DLCV Fall

醫工四 郭思言 B08508002

Hw3

Part 1:

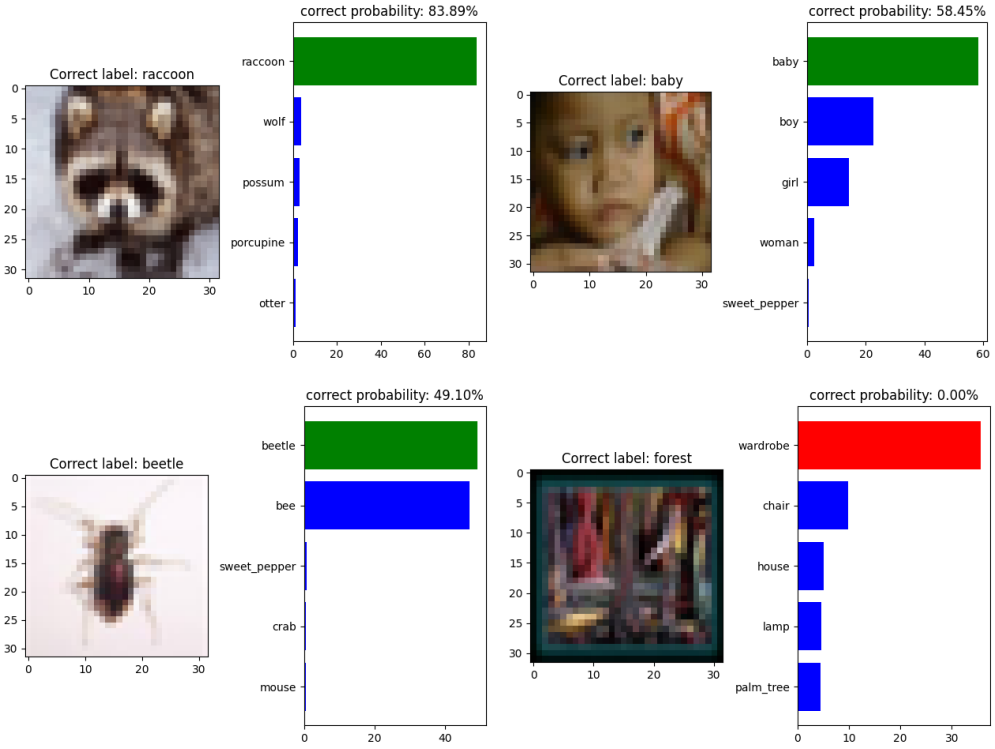
1. Method analysis: Why is CLIP good at one-shot classification?

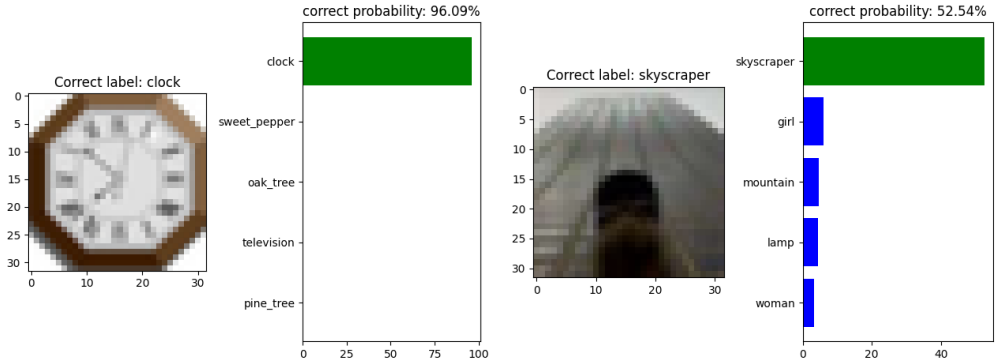
   One-shot means that it can have good results on images that has not been observed during training. I think that CLIP can have good results because it trains on a wide variety of images and texts. Learning such wide range of visual concepts directly from natural language makes it very flexible and general than other image related models.

2. Prompt-text analysis:

Prompt	Accuracy
This is a photo of {object}	0.6088
This is a {object} image.	0.684
No {object}, no score.	0.5648

3. Quantitative analysis:





Part 2:

1. Report my best setting and corresponding CIDEr & CLIPScore on the validation data:
   - For the encoder, I choose "vit_large_patch14_224_clip_laion2b" from timm, and get the features after norm. Therefore, the output shape of the encoder is (1+16x16) x 1024. (1024 is the hidden dimension)
   - For the decoder part, I implement a transformer-based decoder base on the architecture in the paper "An image is worth 16x16 words: Transformers for image recognition at scale".
   - Some parameters:

     Hidden dimension: 1024 (needs to match the encoder part)

     Layers: 8

     Number of heads for multi head attention: 8

     Dimension feed forward: 1024 x 4

     Max position embeddings: 128 (since the max length of the sentences in training data is around 60)
   - Tokenizer: I use the tokenizer provided from TA, which has max vocab size: 18022
   - Training strategy:
     (1) Since I use a collate function which pads each batch to match the max length of the sentence in the batch, for calculating the loss I use pack_padded_sequence to remove the paddings.
     (2) I set the batch size to be 32, the encoder learning rate=1e-5, the decoder learning rate=3e-5. And since the encoder is already pretrained, for the first 5 epochs, I freeze the weights in the encoder and train the decoder first.
     (3) I use cross entropy to calculate the loss and Adam as my optimizer.
   - Decoding strategy: I use beam search with beam size=6 to decode and predict captions.

- After training for 20 epochs:
  CIDEr: 0.9553907331423283
  CLIPScore: 0.7235356134166259

2. Report other 3 different attempts:

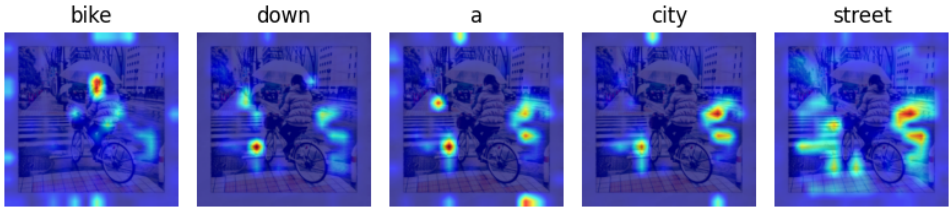
	CIDEr	CLIPScore
Change decoder layers to 6 layers	0.9086886632814632	0.7065505085876362
Change encoder to smaller model: "vit_base_patch16_224_in21k"	0.7860648756363299	0.6953444675469012
Use greedy algorithm (beam size=1) instead of beam search	0.8755360661587708	0.7306641716741196

```
final_output.json
PTBTokenizer tokenized 110968 tokens at 661692.55 tokens per second.
PTBTokenizer tokenized 19900 tokens at 212642.02 tokens per second.
CIDEr: 0.9553907331423283 | CLIPScore: 0.7235356134166259
6layers_output.json
PTBTokenizer tokenized 110968 tokens at 671604.18 tokens per second.
PTBTokenizer tokenized 19330 tokens at 224948.89 tokens per second.
CIDEr: 0.9086886632814632 | CLIPScore: 0.7065505085876362
p32_output.json
PTBTokenizer tokenized 110968 tokens at 692863.84 tokens per second.
PTBTokenizer tokenized 19923 tokens at 223048.87 tokens per second.
CIDEr: 0.7860648756363299 | CLIPScore: 0.6953444675469012
greedy_output.json
PTBTokenizer tokenized 110968 tokens at 683822.83 tokens per second.
PTBTokenizer tokenized 22457 tokens at 248028.29 tokens per second.
CIDEr: 0.8755360661587708 | CLIPScore: 0.7306641716741196
```

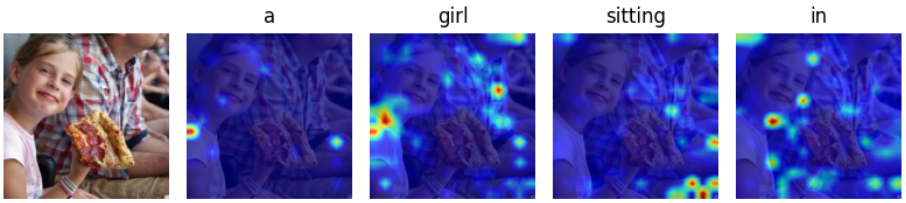
Part 3:

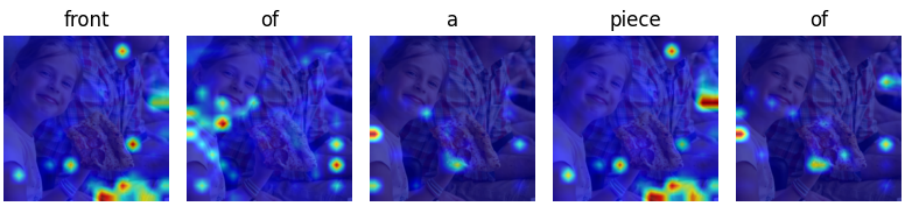
1. Visualize p3_data/images/:

(1) Bike:

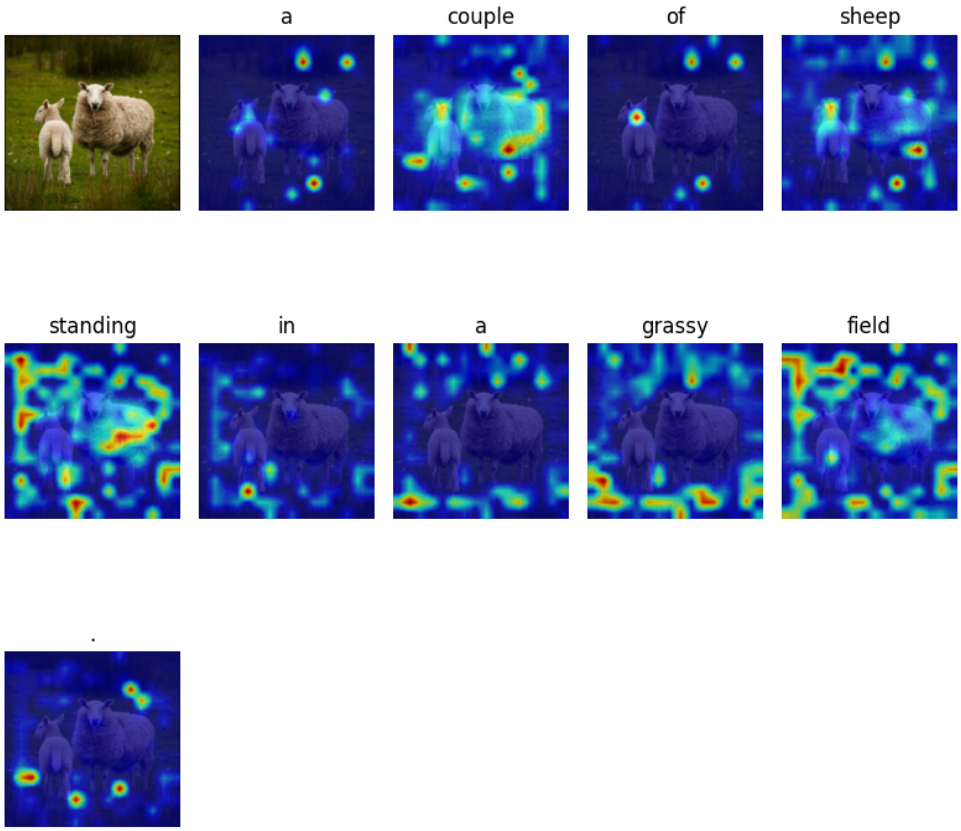


(2) Girl:

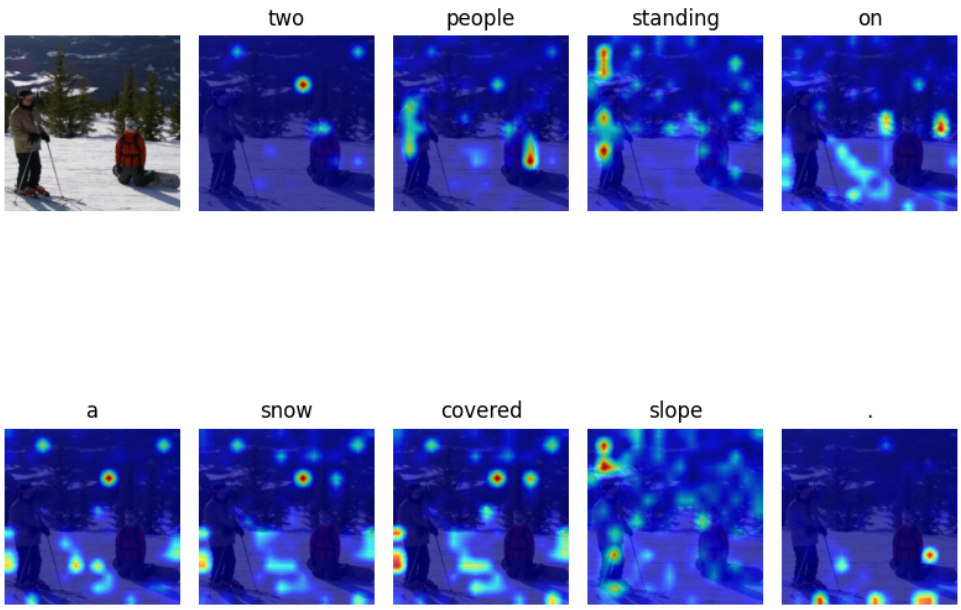




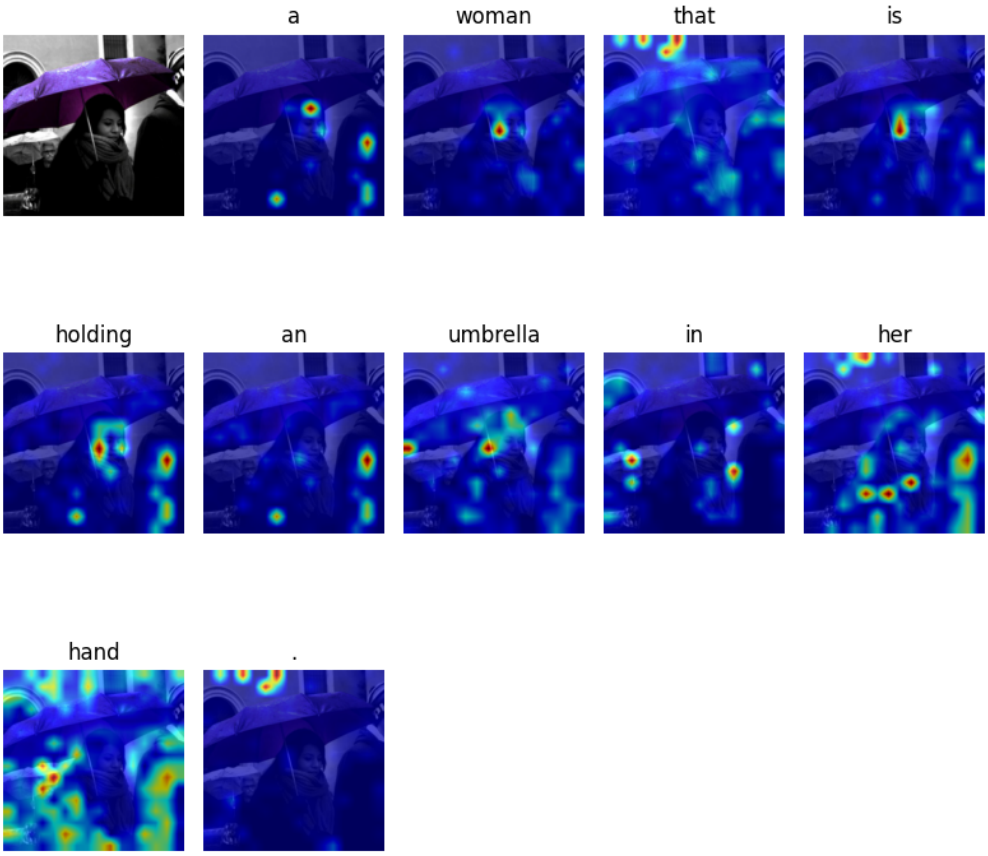
(3) Sheep:



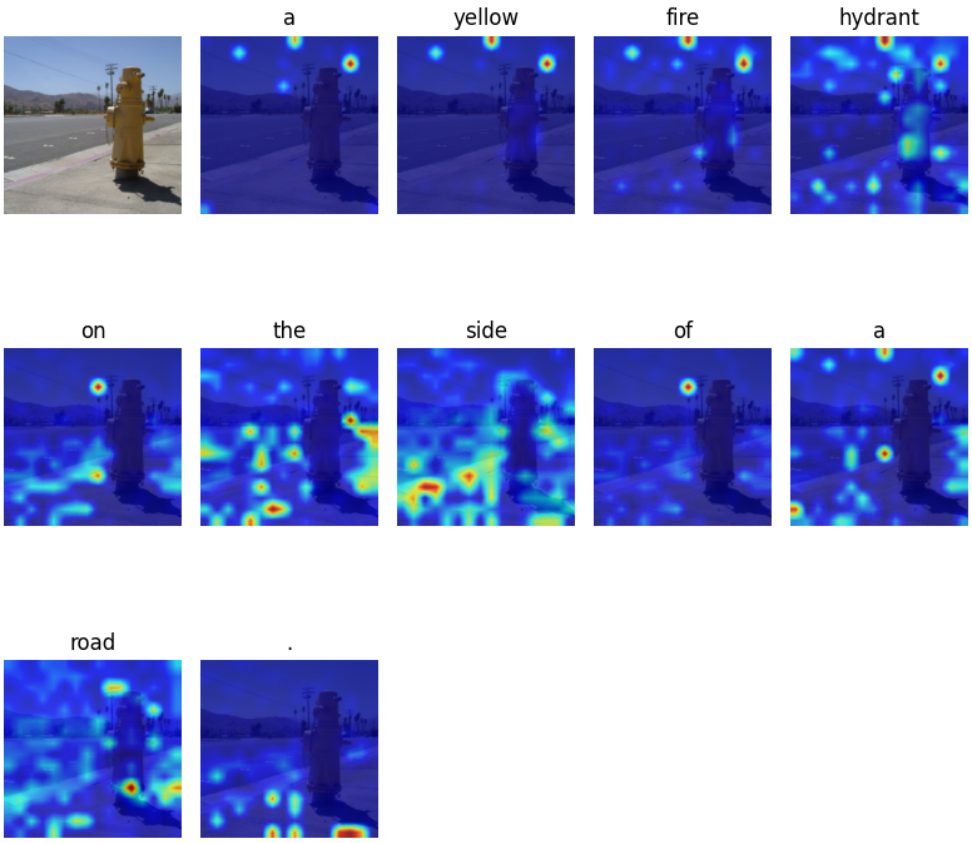
(4) Ski:



(5) Umbrella:

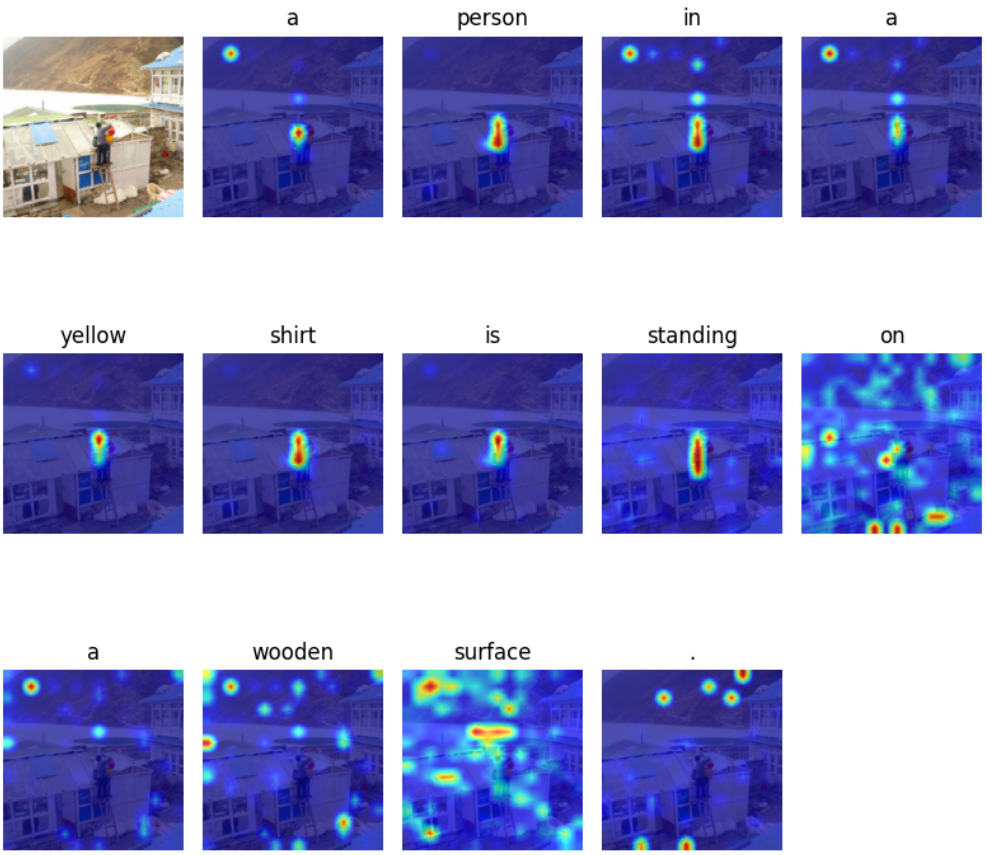


2. According to CLIPScore:
   (1) Top-1: 000000392315.jpg
   CLIPScore: 0.9844970703125



(2) Last-1: 6209779666.jpg

CLIPScore: 0.4656982421875



```
6209779666: 0.4656982421875
a person in a yellow shirt is standing on a wooden surface .
000000392315: 0.9844970703125
a yellow fire hydrant on the side of a road .
```

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?

(1) Bike:

- a person 確實有在人身上具有較大的 weight
- riding a bike 也有明顯在人以及 bike 上有較大的 weight，可以發現好像還沒說到 bike 的前幾個字（例如說在 riding 時）就有在 bike 有明顯較大的 weight 了。
- city street 這部分也可明顯看到原來人與腳踏車有較大的 weight，現在變成在馬路以及斑馬線上。

(2) girl:

- girl 字可以發現在女孩身上有明顯的 weight。
- 在後面 a piece of pizza 的部分也可以看到 pizza 亮起來。

(3) Sheep:

- 在 a couple of sheep 的 couple 就可以看到明顯的羊身上的權重非常高。
- 後面 grassy field 也可看到權重改成在羊以外的草地上面。

(4) Ski:

- Two people 可以看到在 people 那張圖上兩個人身上有亮起來。
- Snow covered slope 可以看到人以外的雪地都有亮起來。

(5) Umbrella:

- A woman 的可以看到女生的頭亮起來。
- That is holding an umbrella 可以課到在 that 時雨傘就已經亮起來了。

(6) Top-1: 000000392315.jpg

- Yellow fire hydrant: 可以看到確實在黃色消防栓上有較大的權重，除此之外不知道為什麼天空中有些部分還是有亮點。
- On the side of the road: 可以到消防栓旁邊都有亮起來，道路以及人行道也都有明顯的亮起。

(7) Last-1: 6209779666.jpg

- A person in a yellow shirt is standing: 圖中可以看到前半部分人的形狀是非常明顯的有亮起，但是從原圖中可以看到這個人其實並不是穿著黃色衣服。
- On a wooden surface 可以看到周圍的環境也都有亮起來，但是事實上好像與圖片中不太一樣。

Discussion:

1. 從 attention map 可以大概地看出大約是由途中哪個部分來推出這個字的，不太確定是否有可能不過我發現好像在這個字要出現的前幾個字時就會有亮起來的現象。
2. 我認為 part2 會有較明顯的 CLIPScore 高低之分應該是圖片形容難易度的問題，在 CLIPScore 較高的圖片中感覺特徵物非常大也非常明顯，可以簡單地形容出來。而在 CLIPScore 較低的圖片中人物特別的小，或許是因為特徵物較小造成判斷失誤的原因。