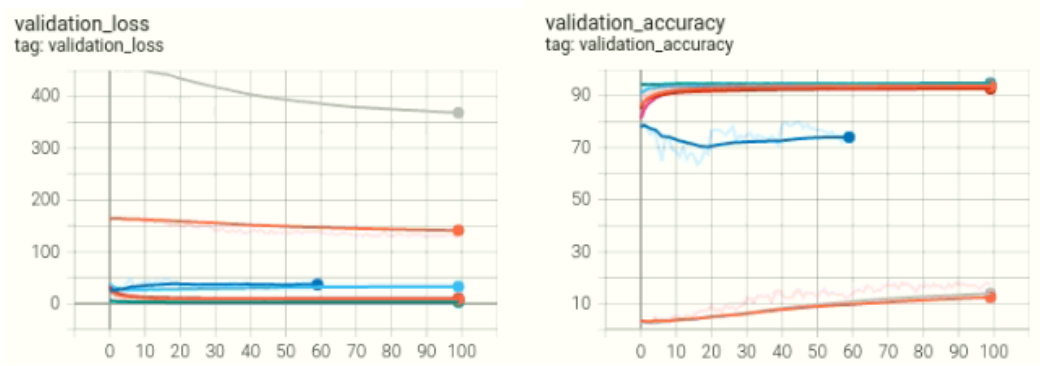
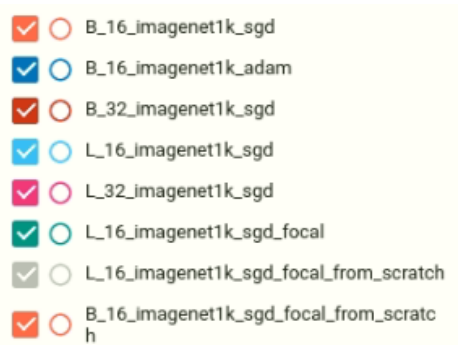
2021 DLCV
HW3
R09942171 電信所碩一 黃繼綸

Problem 1: ViT (no collaborators)

1. Ref : https://github.com/lukemelas/PyTorch-Pretrained-ViT

   In this problem, I conduct extensive experiments, such as model architecture, loss function, pretrain or not, size of input image…etc. I will make some analysis next. The results are illustrated as below:



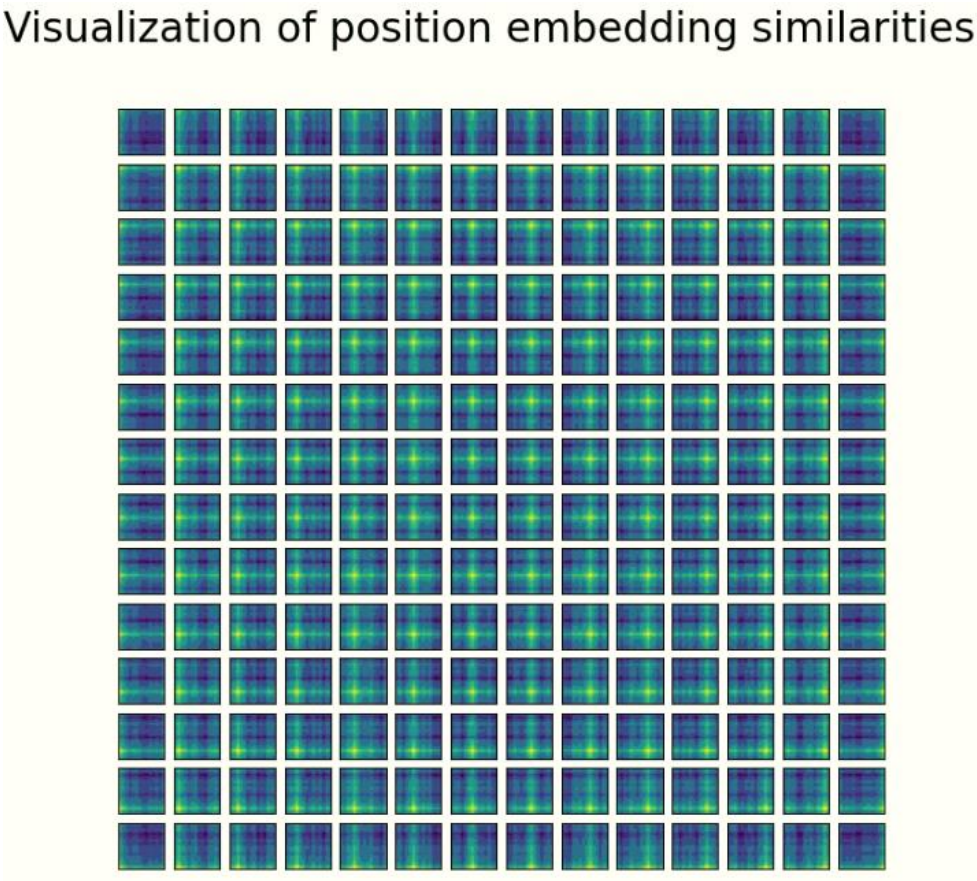


size(384, 384)	Validation Acc(%)
B_16_imagenet1k_adam	79.73
B_16_imagenet1k_sgd	94.33
B_16_imagenet1k_sgd_scratch	18.20
B_32_imagenet1k_sgd	93.27
L_16_imagenet1k_sgd	94.87
L_16_imagenet1k_sgd_scratch	21.33
L_32_imagenet1k_sgd	94.20
L_16_imagenet1k_sgd_focal	95.27
Size(224,224)	
L_16_imagenet1k_sgd_focal	95.33
L_16_imagenet1k_sgd(warm_up)	**95.73**

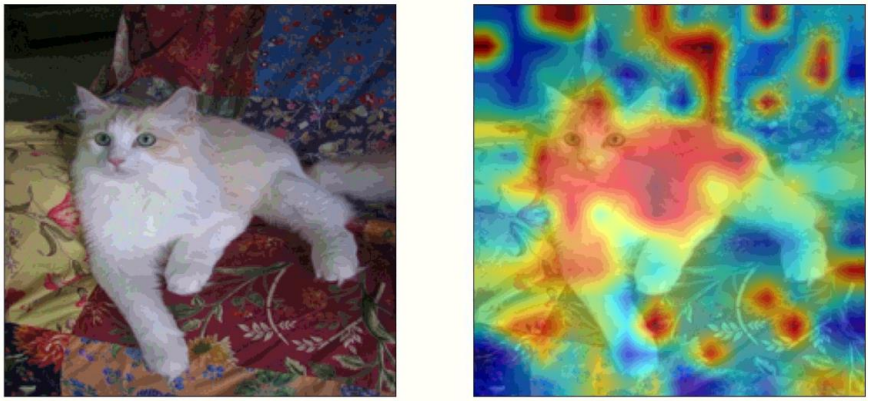
First, the model architecture is not very sensitive to the accuracy of the result. But, if I employ a larger model, such as "L_16_imagenet1k", its performance will significantly outperform the small models. **Second**, the optimizer is very sensitive to accuracy. If I apply the Adam instead of SGD, the model can't learn very well. **Third**, if I don't use the pretrained weight, the model almost can not learn. This tells me that the pretrained weight is very important for the training of ViT. **Fourth**, I try different loss functions on ViT, we can see that the performance is similar. **Fifth**, I try different sizes of images. After experiments, the size of the image is not an essential factor in training ViT. Finally, I utilize the model marked red color as my final decision.

2. **Visualize position embeddings**

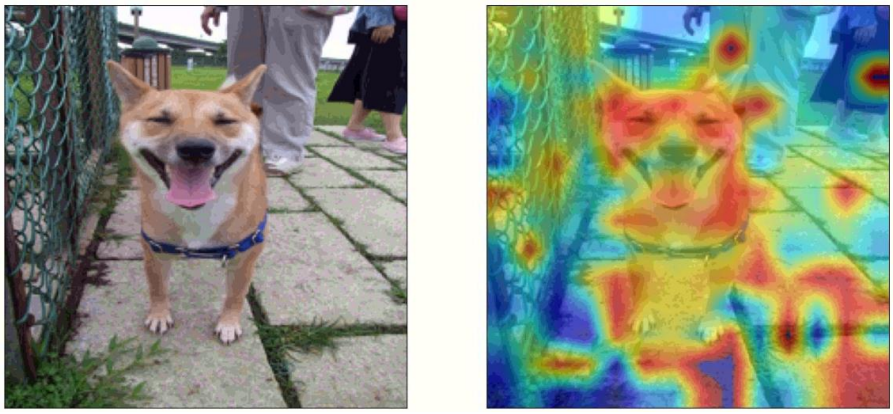
The result demonstrates that the cosine similarity between the same row or the same column will be higher compared to other positions. The result also demonstrates that the cosine similarity between the near pixels will be higher.



3. Visualize attention map

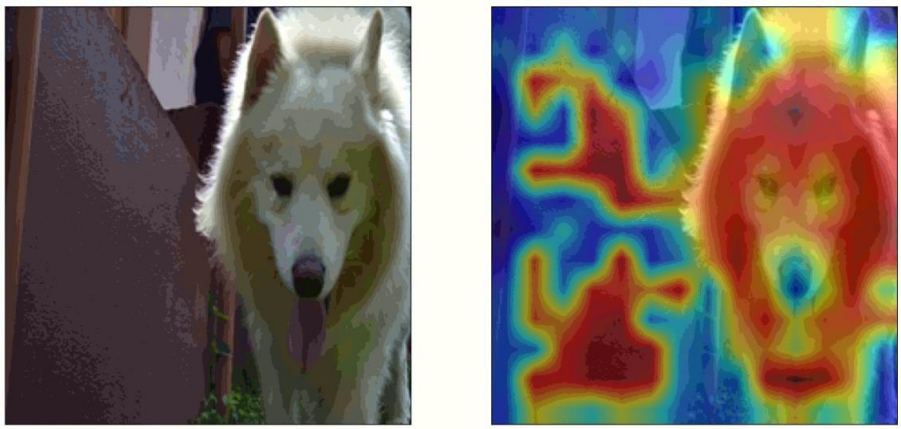


▲ 26_5064.jpg



▲ 31_4838.jpg

Most of the cat images in the training data demonstrate cats lying on the carpet indoors, while most of the dog images in the training data demonstrate dogs standing on the grass or the ground outdoors. Therefore, **in '26_5064.jpg'**, we see that there are some large attention weights on the carpet, but the weights on the cat still occupy the majority of the whole image. In addition, **in '31_4838.jpg'**, we see that there are some large attention weights on the ground or the crowd, but the weights on the dog still occupy most of the whole image.

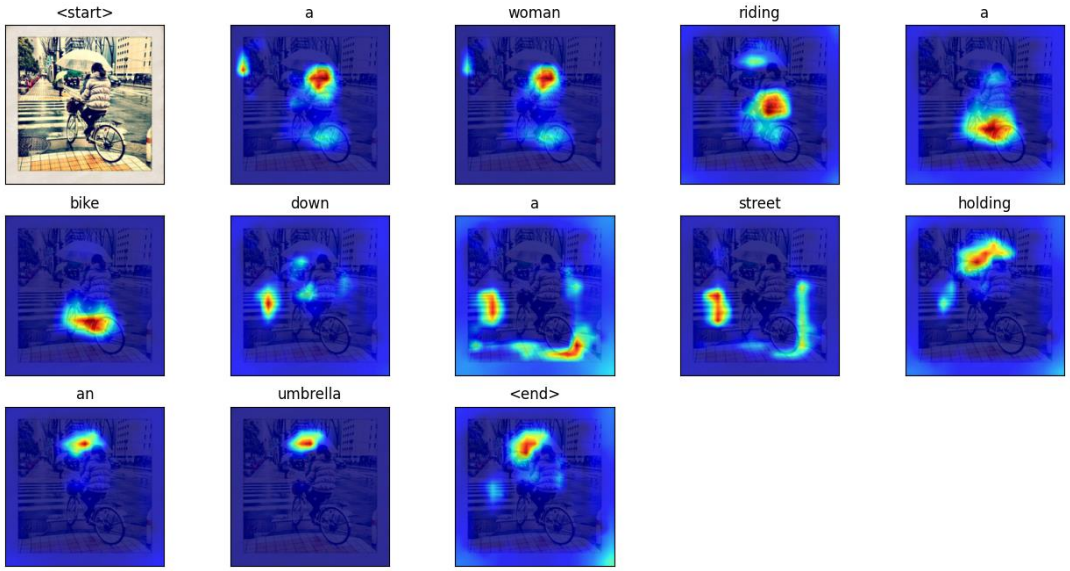


▲ 29_4718.jpg

In '29_4718.jpg', we see that some attention weights are on the wall. By observing the training data, I find that the backgrounds of most of the dog images are in plain textures. As a result, the background texture object may be one of the decision criteria for dogs. However, the attention weights on the dog still occupy the majority of the whole image. In other words, the model decision still depends on dog objects in the images.

Problem 2: Visualization in Image Captioning (no collaborators)

a. The attention maps are reasonable for the first word 'a' and second word 'woman' because the weights are high on the woman. The attention map is not entirely reasonable for the third word 'riding' because the action of 'riding' should focus more on foot. The attention maps are reasonable for the fourth word 'a' and the fifth word 'bike' because the weights are concentrated on the bike. The attention maps are reasonable for the sixth word 'down,' the seventh word 'a', and the eighth-word 'street' because the weights are concentrated more on the ground. The attention map is not entirely reasonable for the ninth word 'holding' because the weight should focus more on her right hand than the umbrella. The attention maps are reasonable for the tenth word 'an', and the eleventh word 'umbrella' because the weights are concentrated more on the umbrella.



b. In this problem, I trace the entire model of ViT, I understand how to use such an attention mechanism to the image captioning task. Before I plot the attention maps, I need to understand how the representations of word embedding and image embedding proceed through. Finally, I don't know how to project the attention maps into the original image. After asking TA, I find that I can use the overlap mechanism in the package of 'matplotlib' to solve this problem.