Deep Learning for Computer Vision (2021 Fall) HW3

R10942152 Ken Yu(游家權)

Problem 1: Image Classification with Vision Transformer (80%)

1. (10%) Report accuracy of your model on the validation set. (TA will reproduce your results, error ±0.5%)

a.(2%) Clearly mark out a single final result for TAs to reproduce.

Acurracy= 0.9507

b. (8%) Discuss and analyze the results with different settings (e.g. pretrain or not, model architecture, learning rate, etc.)

I used timm's pre-trained model to train on our training data for this task. And I found the validating performance on the training set reached astounding 100% for top1 accuracy when I trained over 70 epoches. This shows that ViT has no problem fitting the image classification data at all; however, it also shows ViT can't improve itself after 70 epochs, since its loss is very close to zero. Although the accuracy drops a little at the real validation set, it still maintains a pretty high accuracy.

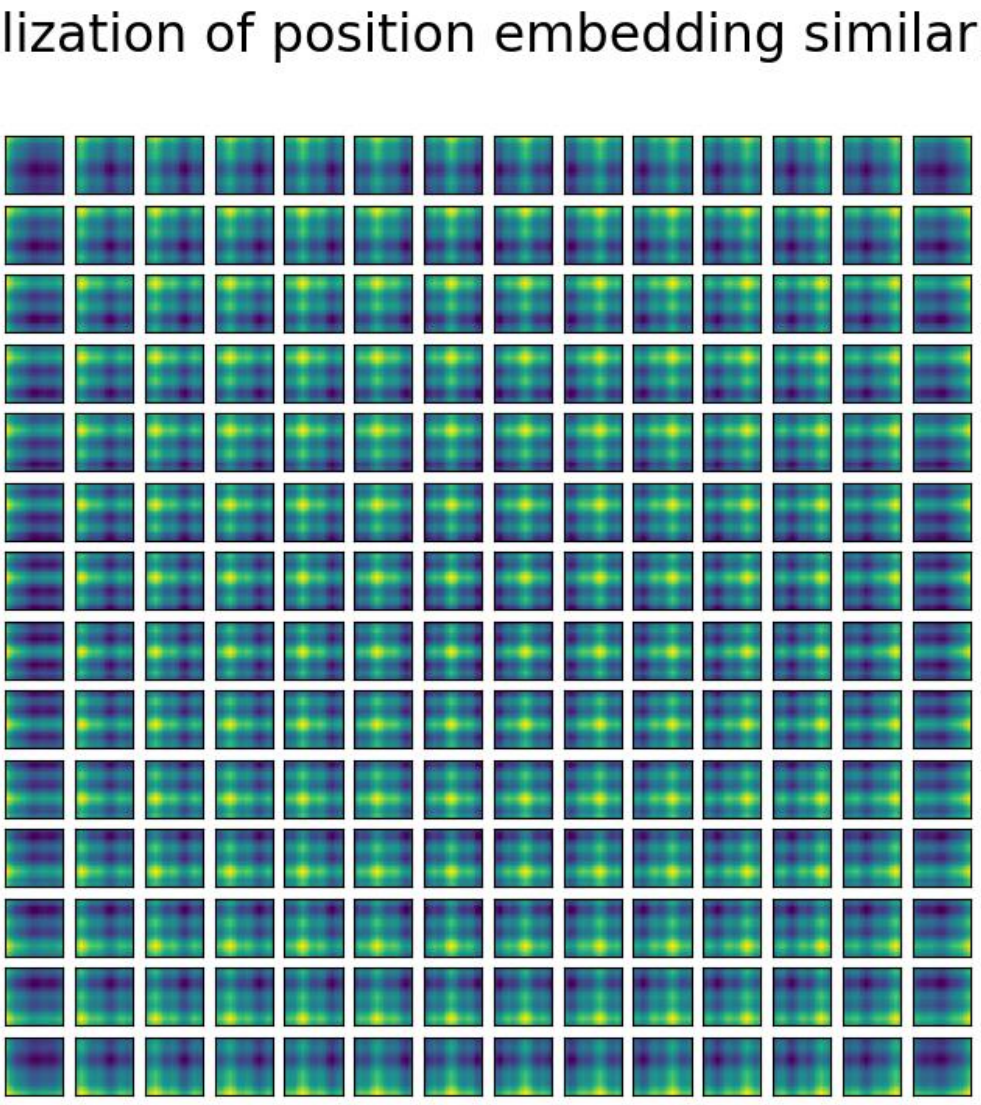
I have tried to train ViT with random init weights, but it's actually very hard to train. The loss didn't go down quickly and it went down at a very slow pace. Its loss is about the same in the first 10 epochs, and it starts to go down a little after 15 epochs. This behavior is very different from the CNN-based Image classifier that we trained before.

I tried to use a bigger batch size to train, but it's not any faster and the loss is actually higher than small batch size in the same epoch. I think it's because I used pre-train weights to start with, and it's already very good at image classification. All it needs is some small update on fewer class dataset. Thus, smaller batch size which makes model update more frequently is actually better in this scenario.

I also tried to use a bigger learning rate to update the model, but none of these experiments show a big difference in the training process.

2. Visualize position embeddings (20%)

a. (15%)Visualize cosine similarities from all positional embeddings.



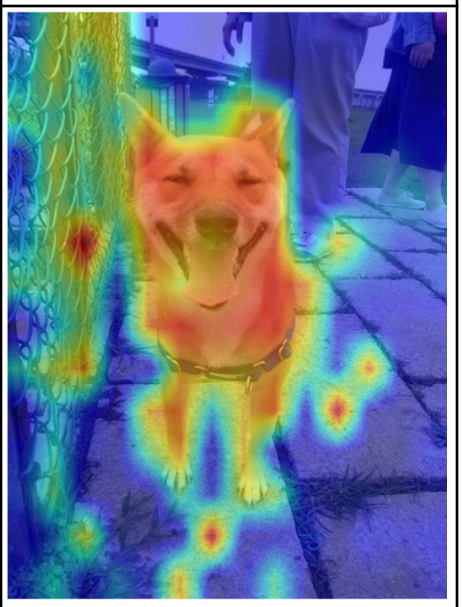
b. (5%)Discuss or analyze the visualization results

The purpose of positional embeddings is to encode position information into vectors. The input image is split into several small patches, and each patch corresponds to a positional embedding, indicating the position information of the patch.

As the above figure shows, positional embedding has highest similarity to itself, and has high but lesser similarity to its neighborhood. As for image patches that are far from itself, their similarities is further lesser than its neighbor patches. All of these facts are quite straightforward and can be easily observed from the figure.

I also find similarities of embeddings are higher, if they’re in the same row or same column. It means the transformer understands the array structure of an image, and knows patches in the same row and column are usually considered sequentially. This trait makes positional embedding of the same row and same column become “closer” than others.

3. (20%)Visualize attention map of 3 images

26_5064.jpg	29_4718.jpg	31_4838.jpg
		

b. (5%) Discuss or analyze the visualization results

Most part of the attention map are reasonable. The attention focuses on the object we want to classify; however, there are some background locations that get strong attention which seems to be a weird thing. I had tried several ways to eliminate that noise. I used a pre-trained model only, but the attention map looks about the same in those locations. I also tried showing only one head output and only one grid to cls, but all those attention maps still have the same problem. Thus, my deduction is that those noises are coming from the pre-trained model I started to train with.

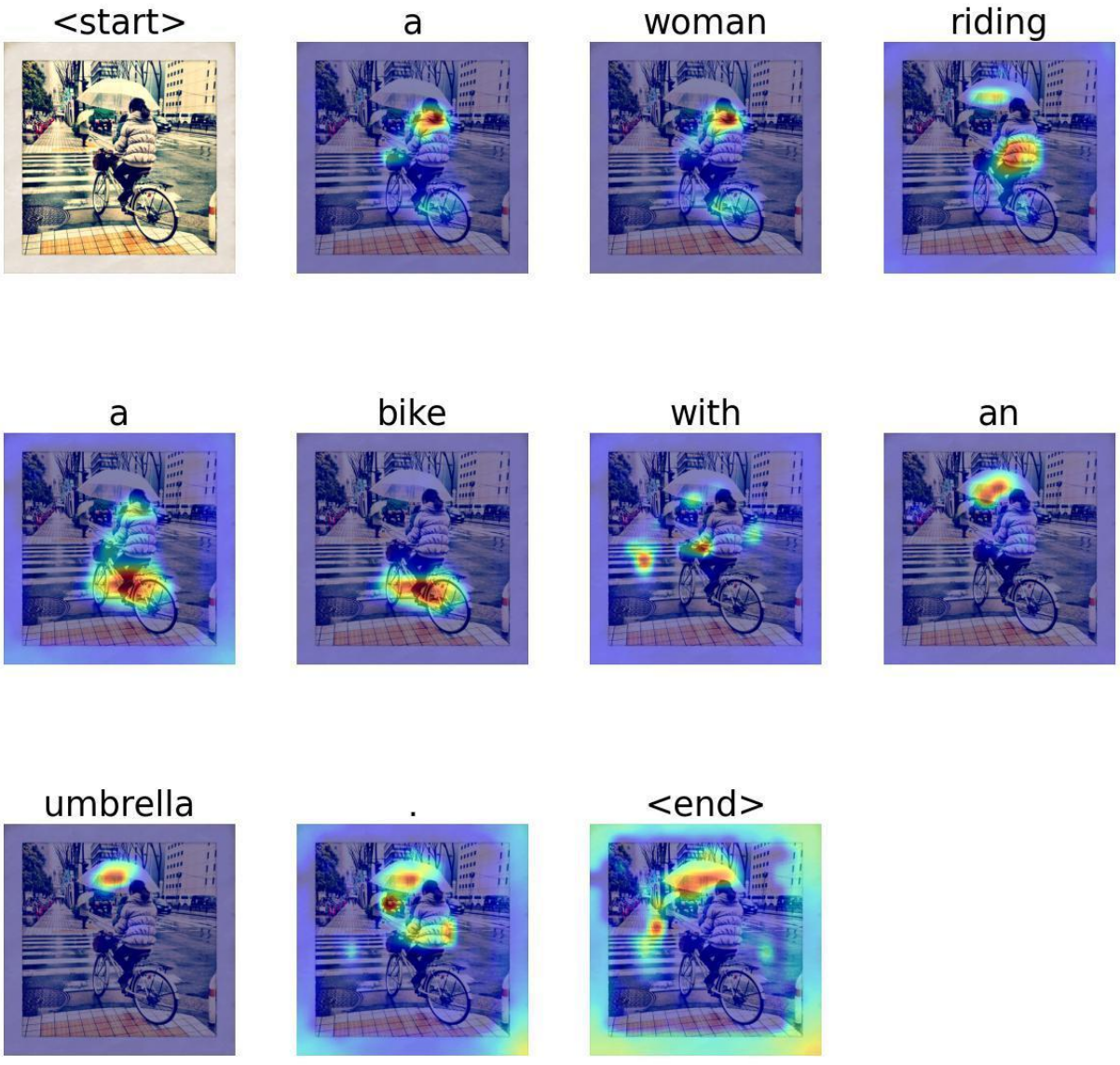
The Timm provided pre-trained model is trained on the ImageNet dataset, which has 1000 classes to classify. However, our ViT only needed to be trained in 37 classes, which is a much smaller dataset. Although I've fine tuned the pre-trained model, I believe the model

inherits the ability to classify 1000 classes. This could explain the weird attention peak on the background of the image, since it might actually have some important features, but it's not used in our 37 classes dataset.

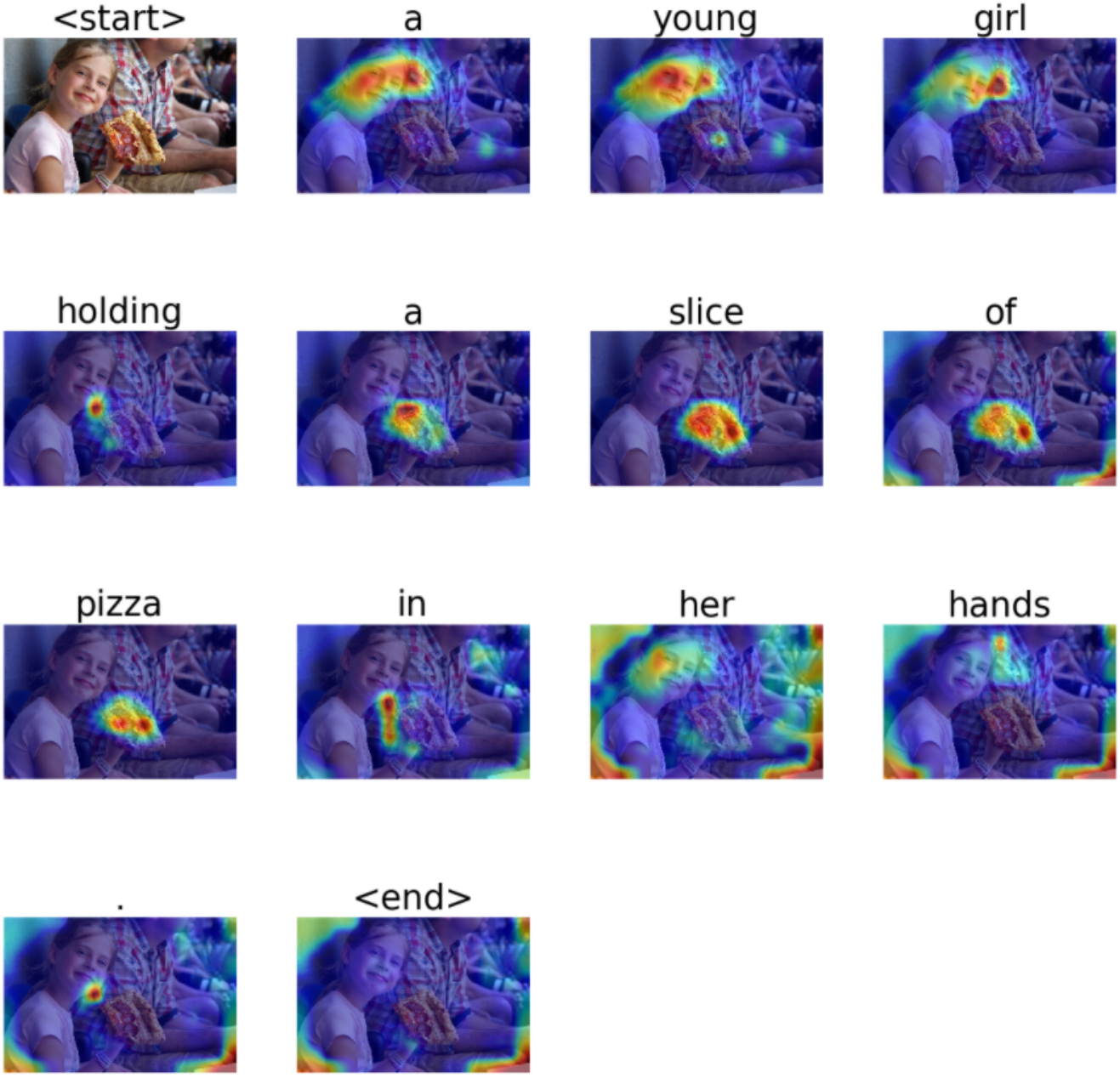
Problem 2: Visualization in Image Captioning (20%)

1. (10%)For the five test images, please visualize the predicted caption and the corresponding series of attention maps in a single PNG.

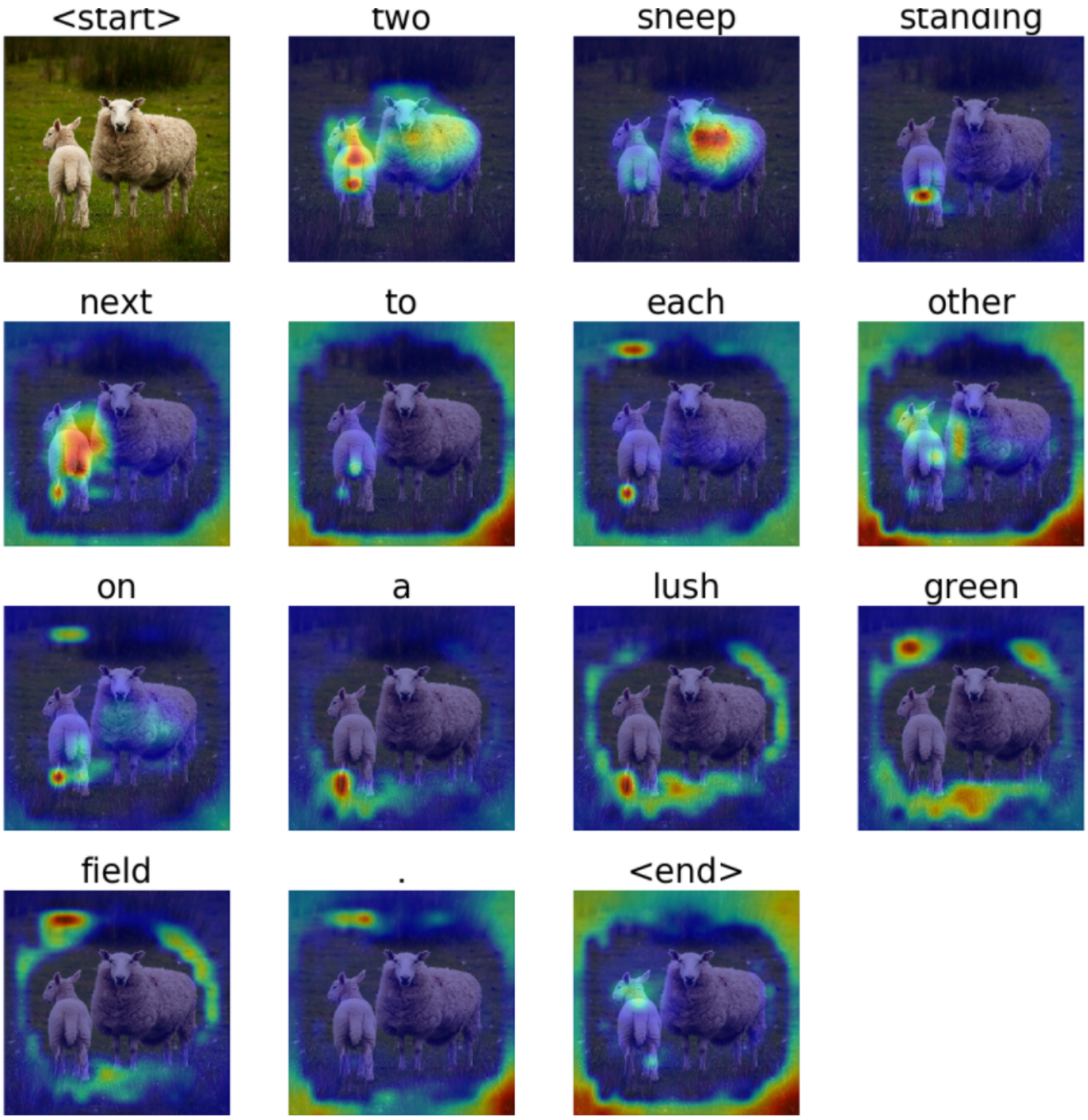
Bike.jpg



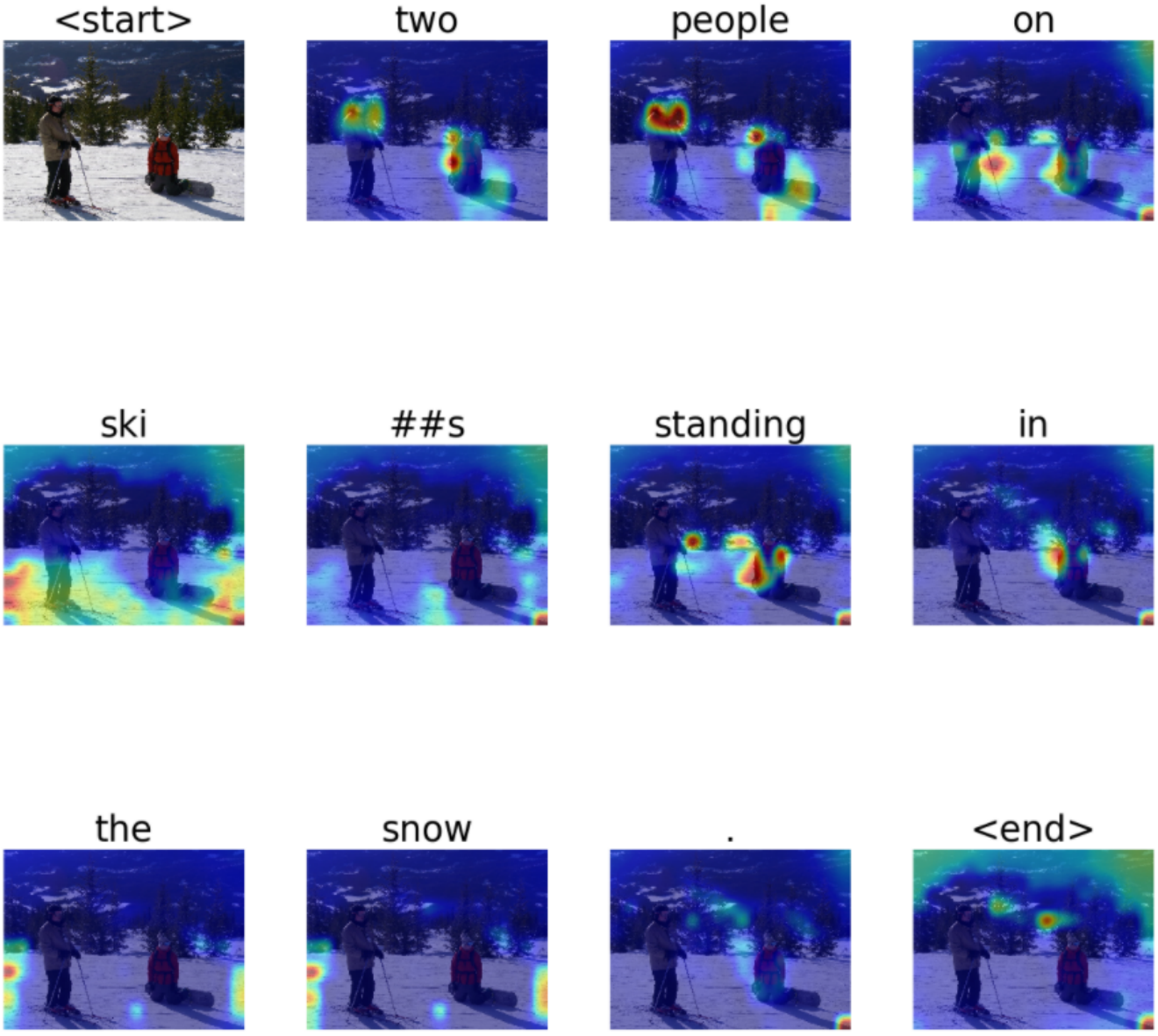
girl.jpg



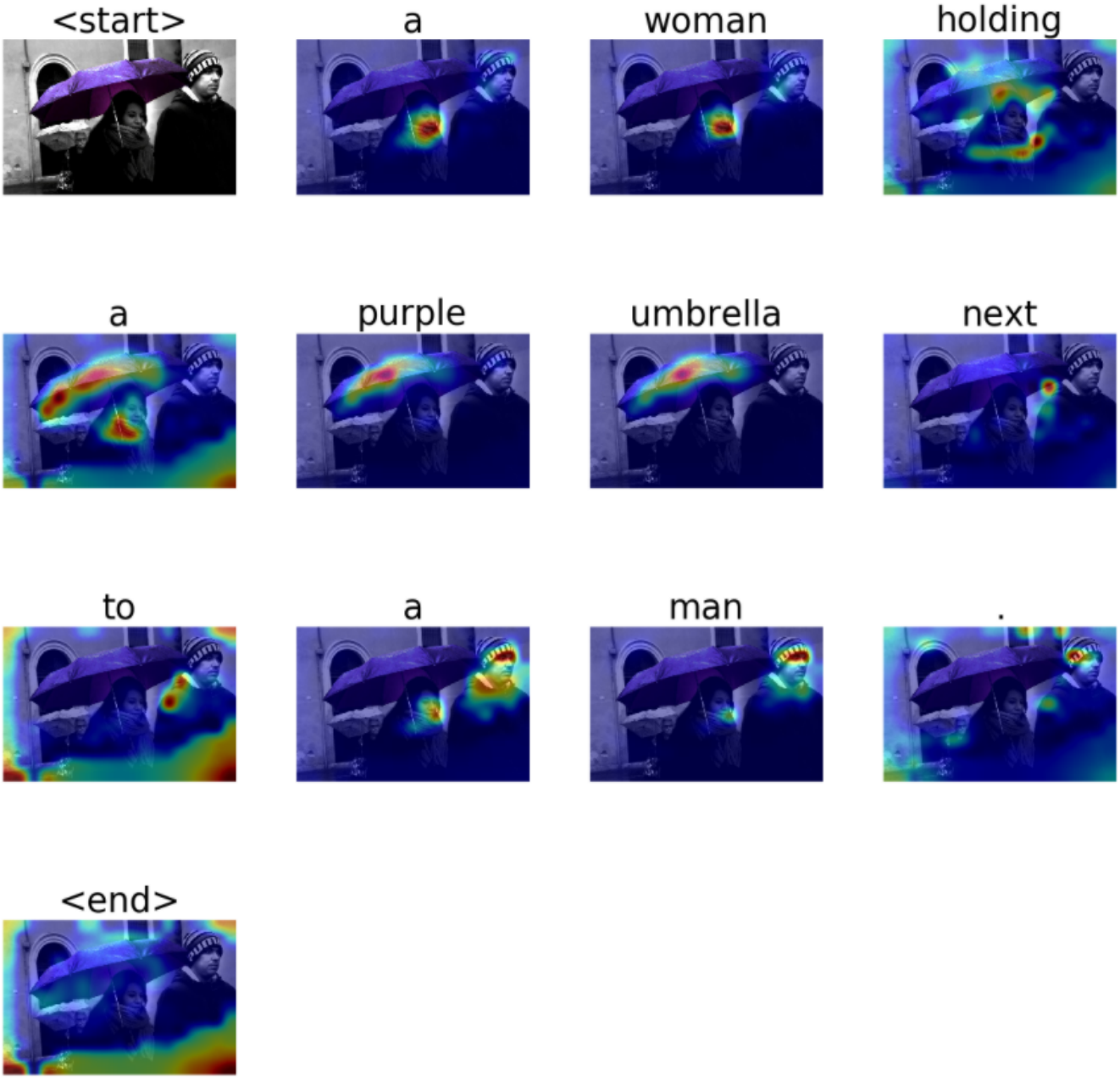
sheep.png


<start>
two
sheep
standing
next
to
each
other
on
a
lush
green
field
.
<end>

ski.png

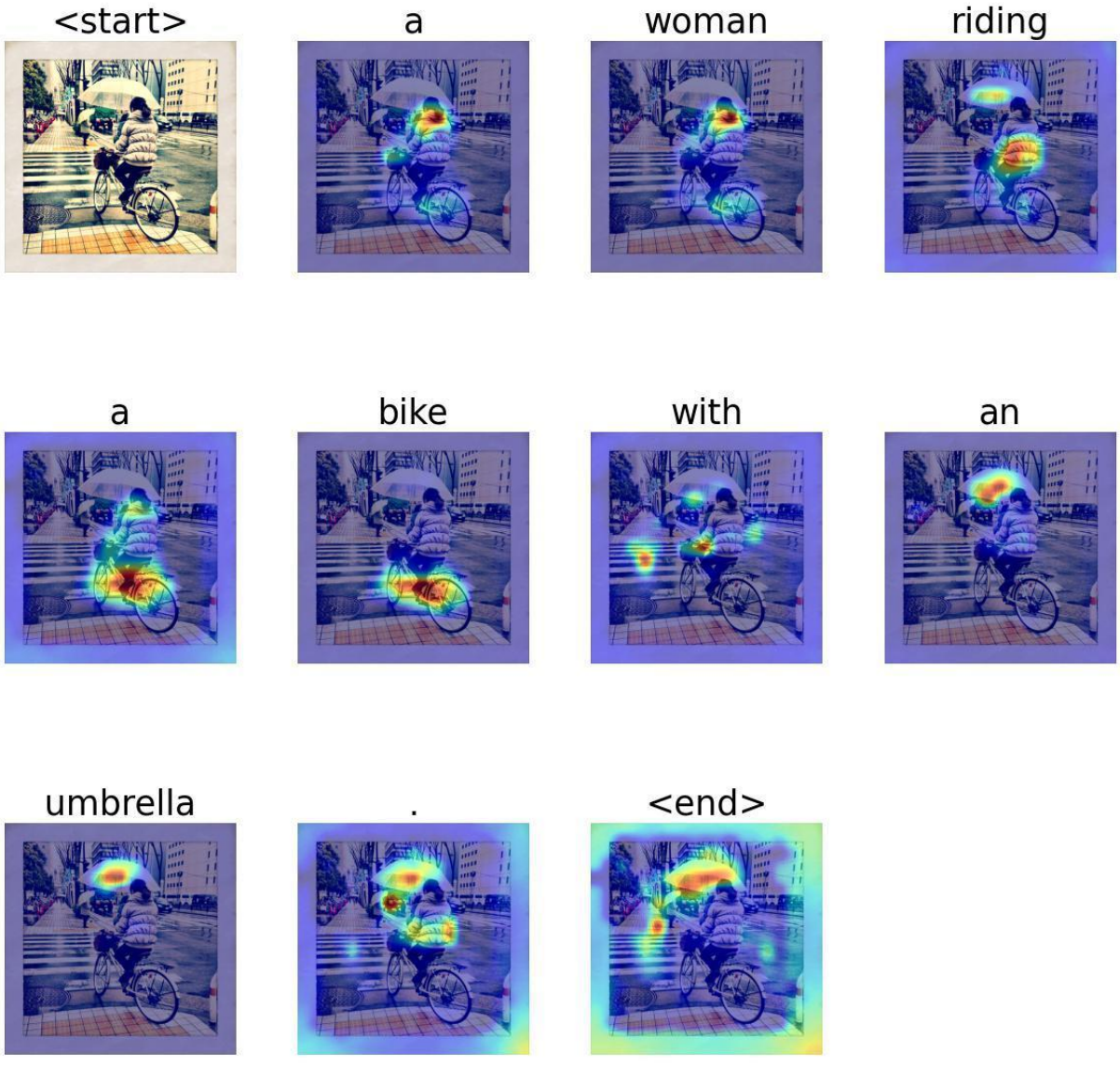


umbrella.jpg



2. (10%)Choose one test image and show its visualization result in your report.

Bike.jpg



Word by word analysis in bike.jpg

A: It's an article of a woman, thus, its attention is laid on the woman and the bike to make sure it's a single instance.

woman: It focus on the hair of the woman, showing machine actually use hair style to check sex of a human.

riding: It paid attention to the woman's hip and thigh, checking the riding gesture of the woman.

a: It's an article of the bike, so its attention lay on the bike and the woman, checking there's only one entity.

bike : It focuses on the gears and the structure of the bike.

with: It's a preposition which indicates an abstract concept, thus, it's attention is much more scattered. However, it did focus on the woman's hand, showing that ViT understands the woman is holding an umbrella "with" her hand.

an : An article on the umbrella, making ViT focus on the umbrella to check the number of instances.

umbrella : ViT's attention focuses on the umbrella, making sure it's an umbrella.

. : the period shows the end of the sentence. Since it's an abstract concept, I think it's hard to explain; nevertheless, I think ViT is paying its attention back to the instances it checked before, making sure that it didn't miss any other objects that needed to be described.

To sum up, I think all of the attention map demonstrates a reasonable result in this experiment. I think the location of attention is very similar

to human perception, hence, once again showing the power of transformers applying on image captioning.

In this homework, I had a hard time locating the cross-attention in Catr's source code. It truly requires lots of background knowledge to trace a deep learning network code. However, once I found the attention map, the rest of the task was quite simple. The attention maps look reasonable and clear without any tuning or adjusting, which is quite a relief. By the way, I also learned that the dimension of a tensor is very useful when tracing the code. I found the attention map by finding the right dimension of a tensor.

Reference

[1] timm

https://github.com/rwightman/pytorch-image-models

[2] Vision_Transformer_Tutorial
https://colab.research.google.com/github/hirotomusiker/schwert_colab_data_storage/blob/master/notebook/Vision_Transformer_Tutorial.ipynb#scrollTo=nI6rRunEO6bI

[3] Visualize attention map

https://github.com/luo3300612/Visualizer

[4] Visualize attention map

https://nbviewer.org/github/luo3300612/Visualizer/blob/main/demo.ipynb