DLCV HW3

R12943010 林孟平

Problem 1: Zero-shot Image Classification with CLIP

1. Methods analysis

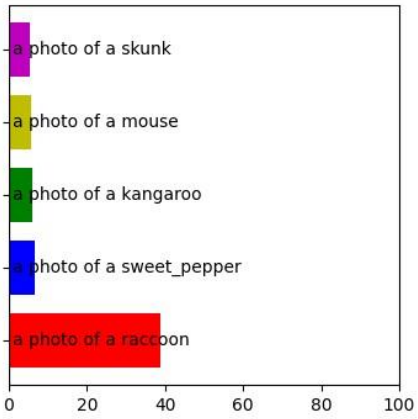
When we use VGG or ResNet to do image classification (by supervised learning). In training stage, we consider the labels as different integers (one-hot vector). The original meaning of the label is ignored. While CLIP model was trained on enormous number of image-text pairs and compare their similarity. In training, the labels are represented as the natural language, which is meaningful and contain high-level information. In zero-shot task, even the model sees something that is not in the training data, it can still grasp some sematic meaning from the text and try to do image classification. We can then narrow the gap between the pre-trained and downstream task without directly fine-tuning the model.

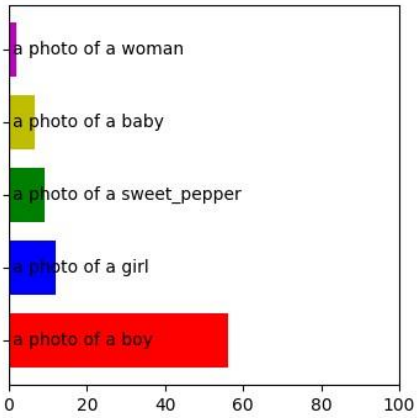
2. Prompt-text analysis

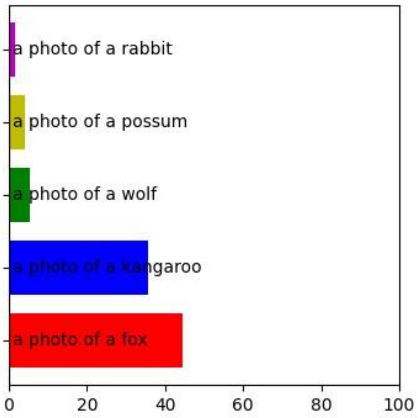
Type1	**This is a photo of {object}**	**60.9%**
Type2	**This is not a photo of {object}**	**65.4%**
Type3	**No {object}, no score.**	**56.4%**

We can see that different prefix gives us difference classification accuracy on validation set. The surprising thing is that as the antonym sentence, the performance of type2 is the best among three. Maybe the sentence of type2 are more common in the pre-trained data and CLIP model thus has better classification accuracy when comparing image-text pair.

3. Quantitative analysis







Problem 2: PEFT on Vision and Language Model for Image Captioning

1. Report your best setting and its corresponding CIDEr & CLIPScore on the validation data.

Base model:

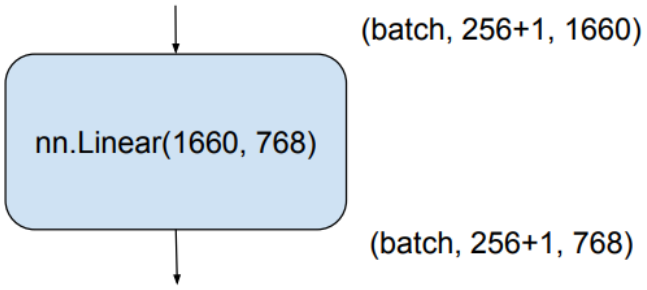
Encoder: “vit_gigantic_patch14_clip_224” from timm

Decoder: from decoder.bin

Cross-attention layer (Direct fine-tuning):

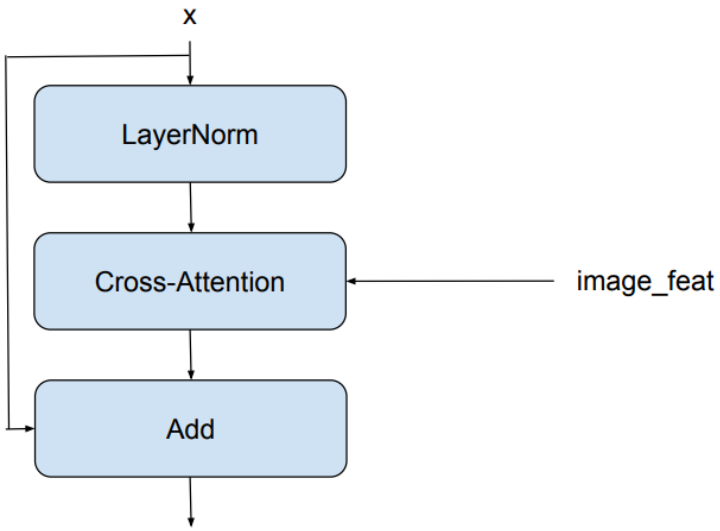
In Decoder:

To convert the feature of image to the same dimension as text-embedding, I first use a linear layer to convert image feature to the same dimension as text feature and then send to cross attention layers.

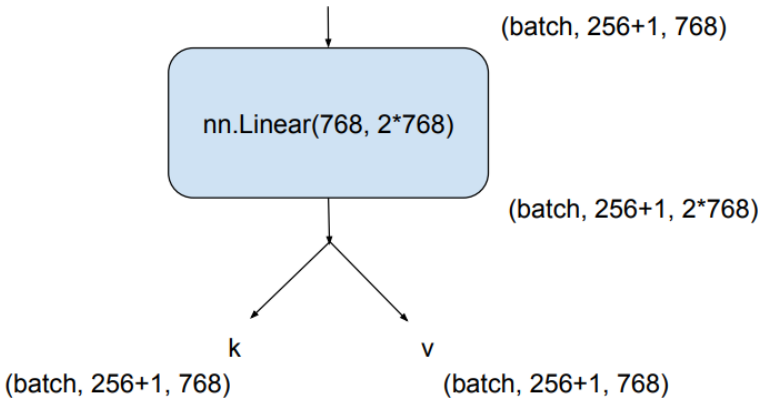


In each block:

I add a layer norm before doing cross-attention.



The implementation of cross attention is similar to self-attention layer, including the hidden size of embedding, number of multi-head and number of layer. In each layer, I use another linear layer to convert dimension-reduced image feature to key, value and then do cross attention computation.



Best setting PEFT method: LORA with r =16

Self-attention layer (LORA PEFT):

(a) In each block of the mlp layer, the “c_fc” layer and “c-proj” layer are replaced as LORA linear layer.

(b) In the self-attention layer of each block, “c_attn” layer and “c_proj” layer are replaced as LORA linear layer.

Hyperparameters:

Batch Size = 8

Learning rate = 3e-5

Total epoch = 10

Optimizer: AdamW

Scheduler: Cosine Annealing with eta_min = 1e-6

Auto-regressive method: Greedy Search with max length = 60

Data Augmentation:

Resize(224, 224)

Normalization with mean = [0.485, 0.456, 0.406], std = [0.229, 0.224, 0.225]

Random Horizontal Flip with p = 0.3

ColorJitter with default parameters

CIDEr	**0.907441**
CLIPScore	**0.733105**

2. Report 3 different attempts of PEFT and their corresponding CIDEr & CLIPScore.

The base model setting and hyperparameters for the following PEFT are all the same as the previous one.

(1) LORA with r = 8:

The model structure is all the same as my best setting except the rank in LORA layer.

CIDEr	**0.872251**
CLIPScore	**0.727208**

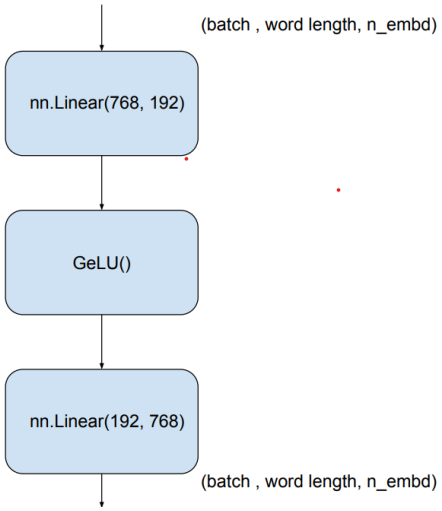
(2) Adapter

I add an adapter at

(a) the last layer for each self-attention layer.

(b) the last layer for each cross-attention layer.

Adapter model:



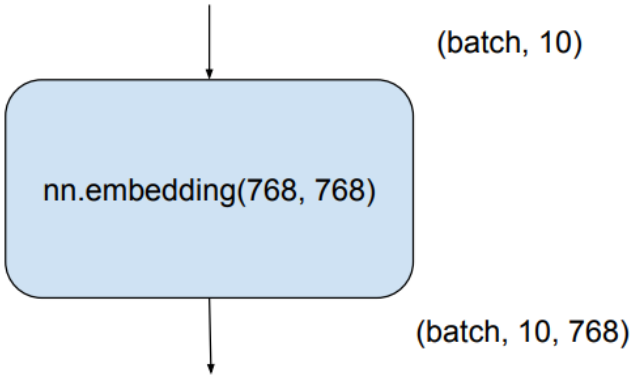
CIDEr	**0.843354**
CLIPScore	**0.713858**

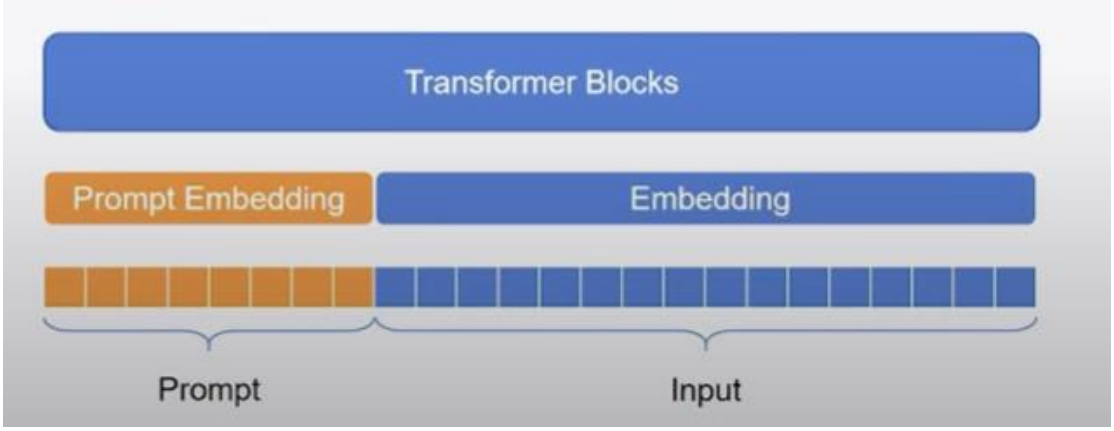
(3) Prompt tuning

I add a soft prompt with random number. It will be added at the front of the input after passing the embedding layer.

Token length = 10

Number of different virtual tokens = 768



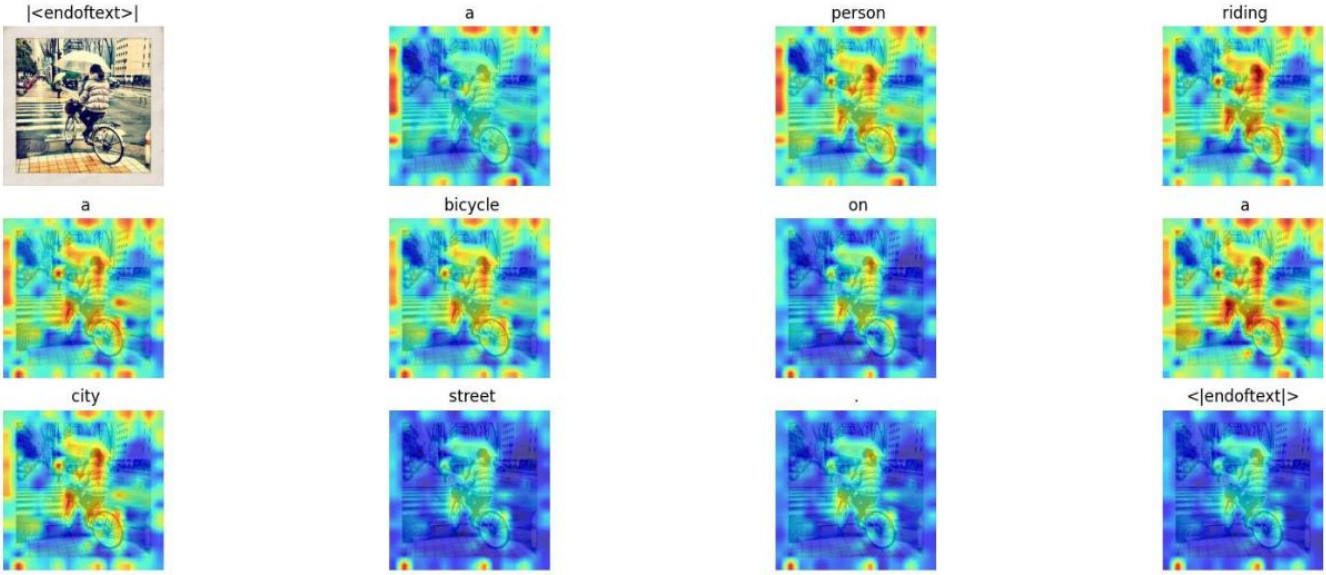


CIDEr	**0.862951**
CLIPScore	**0.725568**

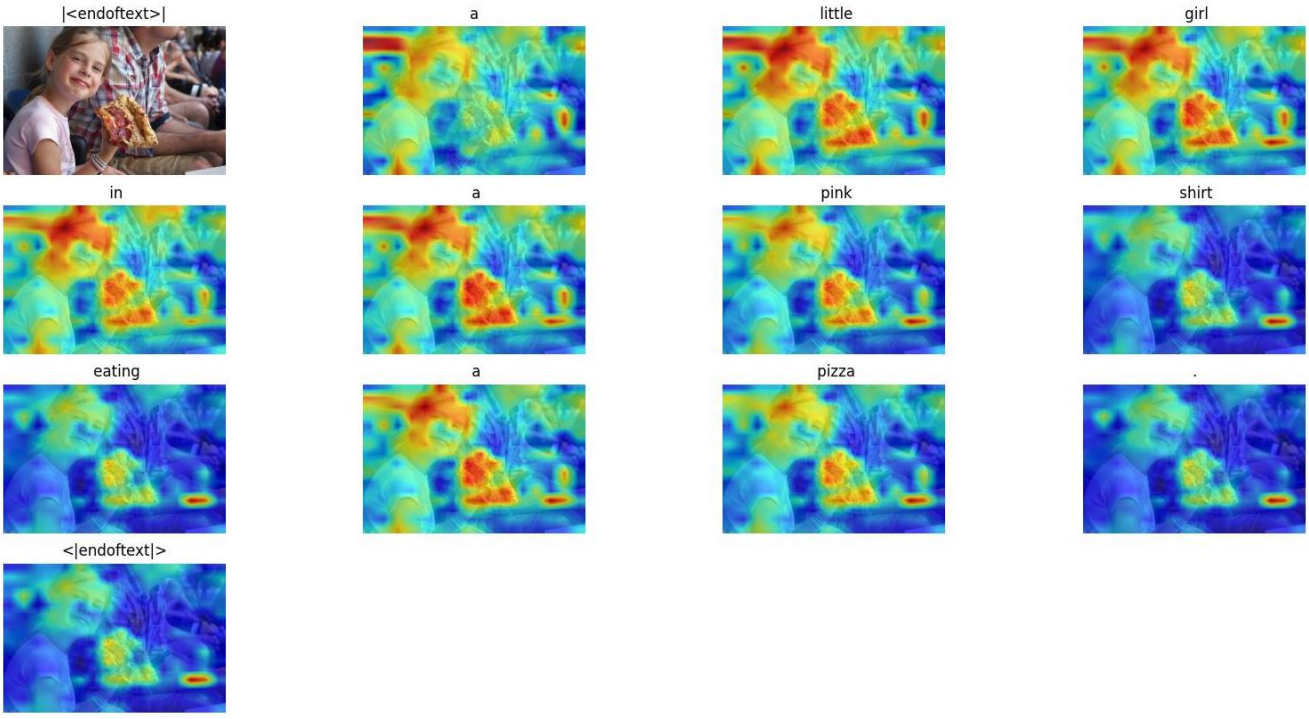
Problem 3: Visualization of Attention in Image Captioning

1. TA will give you five test images ([p3_data/images/]), and please visualize the predicted caption and the corresponding series of attention maps:

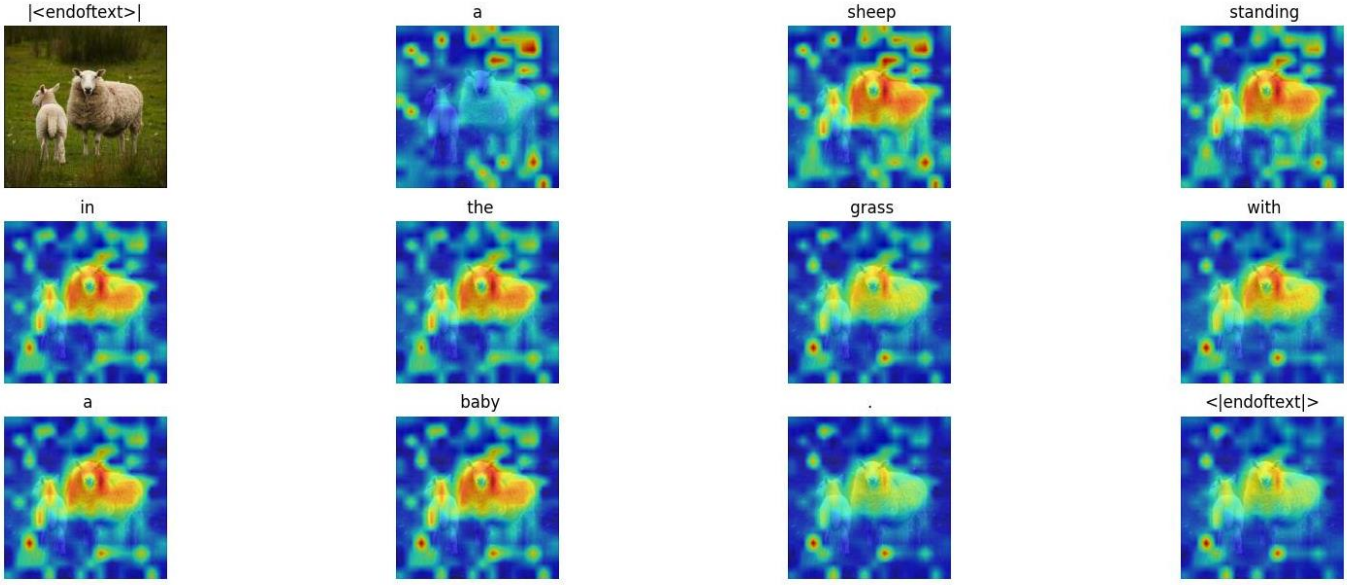
bike.jpg



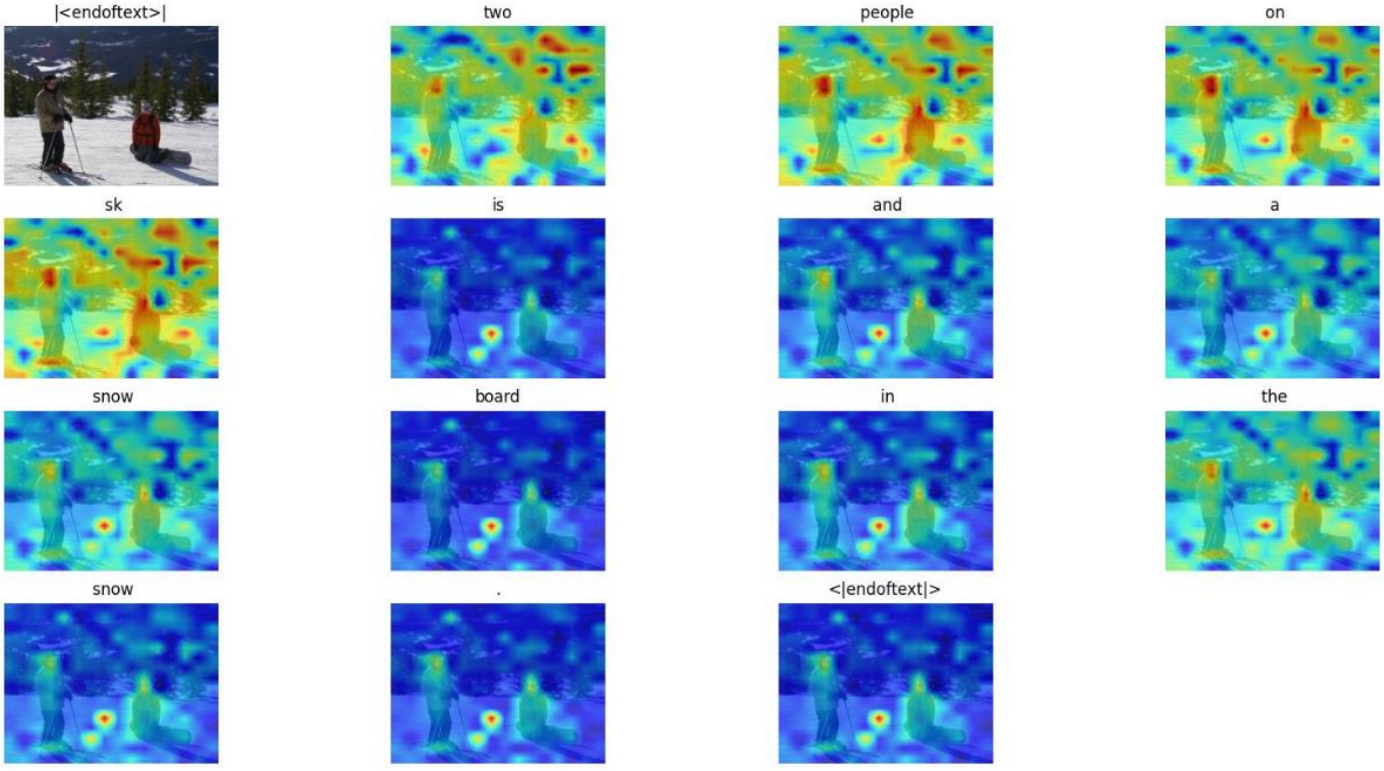
girl.jpg



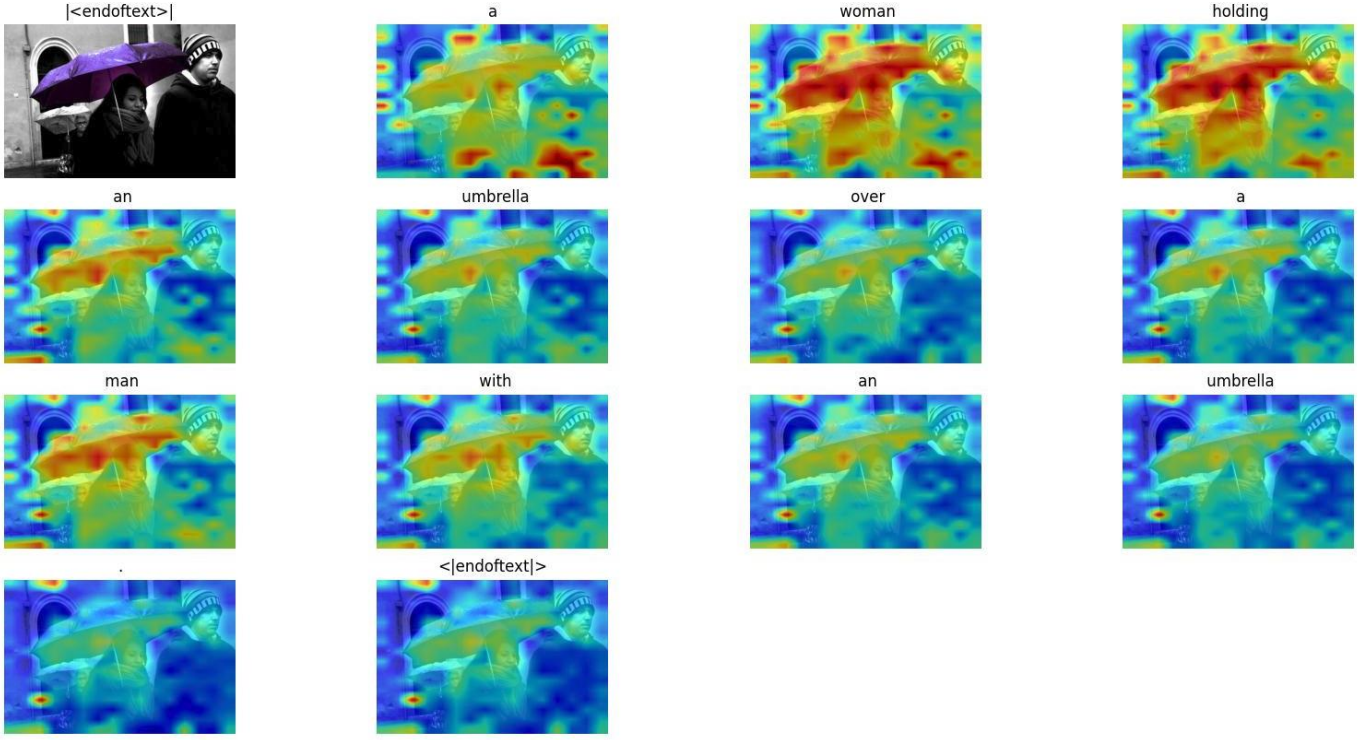
sheep.jpg



ski.jpg

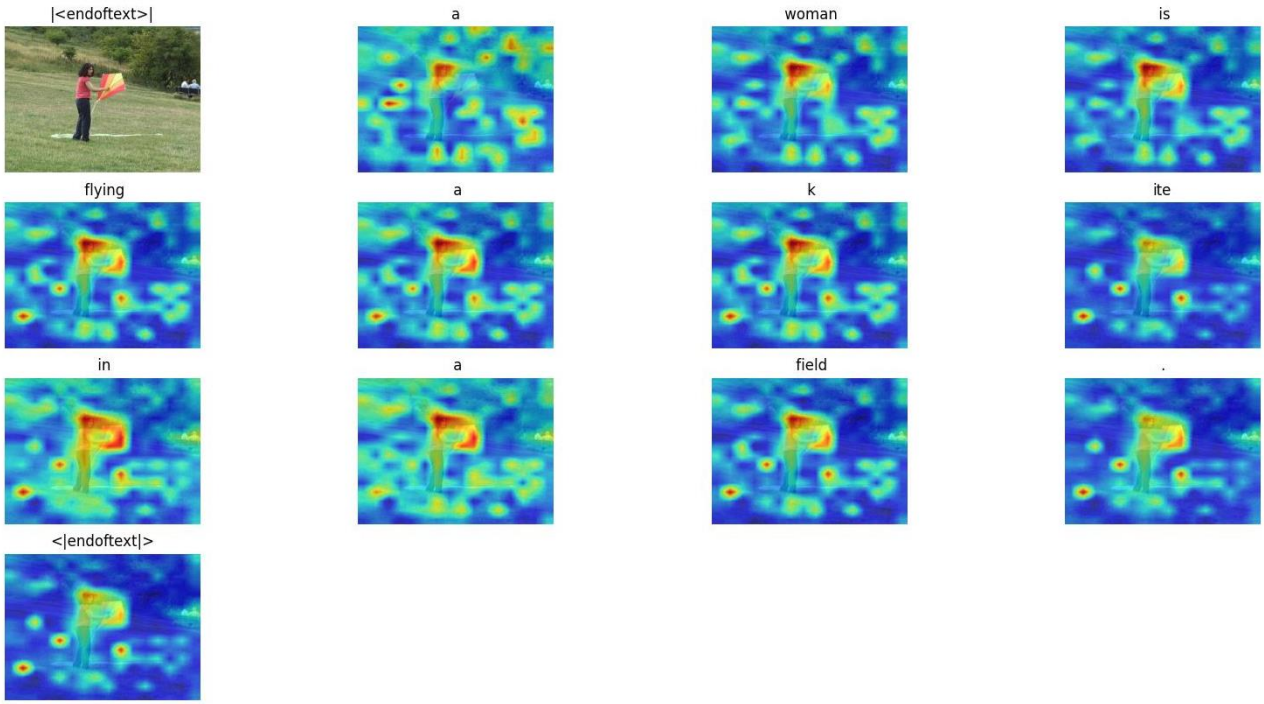


umbrella.jpg

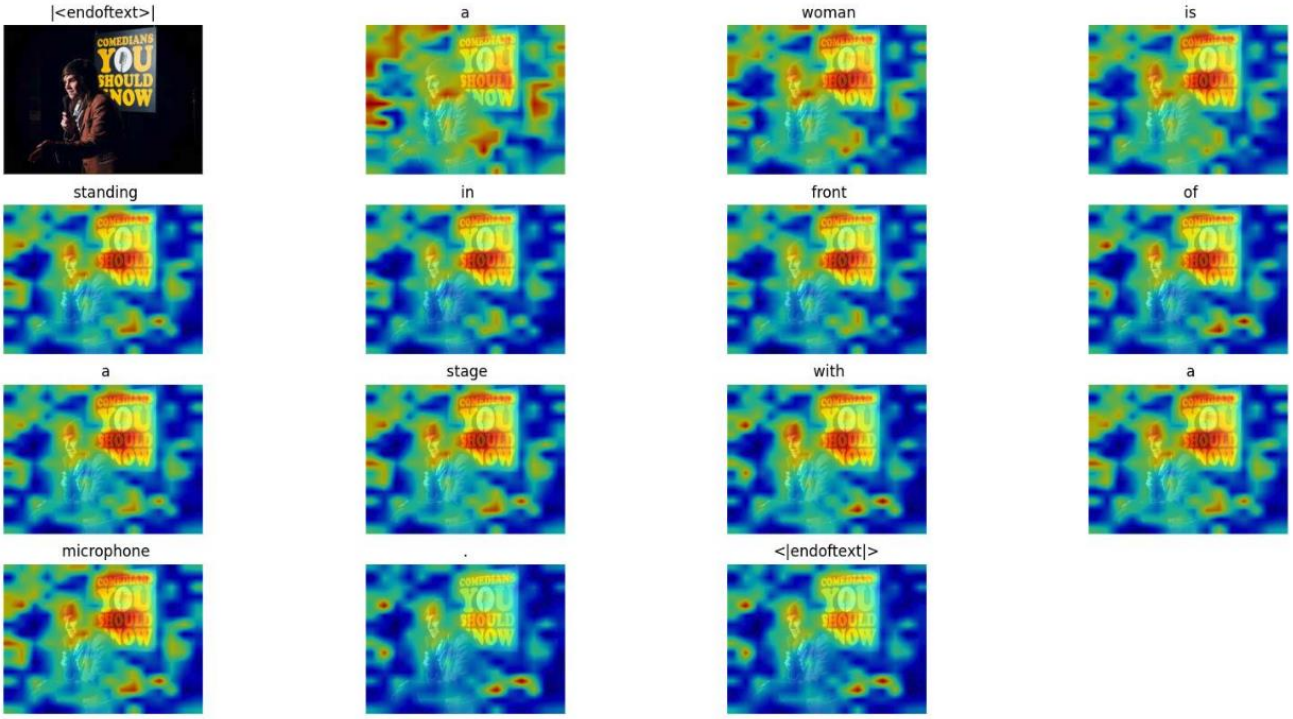


2. According to CLIPScore, you need to visualize top-1 and last-1 image-caption pairs and report its corresponding CLIPScore in the validation dataset of problem 2.

Best: 000000179758.jpg CLIPScore = 0.9979248046875



Worst: 4406961500.jpg CLIPScore = 0.42388916015625



3. Analyze the predicted captions and the attention maps for each word according to the previous question. Is the caption reasonable? Does the attended region reflect the corresponding word in the caption?

For bike.jpg:

Most of the caption-patch matchings related to the person are reasonable, such as “woman”, “bike”, “riding”, but caption-patch matchings related to the background are unclear.

For girl.jpg:

Most of the patches are related to the girl’s face and the pizza, but the difference between caption-patch pairs are small.

For sheep.jpg:

Most of the patched are related to the adult sheep and the baby sheep, but the difference between caption-patch pairs are small.

For ski.jpg:

Most of the caption-patch matchings related to people are reasonable, and the “snow” caption-patch pair is obvious and clear.

For umbrella.jpg:

The caption-patch pairs related to the woman are reasonable, such as “woman”, “holding”, “umbrella”, but caption-patch matchings related to the man are unclear.

For 000000179758.jpg

The caption-patch pairs related to the woman are reasonable, such as “woman”, “flying”, “kite”, but the difference between caption-patch pairs are small.

For 4406961500.jpg

Most of the patches are related to the woman and the sign behind her, and the difference between caption-patch pairs are small.

Generally, the model can grasp the most important meaning (global feature) in the image and generate reasonable caption. However, we can see that some attention maps cannot exactly match the text of output sentence. It is probably because our vision model is relatively larger. When our decoder model is pre-trained and freeze, the result of visualization might not be as our expectation.