IMVFX FINAL REPORT

Team21

杜家漢

● Model

CLIP

A model that can grad similarity between text and image.

We use it to calculate loss of images generated by VQGAN.

VQGAN

A method of generating images using discrete v-vectors. We find it more suitable for integration with CLIP.

● Expectation

While running the project, we expect to get picture with input text like pictures below, so we use VQGAN that can generate picture with or without initial image, and then CLIP will calculate the loss of this picture with input text to improve image.

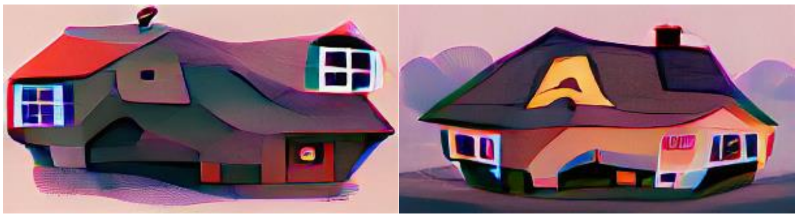
Text prompt: "It's like that drug trip I saw in that movie while I was on a drug trip. Trending on Artstation". VQGAN+CLIP art on NightCafe Creator.

Text prompt: "A colourful cubist painting of a parrot in a cage". VQGAN+CLIP made with NightCafe Creator.

Difficulty

1. Initial image

   Absence of an initial image may result in a lower-quality outcome.

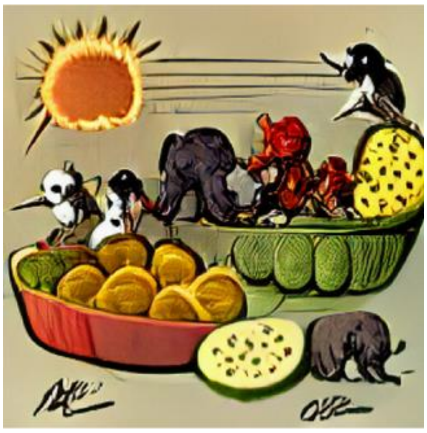


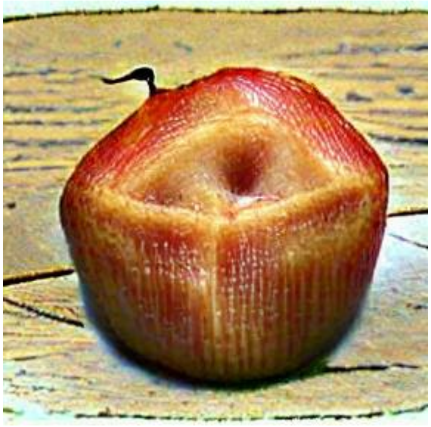
Both images were generated using 'a house' as the text prompt. The image on the left was generated without using an initial image, while the one on the right utilized an initial image. It's clear that the image generated with an initial image produces better results. Fewer words could lead to lower-quality generated images.

2. CLIP
   a. Fewer words could lead to lower-quality generated images.

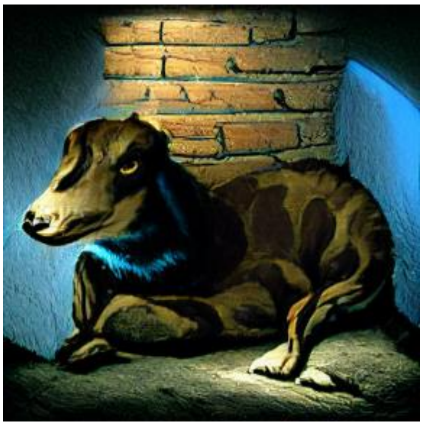
This image was generated using the text prompt 'a house with realistic style' with initial image and resulted in a satisfying outcome.

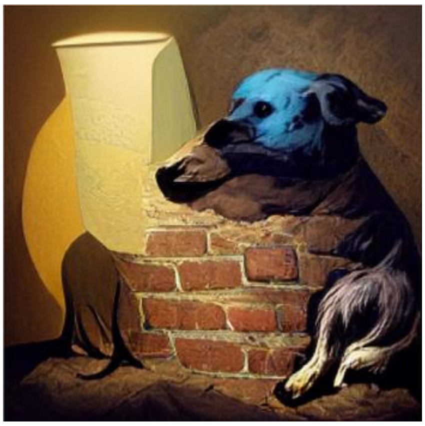
This image was generated using 'a reality dark puppy' as the text prompt, also with initial image, and it's evident that the generated image quality is poorer. Therefore, we believe enhancing the level of detail in the description might provide CLIP with a better basis for assessment.





The image on the right, using “an apple” as the text prompt, although vaguely recognizable as an apple, displays poor quality. On the left is our initial test result, generated using “A comic-style illustration of a fruit platter in sunlight, with other animals attempting to grab some” as the text prompt. Despite lacking an initial image and having some unclear aspects, the overall outcome is quite impressive.





These are our next test results. Both were generated using “A realistic looking dog sitting in a dark corner against a wall, with an overall color scheme in cool tones” as the text prompt. The image on the left utilized an initial image, while the one on the right did not. As compared to the previous image, these show more details and the imagery generated with the style of an initial image seems superior.

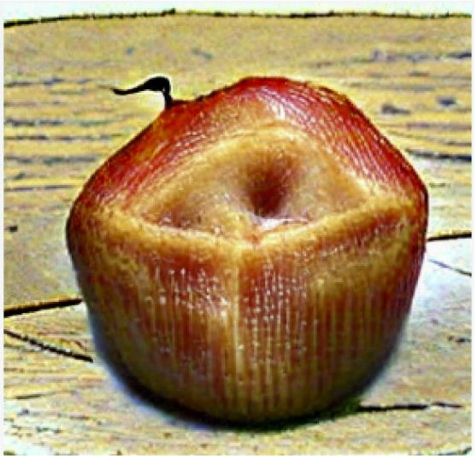
b. Fine-tune

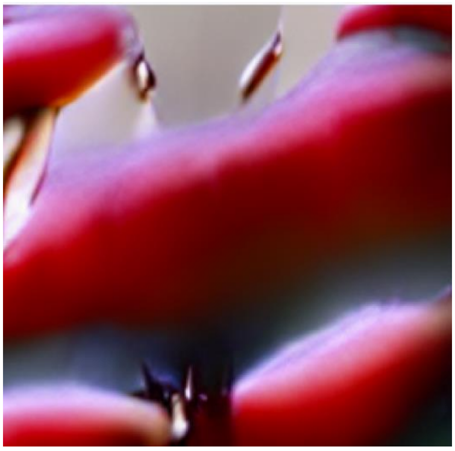
We tried to finetune our Clip model to get better image in specific domain. We use additional dataset on Kaggle.

https://www.kaggle.com/datasets/kritikseth/fruit-and-vegetable-image-recognition

Hoping to get better result in fruit and vegetable image.

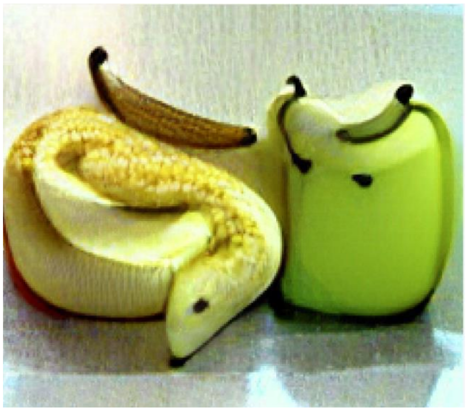
Below is our result. (text = apple)

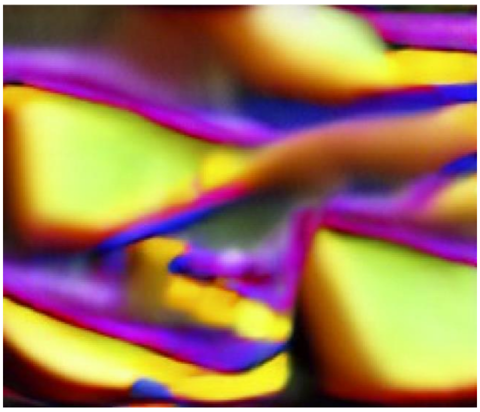




Here are our result images. On the left, we have the outcomes from the model without fine-tuning, while on the right, we employed fine-tuning. It's evident that the results generated by the fine-tuned model are completely inadequate.

Next result. (text = apple and banana)





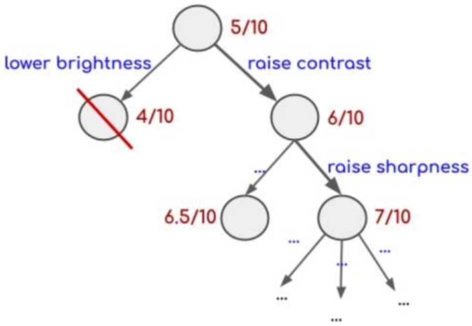
We also observed that the model seems to generate corresponding images only for specific text inputs. Our experiments involved simple textual descriptions like individual fruit names, where inputting 'APPLE' would at least produce an image resembling an apple. However, when we input a more descriptive phrase like 'a picture of an apple' or ‘apple and banana’ the model generated entirely blurred images.

Improvement: Image enhancement

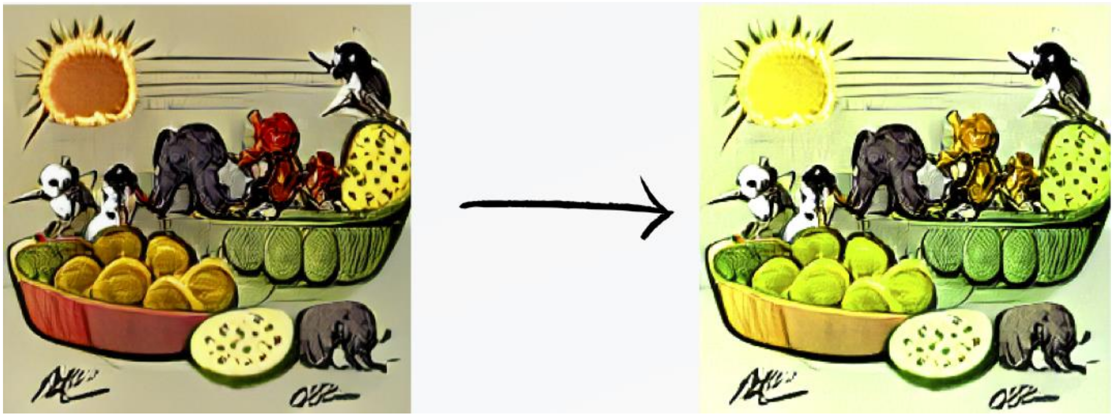
There are multiple ways to enhance an image. It can be done automatically with a simple mathematical analysis of the information in the image (“The sum of pixels is too low, it’s too dark, the brightness needs to be changed”). It can be carried out on the user’s decision with a simple mathematical analysis

("This image would be much better with a black and white filter"). And it can be done with deep learning (DL) algorithms, based on user's decision or automatically.

We use an "aesthetic predictor" on Github. It does what it says: it predicts how "aesthetic" an image is. The concept is adding a decision tree on top of the previous image, and you approximately have the algorithm the author implemented. Each node is a bunch of transformations and I try new transformations on the best nodes of the graph. Like below.



Below is our result.



As you can see the color and brightness of the image does change, however this image enhancement algorithm capable of adjusting brightness, saturation, contrast, and so on, but it doesn't alter the objects within the image. While our model sometimes might generate unrecognized patterns, alter the aesthetic score doesn't improve the image in our situation. Consequently, this algorithm seems less suitable for our purposes.

● Reflection & Conclusion

In our reflection on this semester's project, we found that fine-tuning this model was quite challenging due to its size and the complexity of generating images. Such tasks often hinge on minor errors or improper configurations, which can drastically deviate the outcomes from our intended expectation.

Providing an initial image tends to yield better results in image generation. Typically, the generated image takes shape within the initial few thousand iterations. Both excessively large and small iteration counts tend to yield subpar image results. In summary, this is a novel model, and image generation has remained an intriguing and popular application for a long time. In the future, we aim to delve deeper into exploring these aspects.

Contribution:

1. Generate image process: Both
2. Finetuning model & Command:杜家漢
3. Clip-VQGAN Command :宋冠毅
4. Image enhancement model command:杜家漢
5. PowerPoint: Both

● Reference

https://hyugen-ai.medium.com/how-i-built-an-ai-image-enhancer-f7d77186678e

https://medium.com/geekculture/text-to-image-synthesis-using-multimodal-vqgan-clip-architectures-896b8a6588ef

https://github.com/nerdyrodent/VQGAN-CLIP

https://link.springer.com/chapter/10.1007/978-3-031-19836-6_6

https://medium.com/aimonks/a-guide-to-fine-tuning-clip-models-with-custom-data-6c7c0d1416fb