Deep Learning for Computer Vision HW3 Report

B10901091 李冠儀

P1.

1. Paper reading(3%)
   Please read the paper “Visual Instruction Tuning” and briefly describe the important components (modules or techniques) of LLaVA.

   There are 4 key component / novelty in LLaVA
   1. **Visual Encoder**: Utilizes a pre-trained CLIP model to process images, generating visual embeddings that capture semantic information.
   2. **Projection Layer**: Maps the visual embeddings from the CLIP encoder into the LLM's input space, ensuring compatibility between visual and textual data representations.
   3. **Text (Large Language Model)**: Employs a pre-trained LLM, such as Vicuna, to handle and generate text based on the combined visual and textual inputs.
   4. **Instruction Tuning**: Fine-tunes the integrated model using a dataset of image-instruction-response triplets, enabling it to follow visual instructions effectively.
   These components work together to allow LLaVA to interpret and respond to visual inputs within a conversational context, enhancing the model's ability to understand and generate language grounded in visual information.

2. Prompt-text analysis (6%)
   Please come up with two settings (different instructions or generation config). Compare and discuss their performances.

	Setting A	Setting B
Instructions	Give me a desciption of the image in one short sentence.	
num_beams	1	2
CLIDEr	1.121	1.165
CLIPscore	0.782	0.785

Increasing num_beams allows the model to explore more possible sequences at each decoding step, potentially leading to better, more refined outputs. With higher beam numbers, the model

evaluates multiple candidate sequences, selecting the one with the highest overall score. Hence, setting with num_bean = 2 is a little better than num_bean = 1.

P 2.

1. Report your best setting and its corresponding CIDEr & CLIPScore on the validation data. Briefly introduce your method. (TA will reproduce this result) (5%)
   - Method:
     The model architecture consists of an encoder, decoder, and tokenizer designed for efficient image-to-text generation. The encoder employs a pretrained Vision Transformer (ViT) model, specifically **vit_large_patch14_clip_224.laion2b**, to extract high-dimensional feature representations from input images. The decoder is a Transformer-based mo del fine-tuned with Low-Rank Adaptation (LoRA) to decode these image features into text sequences. It incorporates multi-head attention layers for contextual understanding, feedforward networks for feature transformation, and lightweight linear layers optimized with LoRA to reduce computational overhead while maintaining performance. A byte-pair encoding (BPE) tokenizer is used to convert captions into tokenized sequences.

     During training and inference, the decoder's weights are initialized from a checkpoint file to leverage previously trained representations. For caption generation at inference, the model first extracts image features using the encoder, then decodes these features into token sequences using the decoder. Captions are generated iteratively by predicting the next token until either an end-of-sequence token is reached or the maximum sequence length is achieved.
   - Result: CIDEr:0.9552 / CLIPScore:0.7318

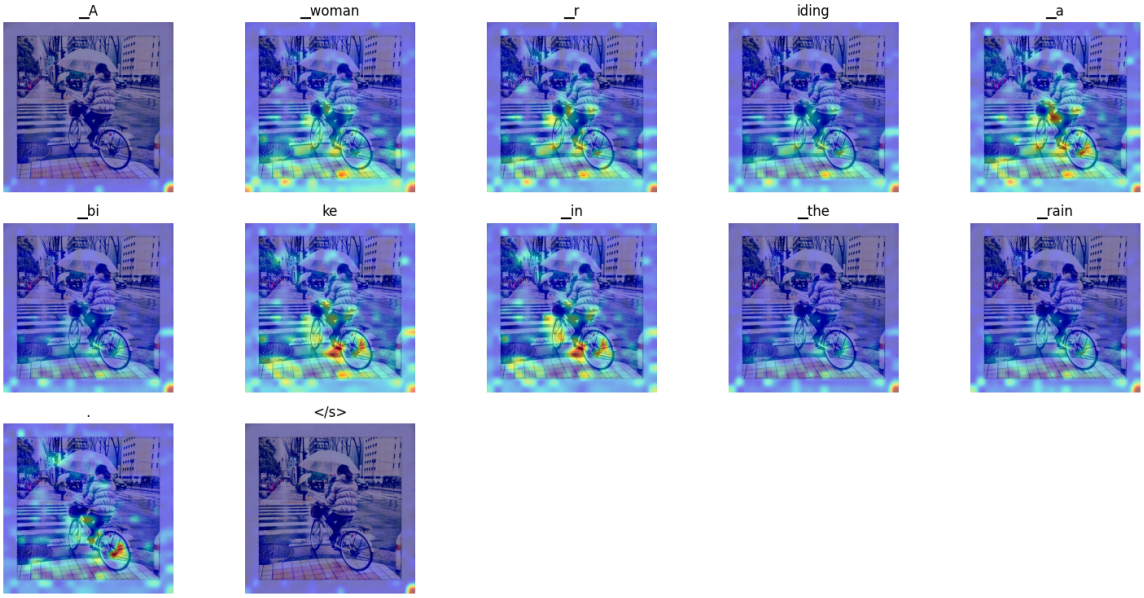
2. Report 2 different attempts of LoRA setting (e.g. initialization, alpha, rank…) and their corresponding CIDEr & CLIPScore. (5%, each setting for 2.5%)
   Rank 16 is better than Rank 8.

	Setting A	Setting B
Rank	8	16
CLIDEr	0.941	0.956
CLIPscore	0.712	0.731

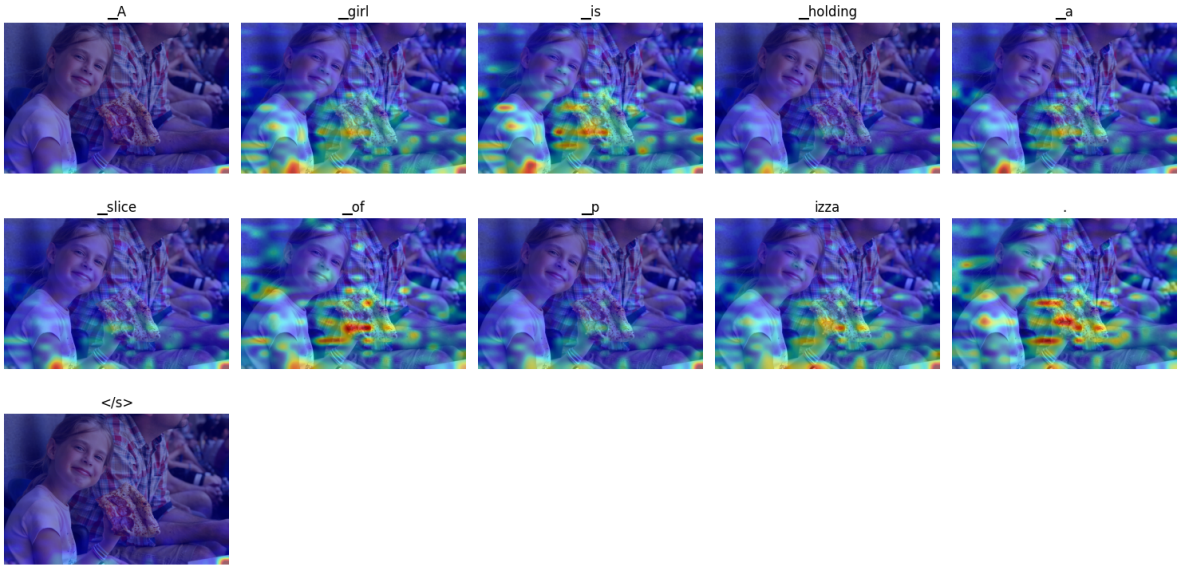
P 3.

1. P1

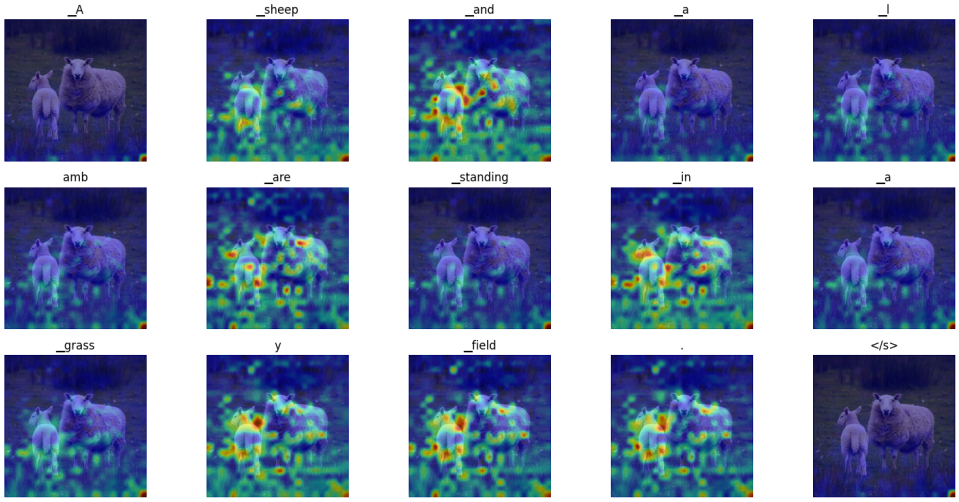
   Bike:



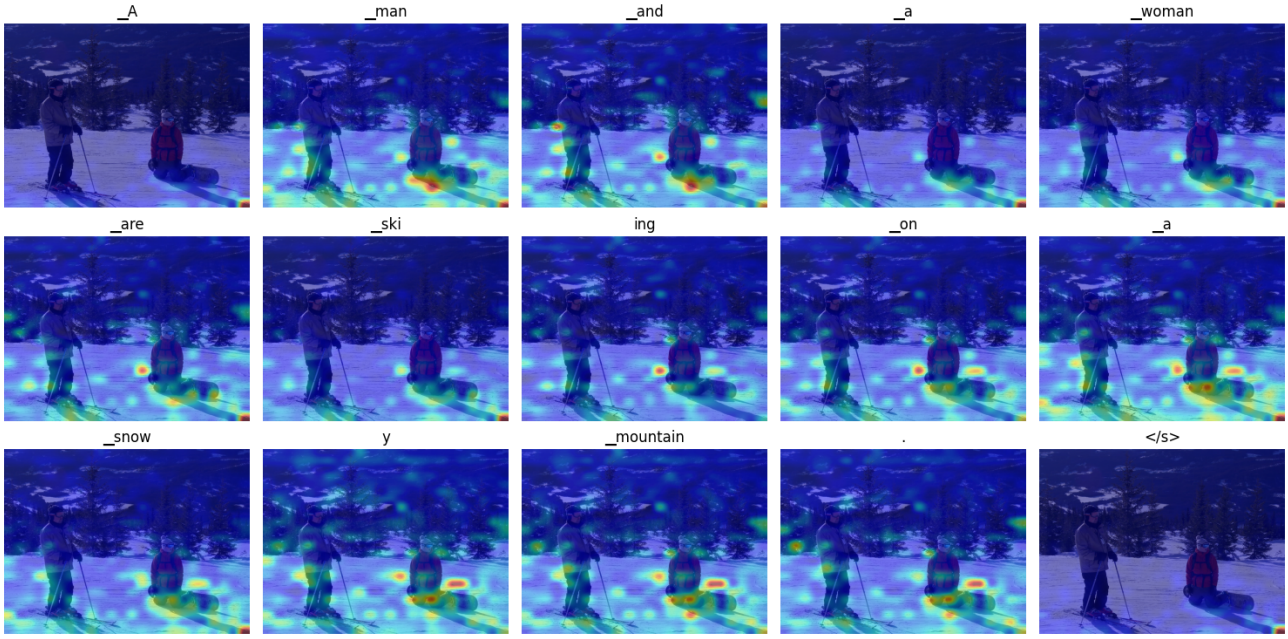
   Girl:



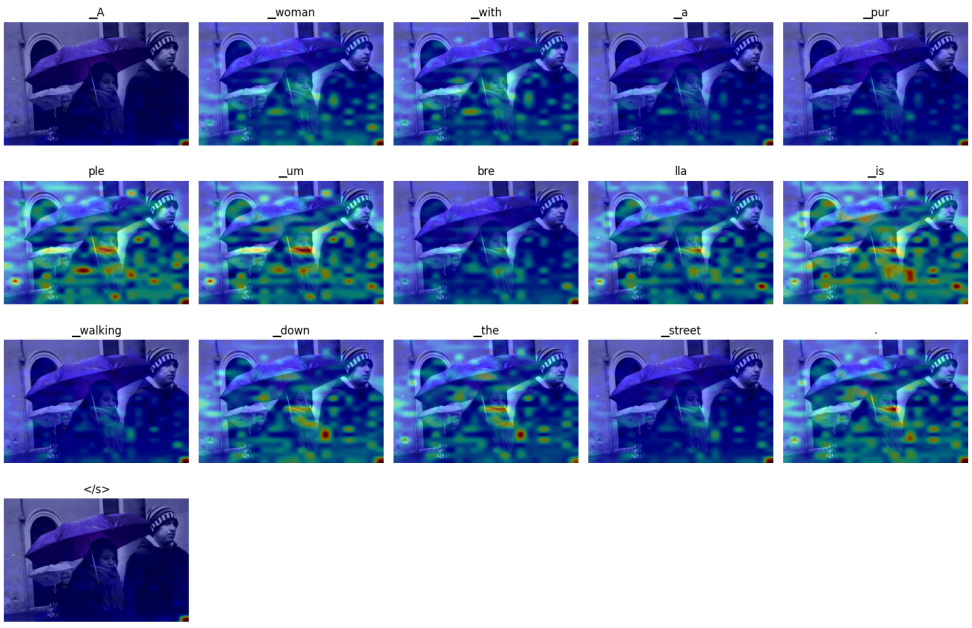
Sheep:



ski:



Umbrella:

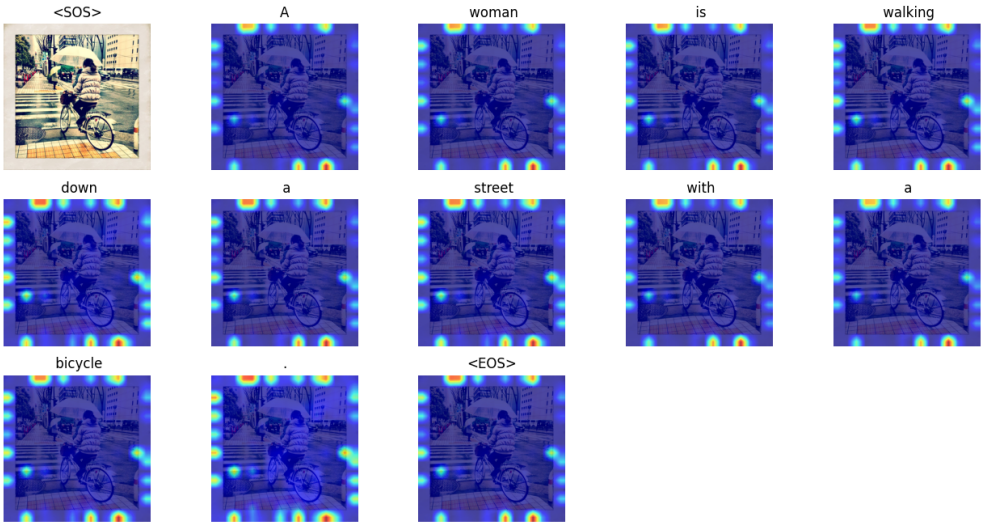


Conclusion:

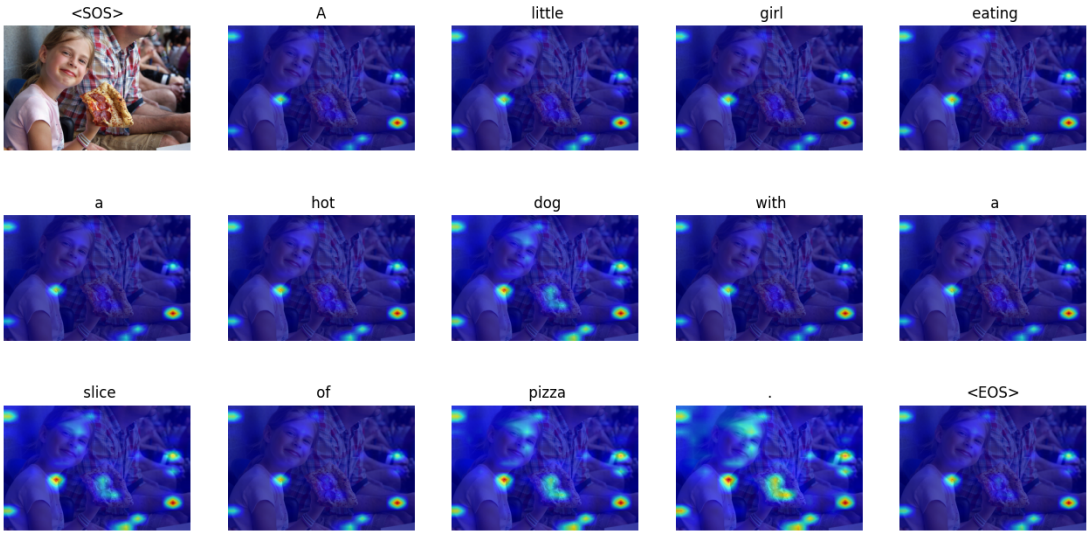
When outputting attention, I found that using the final layer's output does not always yield the best results. Each attention head may focus on different aspects—for instance, some focus on scenes, others on portraits, and some on colors. Therefore, to achieve optimal results, it is necessary to consider and process multiple attention layers comprehensively. Currently, my approach is to use the eleventh layer's output, which provides better results compared to the final layer. However, there are still some limitations, such as the inability to accurately capture certain colors and specific objects.

2. P2

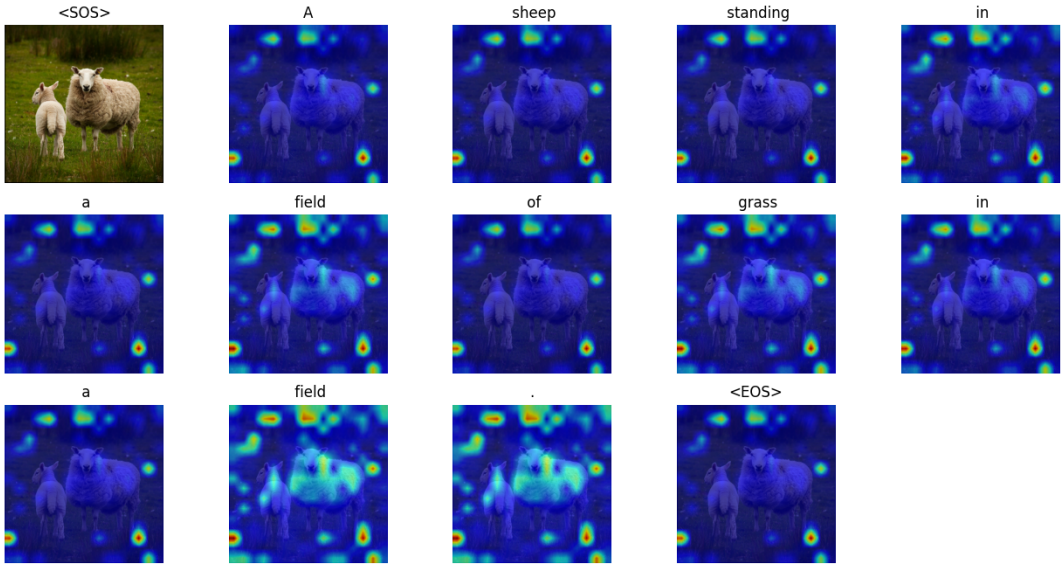
Bike:



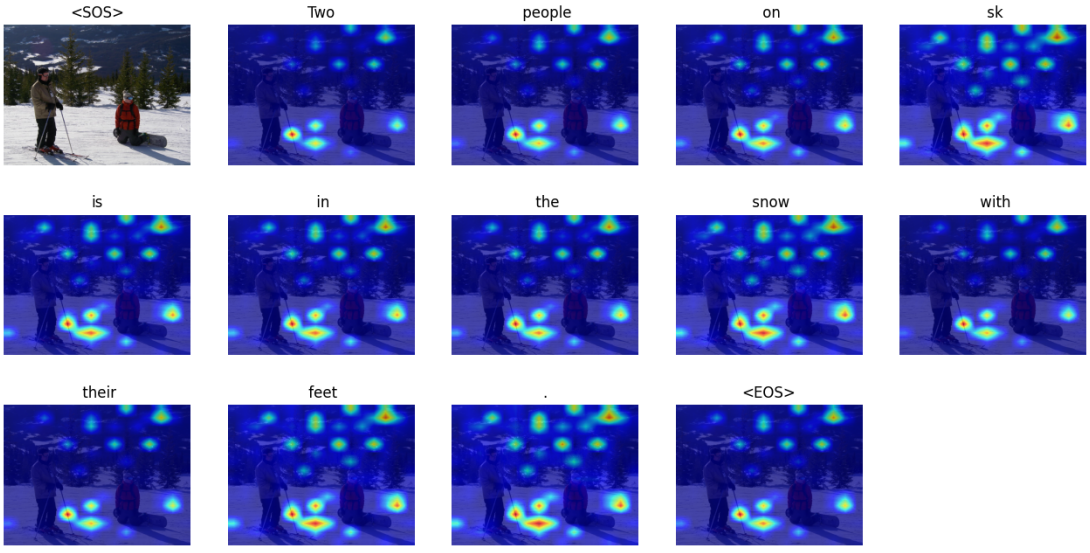
Girl:



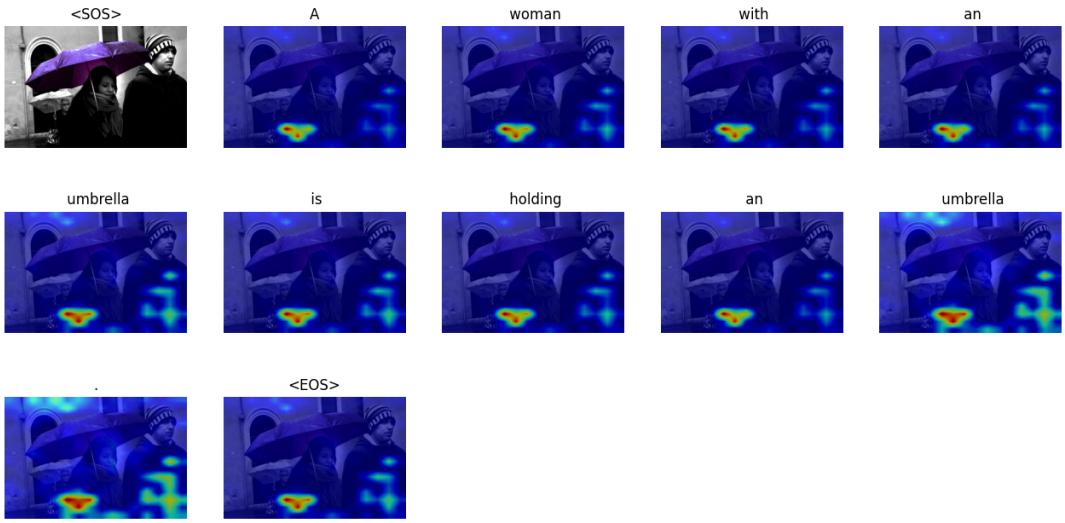
Sheep:



ski:



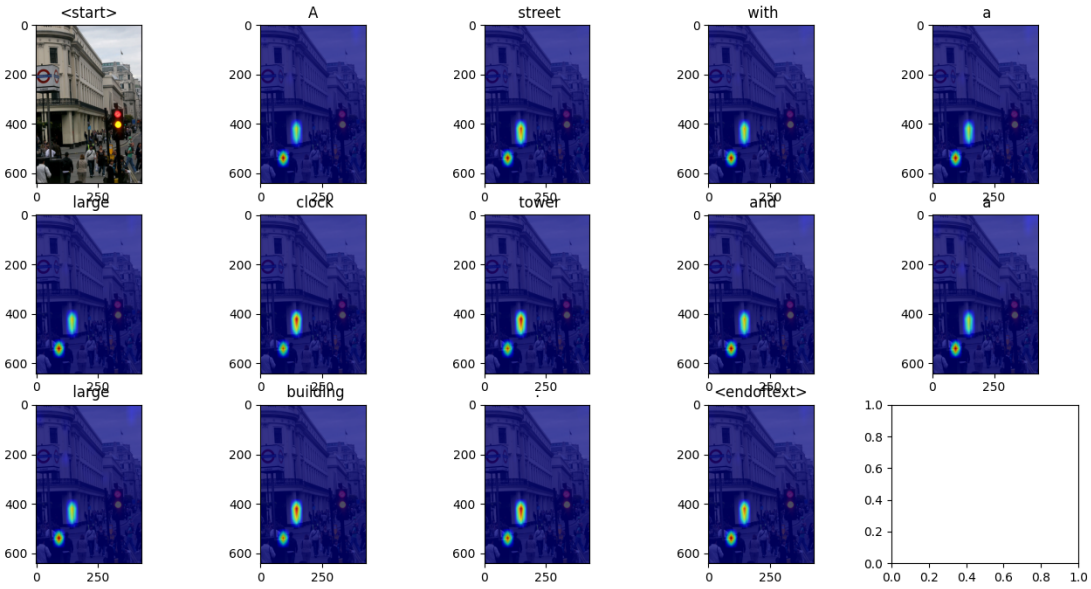
Umbrella:



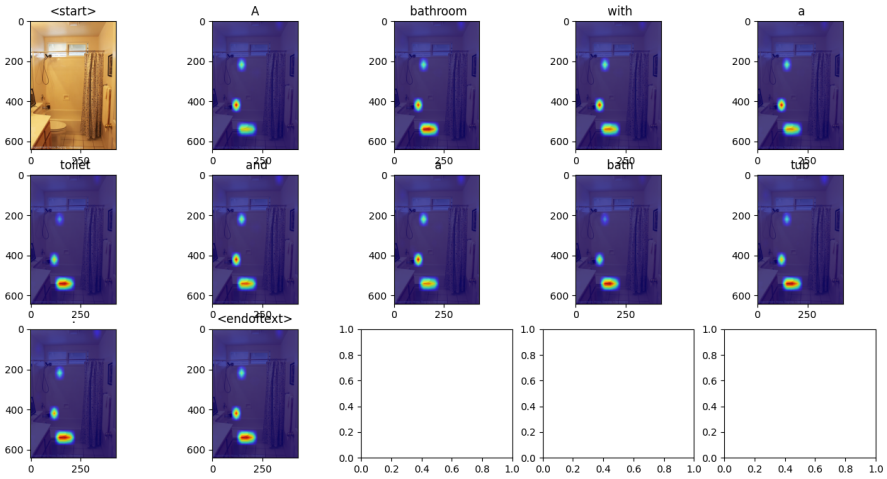
Conclusion: Just like in the first case, I encountered the same challenge: no matter which specific attention layer I output, it fails to accurately reflect the complete capture of all objects. This leads me to believe that different layers and different heads focus on different aspects. However, overall, there is a noticeable trend of increased attention within the range where specific keywords appear. Therefore, my results do capture part of the relevant information.

3.

a. Best:155.png, CLIPScore = 0.91



b. Worst:1654.png, CLIPScore = 0.34



c. I believe the results are reasonable. In the best combinations, when specific objects such as "building" or "tower" are mentioned, the attention is more pronounced within the object's range. However, for connecting words, there is no distinct visualization. Conversely, in the worst combinations, this kind of pattern is absent.