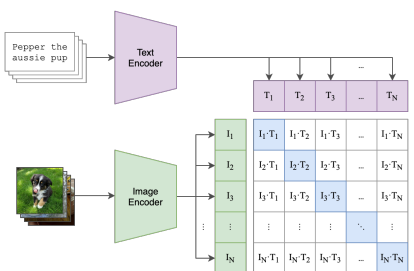
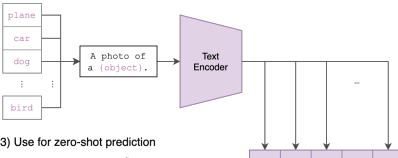
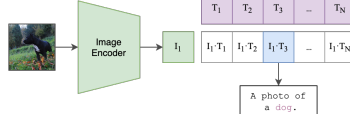


实验报告

|           |   |    |           |      |            |
|-----------|---|----|-----------|------|------------|
| 课程名称      | 人工智能综合实践  | 班级 | 计算机科学与技术  | 实验日期 | 2024.12.10 |
| 姓名        | 孙悟空   | 学号 | 123456789 | 队友   | 六耳猕猴       |
| 实验名称      | 凌霄宝殿建设与维护关键技术研究   |    |           |      |            |
| 实验目的及要求   | (本次实验所涉及并要求掌握的知识点)  |    |           |      |            |
| 实验环境      | (本次实验所使用的平台和相关软件)   |    |           |      |            |
| 实验内容      | (按照本组完成的实现内容和方式，对照实验要求描述，并自评是否达到实验要求)   |    |           |      |            |
| 算法描述及实验步骤 | <div>(适当的 形式表达算法设计思想与算法实现步骤，注意尽量形式化表达，鼓励使用流程图、伪代码等形式)</div> <div><div><p>(1) Contrastive pre-training</p></div><div><p>(2) Create dataset classifier from label text</p></div><div><p>(3) Use for zero-shot prediction</p></div></div> <p>Figure 1: CLIP</p> <pre>import torch import clip from PIL import Image</pre> |    |           |      |            |

|           |  |
|-----------|--|
|           | <pre> device = "cuda" if torch.cuda.is_available() else "cpu" model, preprocess = clip.load("ViT-B/32", device=device)  image = preprocess(Image.open("CLIP.png")).unsqueeze(0).to(device) text = clip.tokenize(["a diagram", "a dog", "a cat"]).to(device)  with torch.no_grad():     image_features = model.encode_image(image)     text_features = model.encode_text(text)      logits_per_image, logits_per_text = model(image, text)     probs = logits_per_image.softmax(dim=-1).cpu().numpy()  print("Label probs:", probs) # prints: [[0.9927937  0.00421068 0.00299572]] </pre> |
| 调试过程及实验结果 | <p>（详细记录在调查过程中出现的问题及解决方法。记录实验执行的结果，对实验结果进行分析，对于效果满意或者欠佳的原因进行分析）</p>  |
| 总结        | <p>（对本次实验的收获和心得体会进行总结）</p>   |
| 附录        | <p>（代码 readme 文件，包括源程序清单，以及每个文件的内容简介和如何运行的介绍）</p>  |