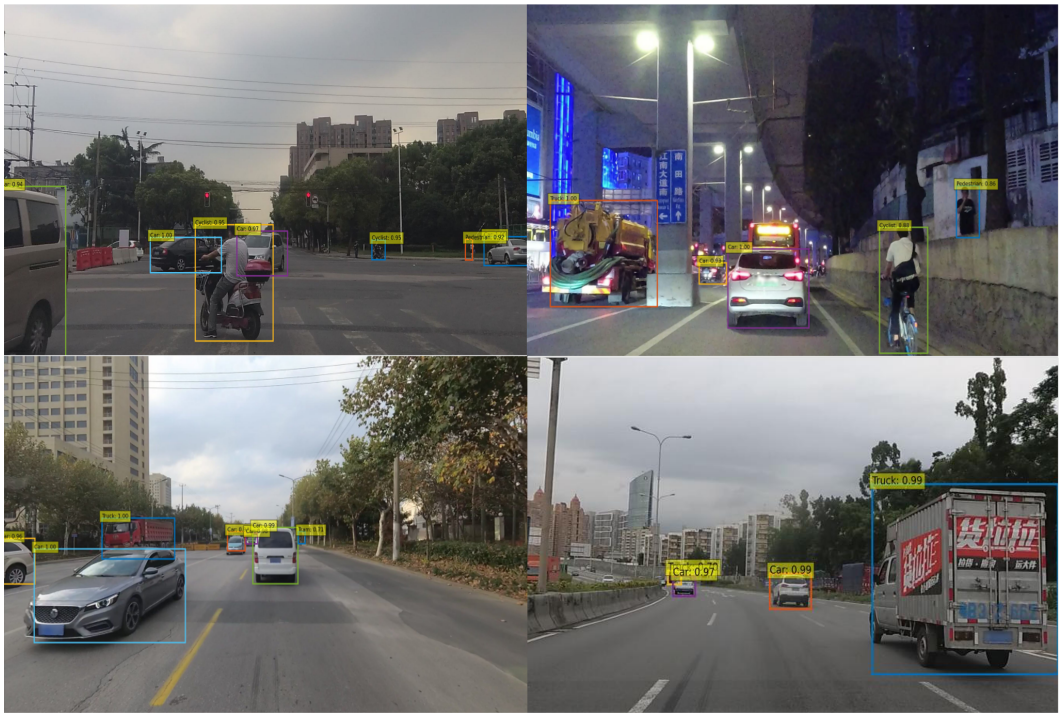

Figure 23. More prediction result of SSLAD-2D

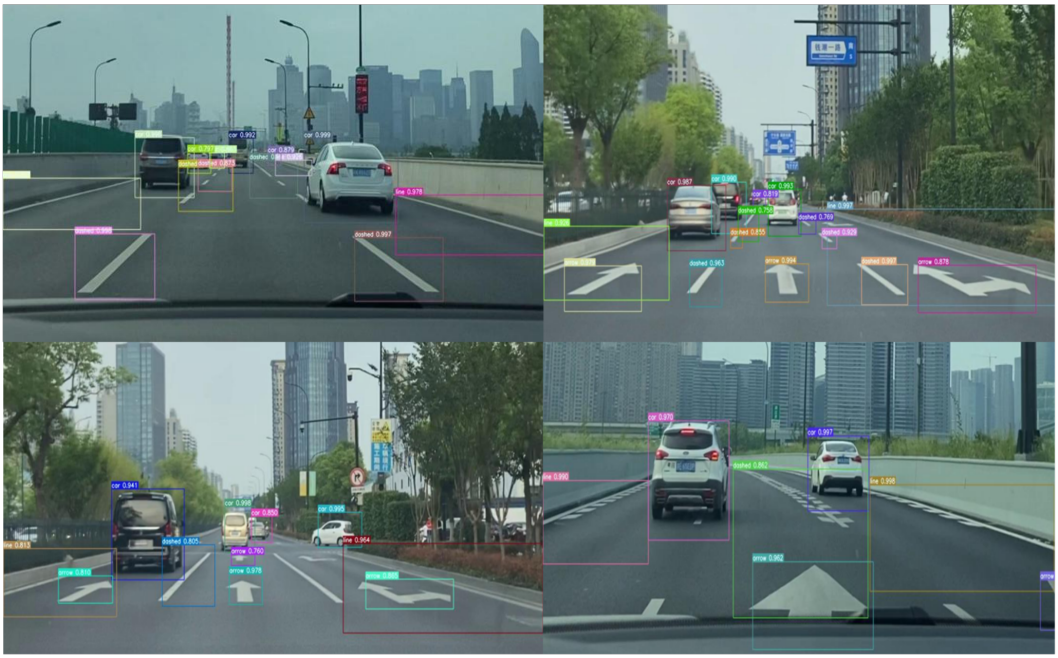

Figure 24. More prediction result of Dataset500

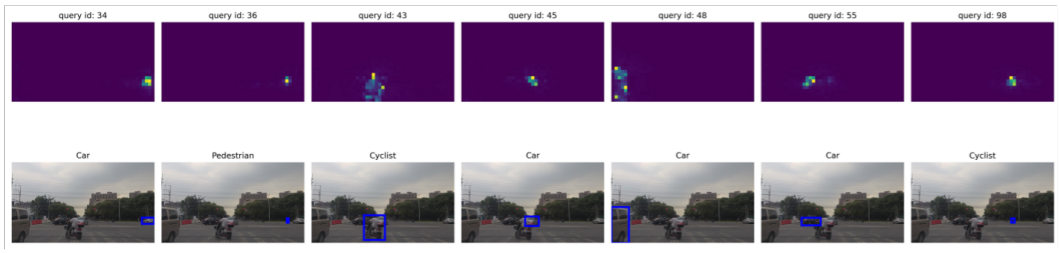

Figure 25. Decoder attention map of SSLAD-2D

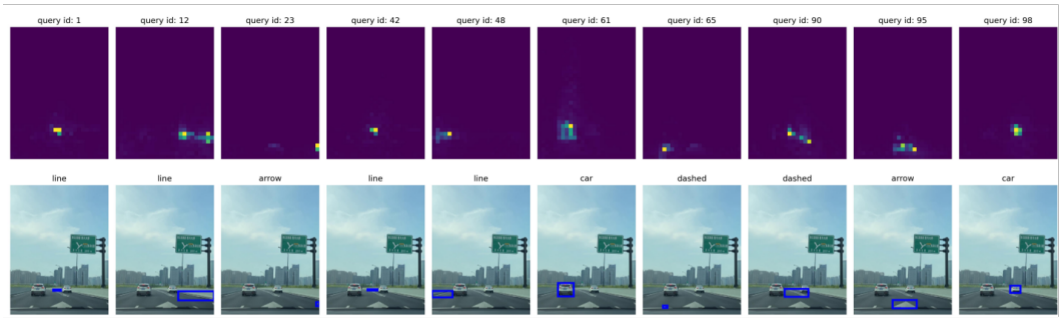

Figure 26. Decoder attention map of Dataset500

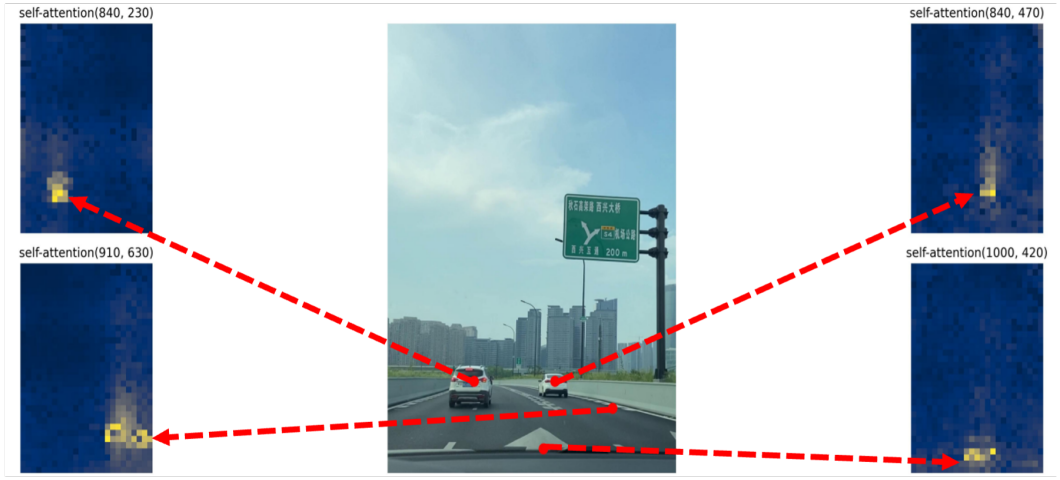

Figure 27. Decoder attention map of Dataset500