Computer Vision Practice with Deep Learning
Homework 3

秦孝媛 R12725026
資訊管理學系研究所一年級

Image Captioning

(a) Compare the performance of 2 selected different pre-trained models in generating captions, and use the one you find the most effective for later problems.

I used Figure 1 as image captioning example.

Figure 1: Image Captioning Example

- `Salesforce/blip2-opt-6.7b`

```
a group of puffin are sitting on rocks near a waterfall
```

- `Salesforce/blip2-flan-t5-xl`

```
a puffin is sitting on a rock in an aquarium
```

(b) Design 2 templates of prompts for later generating comparison.

- Template 1:

```
generate text + label + image height + image weight
```

- Template 2:

```
Template 1 + in an aquarium setting + (set(labels)) + in the
    bounding box, highly detailed and harmonious tones, no
    watermarks or text, no fluorescent colors, complete, moderate
    low saturation , real aquarium photos
```

I believe that the pre-trained model `Salesforce/blip2-opt-6.7b` demonstrates superior performance in generation. Therefore, I will utilize this particular model for text prompt generation in subsequent data augmentation processes.

Text-to-Image Generation

(a) Use 2 kinds of generated prompts from Problem 1(b) to generate images

(b) Select the prompts for better-generating results, and perform image grounding generation.

After utilizing two kinds of generated prompts from Problem 1(b), I observed that Template #2 performed slightly better. The images generated were more closely resembling real aquarium photos. Although the differences between Template #1 and Template #2 were minimal, I ultimately chose Template #2 as the text prompt for image grounding.

(c) Table of performance based on FID:

Table 1: Performance based on FID.

	Text grounding		**Image grounding**
Prompt	Template #1	Template #2	Template #1
FID	136.30	136.76	135.44

(d) Table of the improvement of detection model from HW1 after data augmentation:

Table 2: Improvement of the detection model from HW1 after data augmentation.

	Before Data Augmentation	**After Data Augmentation**	
		Text grounding	**Image grounding**
AP [50:95]	52.42	48.56	50.78

(e) Visualization

Below shows the best 5 images for each category

- fish

Figure 2: Visualization of fish category

- jellyfish

Figure 3: Visualization of jellyfish category

- penguin

Figure 4: Visualization of penguin category

- puffin

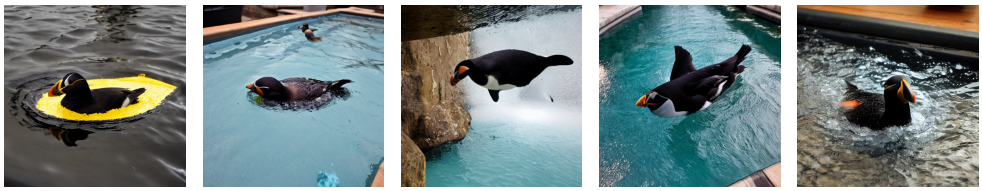


Figure 5: Visualization of puffin category

- shark

Figure 6: Visualization of shark category

- starfish

Figure 7: Visualization of starfish category

- stingray

Figure 8: Visualization of stingray category

Data Augmentation Experiments

(a) Detailed settings of experiments

- Checkpoint

  checkpoint0033_4scale.pth, DINO Official 36 epoch setting, backbone R50 pre-train weight

- Data Augmentation Setting

  My data augmentation strategy aimed to address the imbalance across categories in our training dataset. I identified categories with fewer than 130 unique images and excluded the 'creatures' category due to its initial surplus.

  The augmentation targeted single-category images with up to six annotations, with each eligible image receiving a maximum of four augmentations. This procedure increased the counts for 'jellyfish', 'penguin', 'puffin', 'shark', 'starfish', and 'stingray' to nearer the 130-image goal, as illustrated in the Figure 9 bar plots.

  Post-augmentation, 'jellyfish', 'penguin', and 'puffin' categories saw a rise to over 80 images each, 'shark' to 134, and 'starfish' and 'stingray' to 93 and 91 respectively. This enhanced the dataset's uniformity, setting a stronger foundation for training a balanced image classification model.

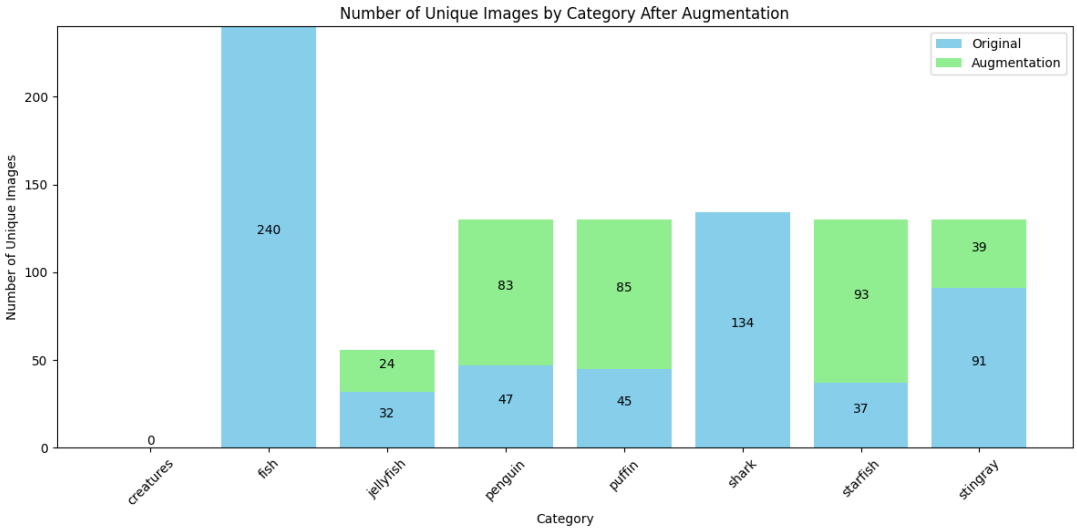


Figure 9: Data Augmentation Setting in Training Set

(b) Reasons and discussion I believe there are two main reasons why Augmentation might decrease after Text-to-Image Generation Augmentation:

(1) Validation Dataset Problems

The validation set also suffers from data imbalance issues. Evaluating with imbalanced data cannot accurately measure the effects of data augmentation. Moreover, since the images generated by GLIGEN differ stylistically from the original images, I suggest implementing Image-to-Text Data Augmentation in the Validation Set as well. This would better adapt the Classifier to the styles of these generated images. However, due to operational guidelines and constraints, this implementation was not carried out in this homework.

(2) GLIGEN Incorrect Image Generation

Although GLIGEN is relatively superior among Text-to-Image models, it still sometimes produces inaccurate or even incorrect images. This might be due to limitations in its training data. The model's ability to generate accurate images is heavily dependent on the diversity and quality of the images it was trained on. If the dataset lacks variety or contains inaccuracies itself, GLIGEN may replicate these flaws in its outputs. Additionally, the complexity of interpreting text descriptions and translating them into visual elements is a challenging task. Subtle nuances in language or abstract concepts can be difficult for the model to accurately represent visually, leading to inconsistencies or errors in the generated images. Finally, the inherent limitations of the algorithm's design might also contribute to these inaccuracies, as it may not be fully capable of handling all the complexities involved in the text-to-image conversion process.

(c) Supporting evidence This section substantiates the two reasons mentioned in the "Reasons and Discussion" section ,

(1) Validation Dataset

As depicted in Figure 10, there is a significant data imbalance within the Validation Dataset. The bar plot illustrates a disproportionate number of images across different categories, with

the 'fish' category having an overwhelming majority of 63 images, while other categories like 'jellyfish' and 'puffin' are underrepresented with only 9 and 15 images respectively. This disparity can skew the performance metrics and does not provide an accurate reflection of the classifier's effectiveness across various classes.

(2) GLIGEN wrong image generation

To evaluate the quality of images generated by GLIGEN, I superimposed the bounding boxes from the training annotations onto the generated images. I discovered that non-square images posed a challenge for GLIGEN, leading to marine life being generated in incorrect regions. This issue, shown in Figure 11a, could be mitigated by resizing all images to a 512*512 square and rescaling the bounding boxes accordingly. Additionally, there are instances where the marine life specified by the bounding boxes still extends beyond these parameters, as evidenced in Figure 11b. Despite filtering out training images with more than one category or more than six bounding boxes, the problem of excessive noise in the background of generated images persists. At times, this noise includes additional marine life, creating a confusing backdrop as indicated in Figure 11c. Lastly, even when the marine organisms generated by GLIGEN are within the bounding box and the background noise is minimal, the organisms themselves can appear somewhat bizarre and unrealistic, as seen in Figure 11d.

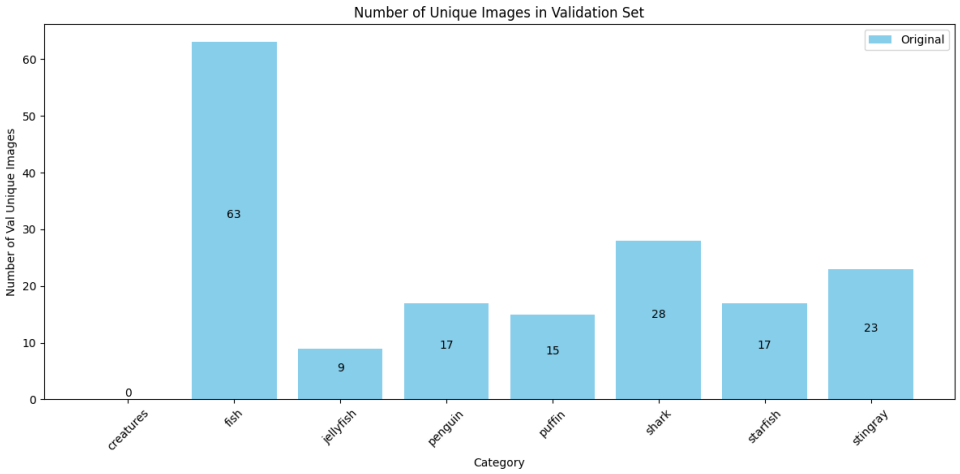
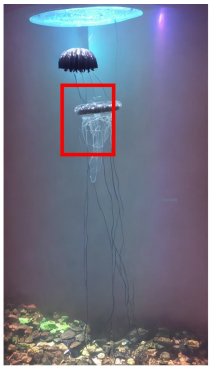
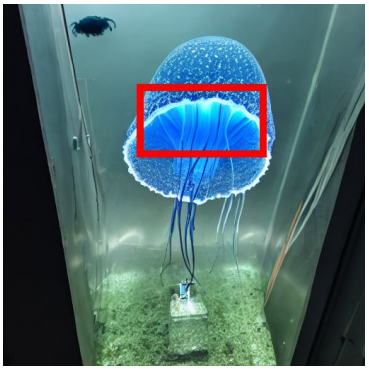


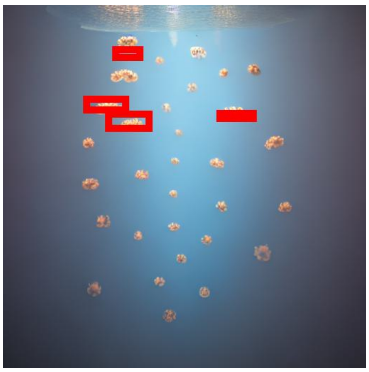
Figure 10: Data Imbalance in Validation Set



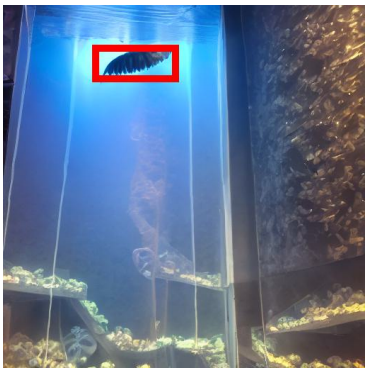
(a) Non-square image misalignment



(b) Generated range exceeds the bounding boxes



(c) A lot of noise in the background



(d) Generate unrealistic animals

Figure 11: GLIGEN wrong image generation evidences