CSIE5428 Computer Vision Practice with Deep Learning

Homework 3 Report

Name: 高榮浩 **ID:** R12922127

1. Image Captioning - BLIP-2

BLIP-2 is a cost-effective vision-and-language pre-training strategy that leverages off-the-shelf frozen models. It utilizes a lightweight Querying Transformer in two stages to effectively bridge the modality gap. Despite its significantly reduced parameter count, BLIP-2 outperforms existing methods, surpassing Flamingo80B by 8.7% in zero-shot VQAv2 while employing 54 times fewer parameters. Notably, the model exhibits emerging capabilities in zero-shot image-to-text generation based on natural language instructions.

A. Comparative Analysis of Caption Generation Performance

Begin by strictly adhering to the homework instructions, ensuring the exclusion of images with more than one category or those containing more than six bounding boxes. This is a crucial step to guarantee the avoidance of low-quality image generation in text-to-image processes.

Subsequently, embark on experiments involving four distinct pre-trained models: [Salesforce/blip2-flan-t5-xl], [Salesforce/blip2-opt-2.7b], [Salesforce/blip2-opt-6.7b], and [Salesforce/blip2-opt-6.7b-coco]. To facilitate a straightforward comparison of the performance across these models, the following tables present two sets of results. Each set highlights the image captioning outcomes produced by the four selected pre-trained models in response to specific input images.

- **Example - Jellyfish**

IMG_2489_jpeg_jpg.rf.ffb357957a29cdef43f3fdfb2a13c417.jpg

[Salesforce/blip2-flan-t5-xl]	jellyfish in the ocean
[Salesforce/blip2-opt-2.7b]	jellyfishs in the aquarium
[Salesforce/blip2-opt-6.7b]	jellyfish in the aquarium
[Salesforce/blip2-opt-6.7b-coco]	a group of jellyfish swimming in the ocean under blue water

- **Example - Fish**

IMG_2402_jpeg_jpg.rf.ff2e5af0a2d1693c155a01d7494fc8e4.jpg

[Salesforce/blip2-flan-t5-xl]	a fish tank with a lot of fish in it
[Salesforce/blip2-opt-2.7b]	a group of fish in a tank with coral
[Salesforce/blip2-opt-6.7b]	a fish tank with a black and white fish
[Salesforce/blip2-opt-6.7b-coco]	a group of fish swimming in an aquarium with rocks and coral

Based on the aforementioned results, it is evident that the text generated by [Salesforce/blip2-opt-6.7b-coco] is notably more accurate and detailed compared to the outputs of the other three selected pre-trained models. Consequently, I opt to utilize the text generated by [Salesforce/blip2-opt-6.7b-coco] for subsequent text-to-image generation.

It is important to note, in accordance with the TA's reminder, that the BLIP-2 model lacks knowledge of the term "puffin." Therefore, I manually refine the generated text to mitigate any potential errors in the text-to-image generation process.

B. Template Design for Text-to-Image Generation Comparison

The following tables present two templates of prompts that I have designed.

Template #1 `prompt_w_label`	`f"{generated_text}, {category}, height: {height}, width: {width}"`
Template #2 `prompt_w_suffix`	`f"{generated_text}, {category}, height: {height}, width: {width}, HD quality, highly detailed"`

Please be aware that the term "`generated_text`" refers to the output generated by [Salesforce/blip2-opt-6.7b-coco].

2. Text-to-Image Generation - GLIGEN

GLIGEN represents a groundbreaking advancement in text-to-image generation models, enhancing controllability through the integration of grounding inputs alongside textual prompts. This model extends pre-trained models by introducing grounding information into novel trainable layers. Notably, GLIGEN excels in open-world grounded text-to-image generation, showcasing robust zero-shot performance on COCO and LVIS datasets when compared to existing baselines.

A. Text Grounding Generation

In text-to-image generation using text grounding, I employ the [masterful/gligen-1-4-generation-text-box] model. For each image generated, the model's inputs include a designed prompt, categories expressed as phrases, bounding boxes, image height, and image width.

B. Image Grounding Generation

In text-to-image generation using image grounding, I employ the [anhnct/Gligen_Text_Image] model. For each image generated, the model's inputs include a designed prompt, categories expressed as phrases, bounding boxes, image height, image width, and reference image.

3. Performance Evaluation Based on FID for Text-to-Image Generation

The Fréchet Inception Distance (FID) is a metric employed to evaluate the quality of images generated by a generative model. In contrast to the earlier Inception Score (IS), which assesses solely the distribution of generated images, the FID goes further by comparing the distribution of generated images with that of a set of real images (ground truth).

The table below presents the performance of text or image grounding in text-to-image generation for two templates of prompts that I designed.

	Text Grounding		*Image Grounding*	
Prompt	Template #1	Template #2	Template #1	Template #2
FID	**167.78**	169.39	169.11	**161.69**

In each evaluation, I meticulously choose 20 images per category from the training dataset and subsequently generate an additional 20 images per category. This process results in a total of 140 images, comprising 140 real images and 140 synthesized images. However, it's crucial to note that only 6 images from the training dataset fall under the category "jellyfish" while meeting the specified criteria of containing only one category and having less than or equal to six bounding boxes, as mentioned earlier. Consequently, I compute the FID between the original and generated datasets, each containing 126 images. Additionally, all images are resized to 512x512 before evaluation.

In the text grounding evaluation results, Template #1 exhibits a lower FID score than Template #2, prompting us to select Template #1 for subsequent text grounding data augmentation. Despite Template #1 having a lower FID score than Template #2, their scores are roughly comparable. Upon scrutinizing the design of these two templates, it becomes apparent that Template #1 is a substring of Template #2, suggesting that Template #2 contains more information than Template #1. As a result, for the forthcoming image grounding evaluation, we will persist in utilizing both templates for analysis.

In the image grounding evaluation results, Template #2 displays a lower FID score than Template #1, leading us to choose Template #2 for subsequent image grounding data augmentation. Moreover, the difference between these two templates is more noticeable than in the text grounding evaluation before, validating the earlier analysis.

In summary, the FID score for template #2 in image grounding text-to-image generation is the lowest overall.

4. Detection Model Improvement Analysis

A. Object Detection – DINO

I choose to use the DINO model, as I did in homework 1. The pre-trained checkpoint I leverage from the model zoo is specifically DINO-4scale, with the Swin-L backbone. Consequently, this implementation involves the utilization of two pre-trained weights, namely those for DINO-4scale and Swin-L, as indicated in the following table.

Name	*File*	*Dataset*	*Source*
DINO-4scale (36 epoch setting)	`checkpoint0029_4scale_swin.pth`	COCO 2017	Link
Swin-L	`swin_large_patch4_window12_384_22k.pth`	ImageNet-22K	Link

Please note that I have deliberately avoided using the DINO-5scale (36 epoch setting) pre-trained checkpoint. While it does offer a higher box AP, my GPU (GeForce RTX™ 2080 Ti 11G) lacks the capacity to accommodate it. This is due to a `RuntimeError` that occurs, specifically "`CUDA out of memory`," even with the batch size initially set to 1.

In that case, I attempted to use DINO-4scale (36 epoch setting) with an initial batch size of 2, but my GPU still couldn't handle it. As a result, I had to reduce the batch size to 1, and that ultimately led to successful execution.

B. Data Augmentation

For both text and image data augmentation, I begin by counting the occurrence of each category in the original training dataset. The following table presents the results of this counting process.

Category	*fish*	*jellyfish*	*penguin*	*puffin*	*shark*	*starfish*	*stingray*
Occurrence	1961	385	330	175	259	78	136

From the provided table, it's evident that the original training dataset exhibits a significant imbalance across categories. To quantify this imbalance, we calculate the standard deviation of the occurrence numbers for each category. The calculated standard deviation is 614.88.

Therefore, data augmentation is deemed necessary. Given that the category "fish" has the highest occurrence, the dataset related to the category "fish" will not undergo augmentation. Moreover, the occurrences of other categories in the augmented (original + generated) training dataset should be either lower or equal to the number of occurrences in the category "fish."

Next, I collect images from the original training dataset that meet the specified criteria of containing only one category and having less than or equal to six bounding boxes, as mentioned earlier, while also adhering to the requirement stated in the last paragraph. Subsequently, I generate images corresponding to these selected images for both text and image data augmentation. The following table presents the occurrence of each category in the augmented training dataset.

Category	*fish*	*jellyfish*	*penguin*	*puffin*	*shark*	*starfish*	*stingray*
Occurrence	1961	412	375	252	281	132	167
Added	0	27	45	77	22	54	31

From the provided table, it is apparent that the augmented training dataset still displays a notable imbalance across categories, albeit less severe than the original training dataset. To quantify this imbalance, the standard deviation of the occurrence numbers for each category is calculated. The resulting standard deviation is 599.14, which is lower than it was in the original training dataset before.

C. Performance of Data Augmentation

The provided table showcases the performance results for the validation set with and without data augmentation. It's crucial to note that the hyperparameters used to generate these results remain consistent, with a learning rate of 0.00005 and the number of epochs set to 48, ensuring fairness in the comparison.

	Before	*After*	
		Text Grounding	*Image Grounding*
Best Epoch	36	19	30
$AP_{[50:5:95]}$	**0.588**	0.586	0.583
AP_{50}	**0.866**	0.861	0.853
AP_{75}	**0.621**	0.601	0.611

It's important to note that the index for the "Best Epoch" in this table begins at 0.

D. Deterioration Analysis

The above table demonstrates that both text and image grounding data augmentation lead to a degradation in performance. The reasons for this deterioration are as follows.

Firstly, the original validation dataset also displays a notable imbalance across categories. The following table presents the occurrence of each category in the original validation dataset.

Category	*fish*	*jellyfish*	*penguin*	*puffin*	*shark*	*starfish*	*stingray*
Occurrence	459	155	104	74	57	27	33

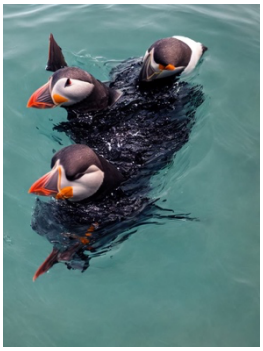
Thus, when using this imbalanced dataset to evaluate the model, the imbalance may lead to inaccurate evaluations.

Secondly, limited to the BLIP-2 model, if there are any errors due to BLIP-2, these errors may propagate to the data augmentation process. For instance, as illustrated in the following table, BLIP-2 does not successfully detect "penguin" in the image. Consequently, the generated text erroneously describes the penguin as a "blue fish."

IMG_2309_jpeg_jpg.rf.088f73ff0b07c30ce6212eb4e3013708.jpg

[Salesforce/blip2-opt-6.7b-coco]	a blue fish swimming in a clear water pool near rocks

Thirdly, limited to the GLIGEN model, if there are any errors due to GLIGEN, these errors may propagate to the data augmentation process. For example, in the image below, which should only contain one puffin, there are three heads but one body, indicating an error in the generation process.

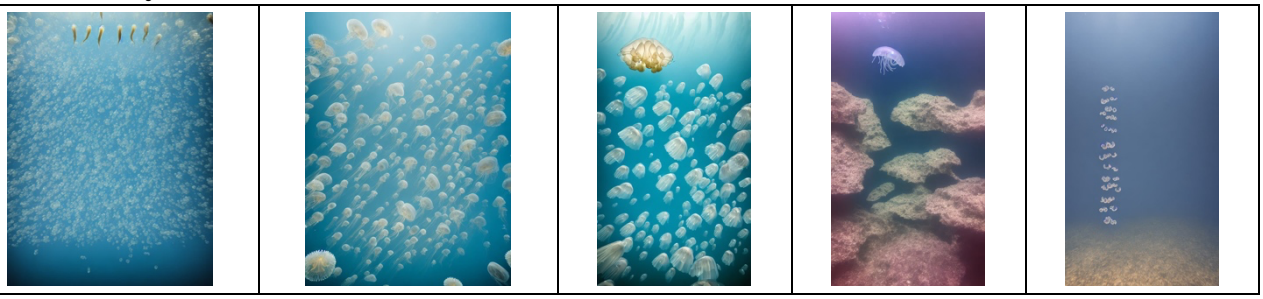


Due to the aforementioned three reasons I speculate, the performance after data augmentation is worse than before.

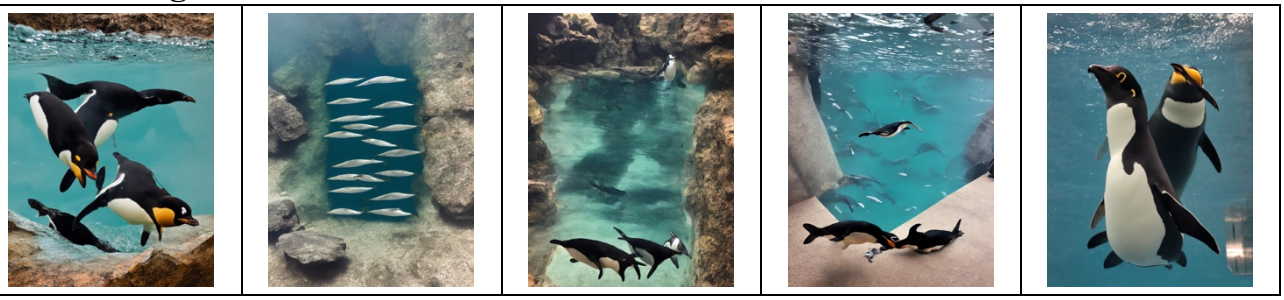
5. Visualization

A. Fish

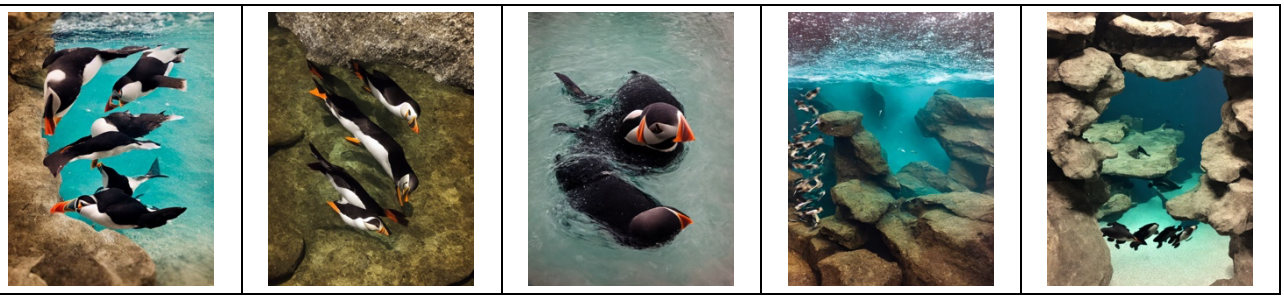
B. Jellyfish



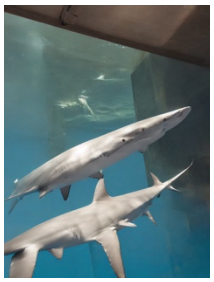
C. Penguin

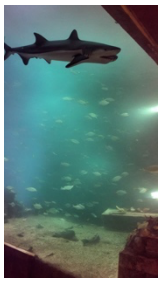
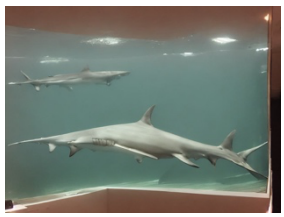


D. Puffin

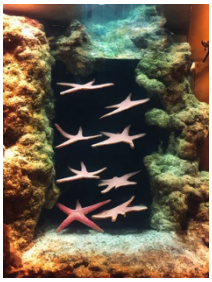


E. Shark

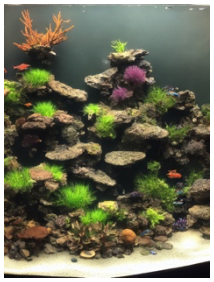


F. Starfish







G. Stingray