DLCV HW1

工科海洋四 B08505048 劉名凱

Problem 1

1.

Firstly, previous methods such as ResNet, VGG are trained with data that require lots of effort on labeling and are limited to fixed number of classes. On the other hand, Clip is trained with {image, text} pairs which are publicly available and ubiquitous on the Internet. Due to a wide variety of data on Internet and its training method ({image, text} pairs), Clip can learn the image representation and connects the representation with language, establishing a relationship between images and texts, which helps Clip achieve competitive zero-shot performance.

Moreover, Clip is not directly optimized for benchmark's performance, which makes it more representative. As a result, Clip can deal with a variety of datasets without any modification on model.

2.

Using ViT-L/14

	This is a photo of {object}	*This is a {object} image.*	*No {object}, no score.*
Accuracy	0.6752	0.7288	0.458

This is a photo of {object}:

This prompt text is most similar to the recommended prompt text *a photo of a {object}* among three prompt texts, I believe this is why it achieve a good result.

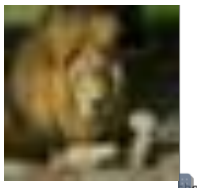
This is a {object} image.:

This prompt text yields the highest accuracy, I think this is because lots of images on internet are describe as *This is a {object}*.

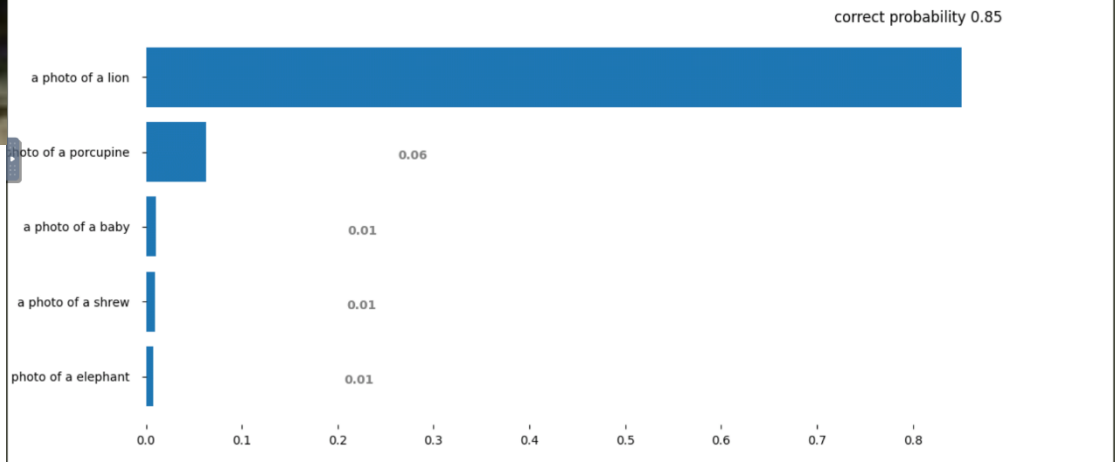
No {object}, no score.:

This prompt text gives worst accuracy among three prompt texts. It's probably because it does not effectively establish a concrete connection between images and languages. A text 'No milk, no score.' does not necessarily be used to describe a photo of milk, in fact, I think it has nothing to do with a photo of milk.

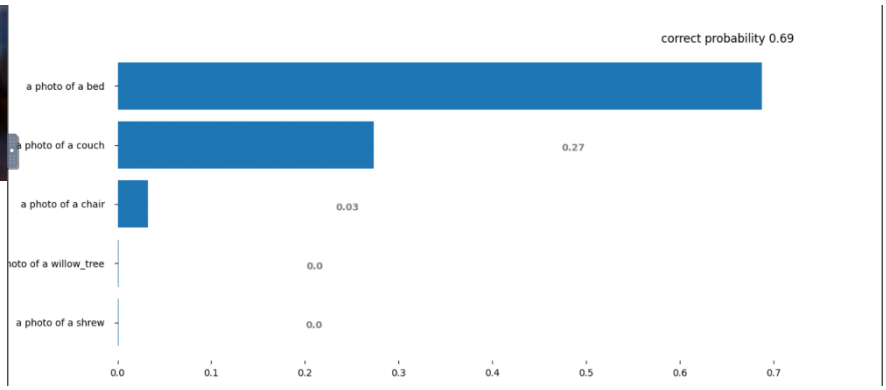
3.

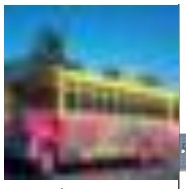


Lion

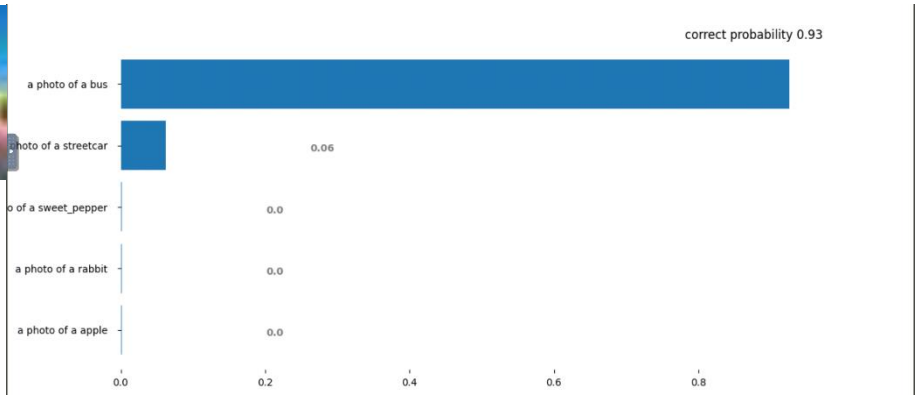


bed





bus



Problem 2

1.

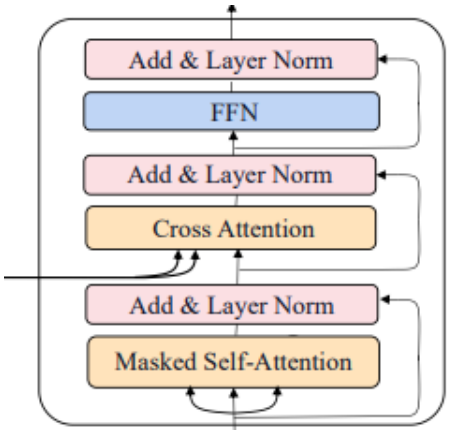
Encoder:

I used the pretrained vision transformer provided in the timm as image encoder. The version of the pretrained ViT is *vit_large_patch16_224_in21k.*

Decoder:

Decoder contains 6 decoder blocks, where decoder blocks is same as those in paper *CPTR: FULL TRANSFORMER NETWORK FOR IMAGE CAPTIONING.* The head number in multi-head attention is 8.

Decoder output is fed into a Linear layer to predict the probability of each token.



Decoder block

Detail:

Batch size = 32

Learning rate = 1e-4 (multiplies by 0.1 every 20 epoch)

Embedding size = 384

Input images resolution = 3 x 224 x 224 (normalized to mean = 0.5, std = 0.5 in every channel)

Loss function = Cross Entropy Loss (without ignoring padding token)

CIDEr = 0.84, CLIP = 0.764

2.

I encountered numerous problems when dealing with image captioning vision transformer and spent total two weeks on finding suitable models for this task. Below are three major model settings I have tried.

1. pretrained Levit + CPTR decoder (CIDEr = 0, CLIP = 0)

In the beginning, I used a pretrained vision transformer in timm called Levit. The team that proposed Levit claims that it provides efficiency and accuracy simultaneously, which is why I chose it. I implemented my own decoder which was exactly same as the decoder in the paper *CPTR: FULL TRANSFORMER NETWORK FOR IMAGE CAPTIONING.* After training for 30 epochs, loss became almost 0 and I thought I succeed. But I found out that my model is simply copying captions I fed to it (I was told that I trained my model in a wrong way when I ask for TAs help). Therefore, when inferencing, the model can only predict EOS. Every predicted sentence is [EOS, EOS, EOS, EOS…]. As a result, both CIDEr and CLIP score were 0.

2. pretrained Resnet101 + Transformer (CATR: **Ca**ption **Tr**ansformer) (CIDEr = 0.62, CLIP = 0.63)

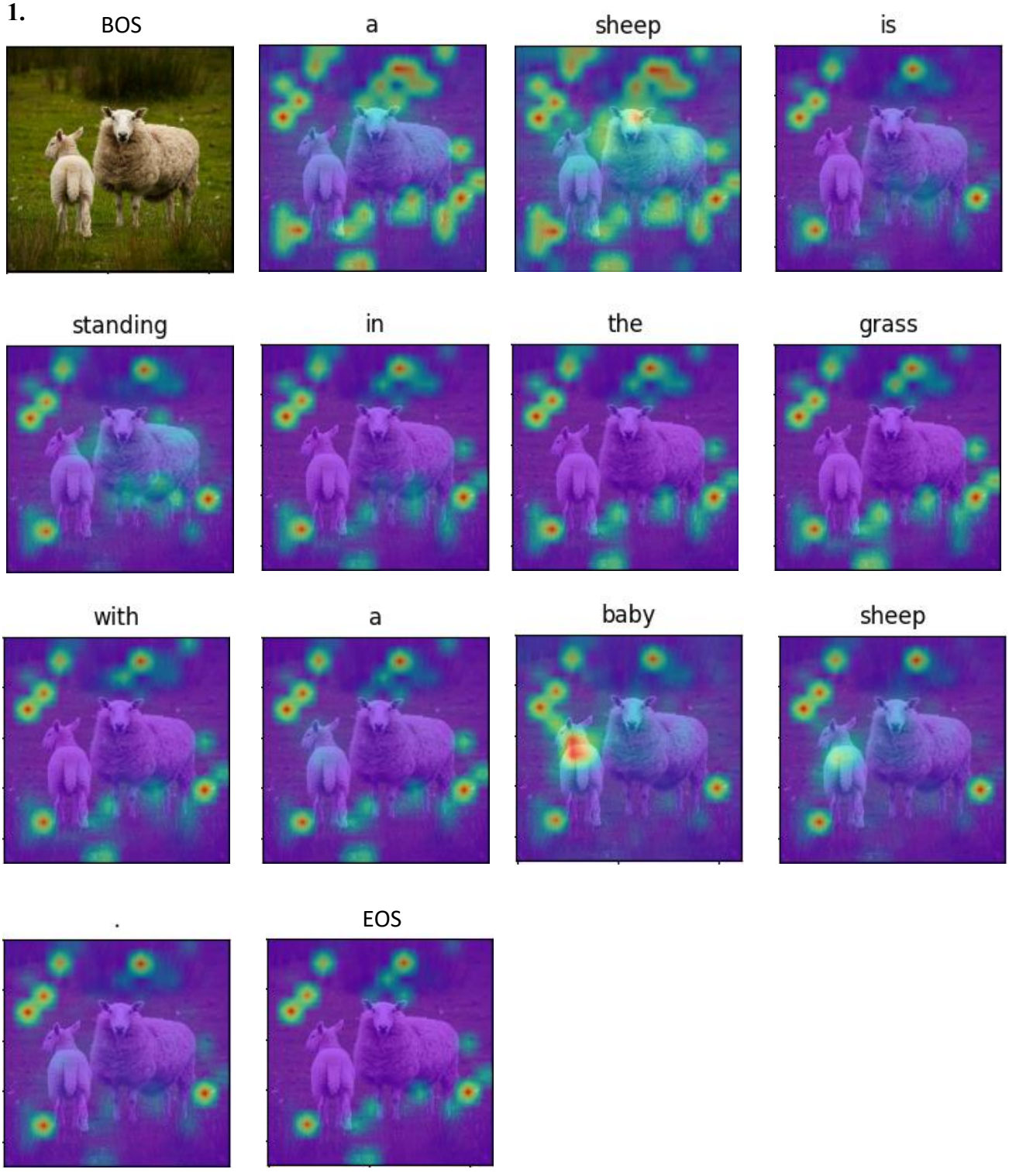
I tried this method since Levit + CPTR decoder didn't yield satisfactory result even after I fixed the bug. I found this open source code on Github and gave it a try. For this setting, I trained the transformer from scratch. But after several tries, the best score I got was CIDEr = 0.62, CLIP = 0.63, which was not enough for this assignment. So, I gave up this setting.

3. petrained ViT + CPTR decoder (CIDEr = 0.84, CLIP = 0.76)

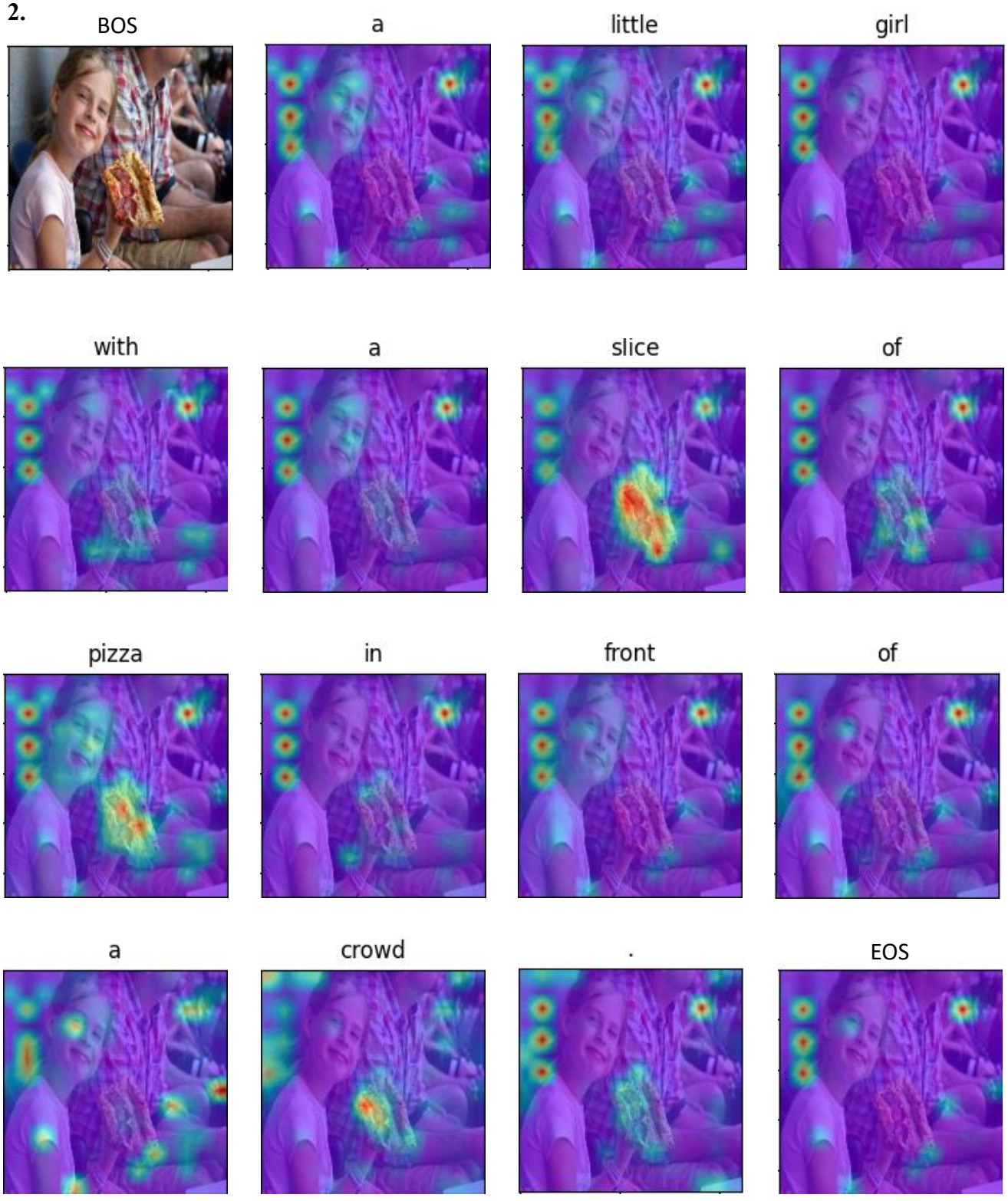
I used the pretrained Vision Transformer in timm which was trained on ImageNet 21k dataset as the image encoder. Then used the CPTR decoder. I freezed the pretrained Vision Transformer when training since I found that model's performance dropped without freezing it. After 50 epochs of training, CIDEr = 0.84, CLIP = 0.76. I also tried beam search when inferencing. However, beam search predicted longer but unreasonable sentence, such as repeating sentence structure or wrong grammar, so I gave up.

Problem 3

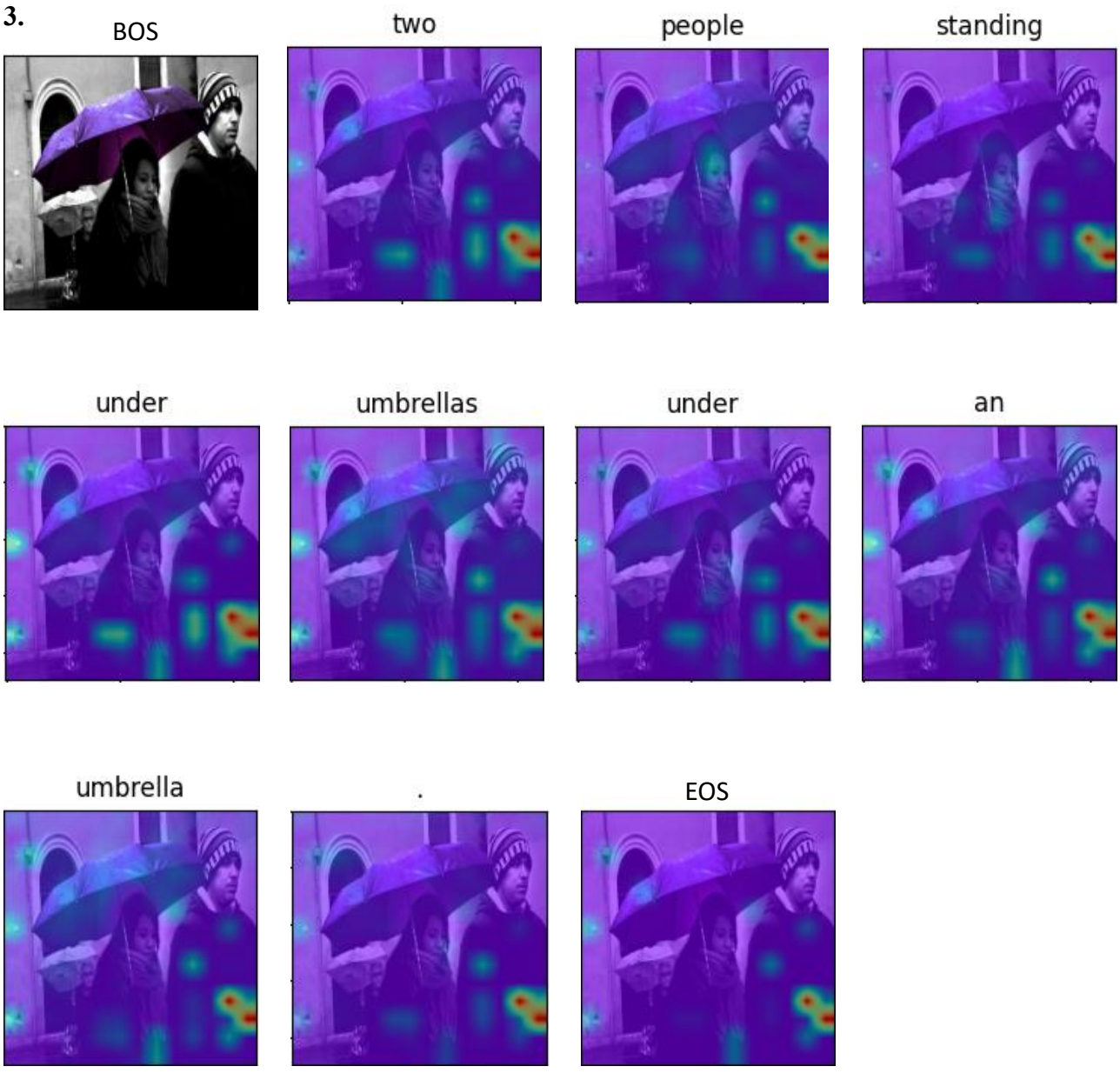
1.



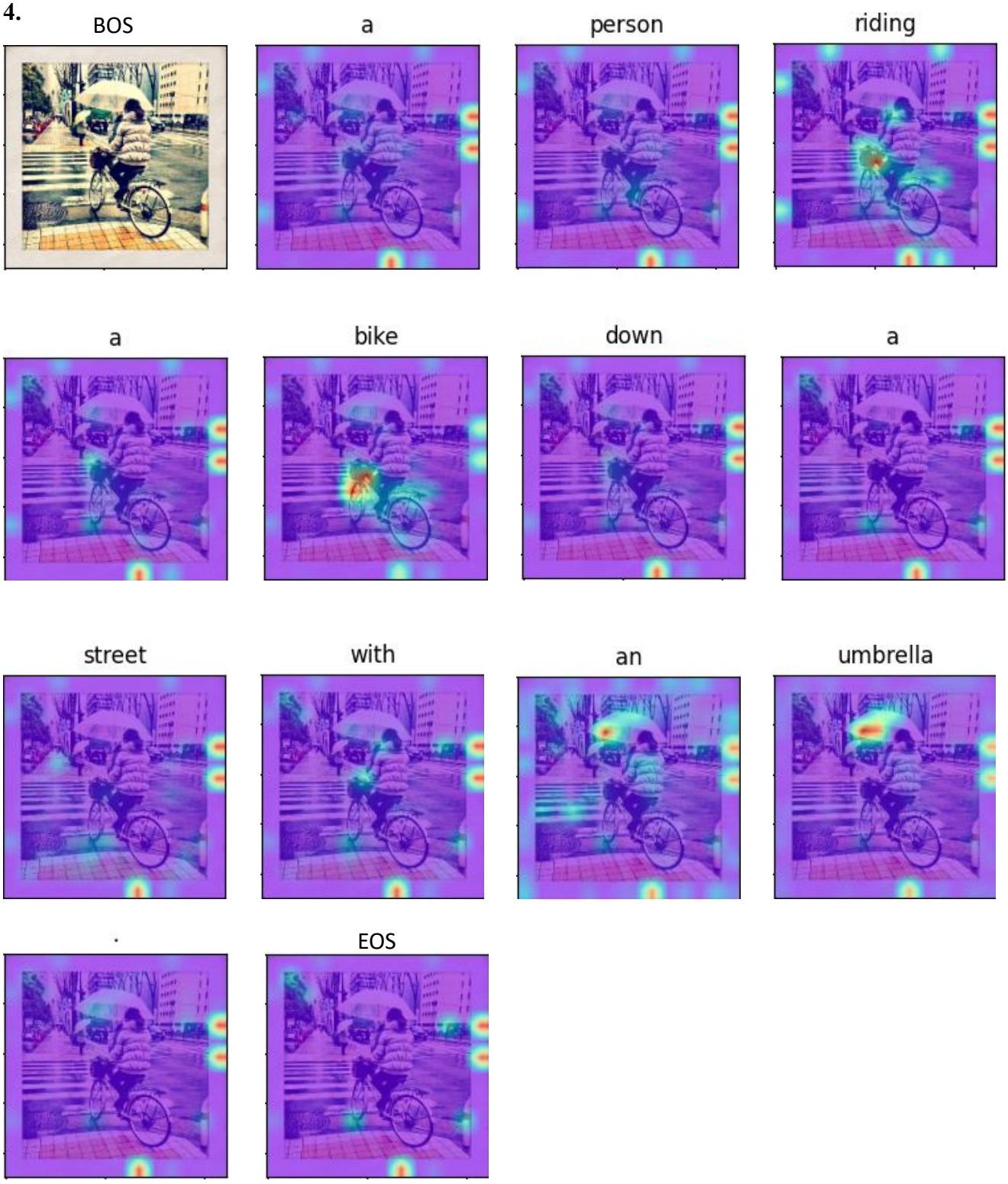
2.



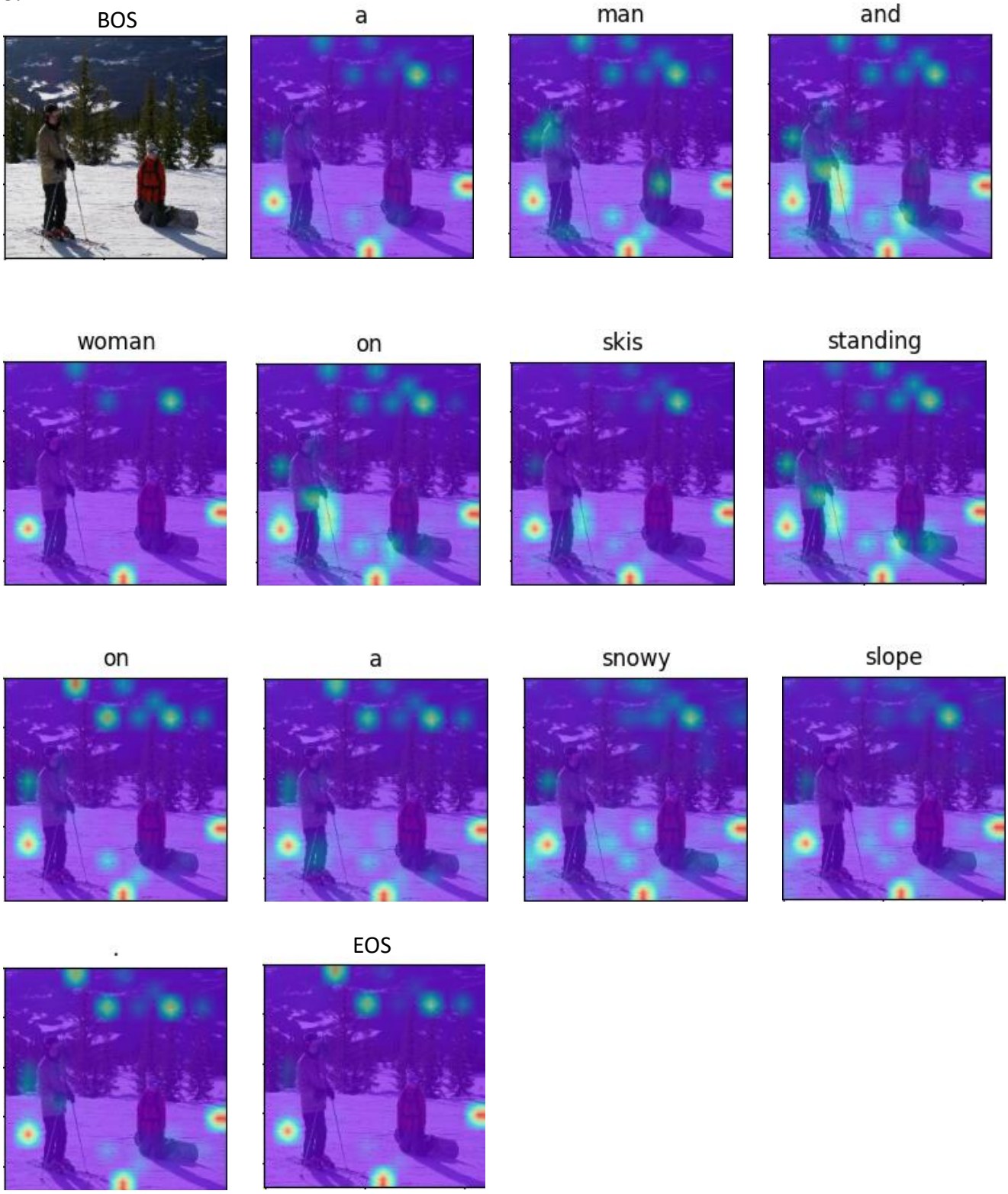
3.


BOS
two
people
standing
under
umbrellas
under
an
umbrella
.
EOS

4.

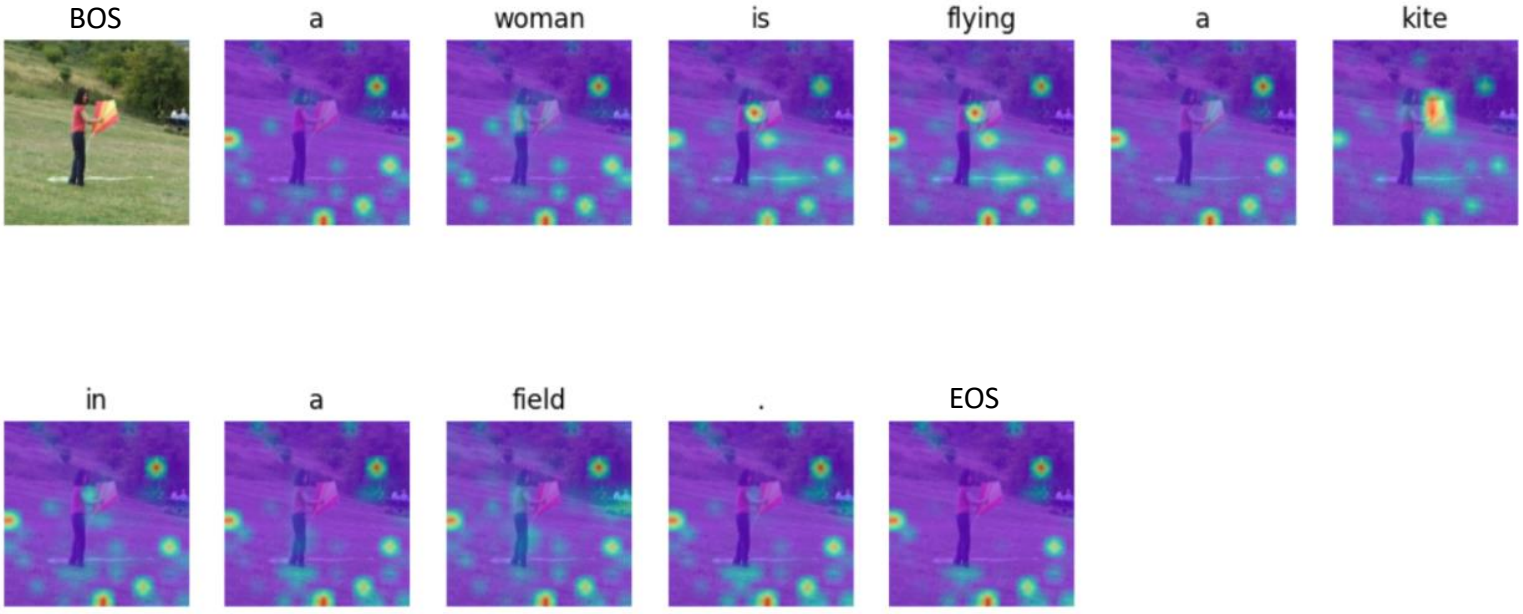


5.

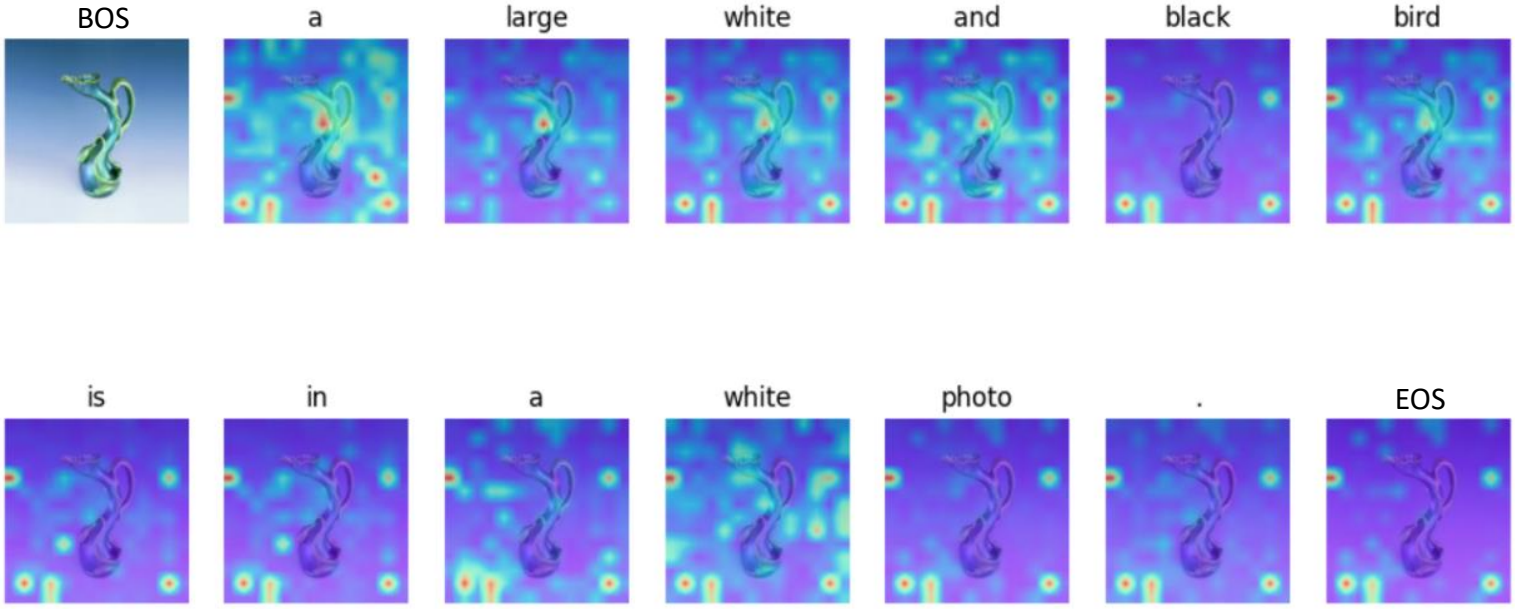


2.

Top 1: 000000179758.jpg (CLIP score = 0.998)



Last 1: 000000406155.jpg (CLIP score = 0.396)



3.

In top-1 image-caption pair, predicted caption precisely describe the photo and the attended regions for words that describe objects such as '**woman**', '**flying**' and '**kite**' indeed represent the meaning of word. However, for words that are meaningless when present along such as '**a**', '**is**' and '**in**', attended regions doesn't give much clues why these words are generated. I think this is because 'a', 'is' or other words used to

make sentences look more grammarly-corrected are linguistic features, which have little to do with image. On the other hand, ‘kite’, ‘woman’ or other words describe objects are strongly related to the image, which is why the attended regions reflect the corresponding word.

In last-1 image-caption pair, the predicted caption is not reasonable at all. The model classifies the object in the photo as a black and white bird. However, for words ‘**white’, ‘black’** and **‘bird**’, the attended region does focus on the object, the only problem is that the object in the photo is not a bird.