实验3：基于 CLIP 的图文检索

1. 实验目标

- 了解 CLIP 的基本思想：把图像与文本映射到同一语义空间。
- 掌握图文检索流程：特征提取 → 相似度检索 → 指标评测（Recall@K）。

2. 环境准备

- 新建虚拟环境
- 安装所需要的工具包
  - `transformers`与`modelscope`（或者`openai-clip`）：获取CLIP模型
  - `scikit-learn`：计算指标
  - `matplotlib`：可视化
  - `pillow`：读取图片数据

3. 数据集

（1）Flickr8k

Flickr8k 是一个经典的**图像描述数据集**，由 Hodosh 等人在 2013 年提出。它的核心目标是为研究**视觉与语言的结合**提供基础资源。

- **规模**：包含 **8000 张图像**，所有图像均来自 **Flickr** 平台的日常生活照片。
- **图像内容**：大多是人物或动物的自然场景，例如：小孩玩耍、狗奔跑、人物互动等。
- **描述文本**：每张图像配有 **5 条英文描述**，这些描述来自人工标注人员的自然语言输入，覆盖图像中的不同要点与角度。
- **总文本量**：约 **40,000 条句子**（8000 × 5）

（2） 数据获取

- 获取方式
  - kaggle
    - https://www.kaggle.com/datasets/adityajn105/flickr8k

- Hugging face Datasets
  - https://huggingface.co/datasets/jxie/flickr8k
- github
  - https://github.com/jbrownlee/Datasets/releases/download/Flickr8k/Flickr8k_Dataset.zip
  - https://github.com/jbrownlee/Datasets/releases/download/Flickr8k/Flickr8k_text.zip

（3） 数据组织格式

- 数据目录

```
data/
  images/
    0001.jpg
    0002.jpg
    ...
  captions.json        # 结构示例：
```

```
[
  {"image": "0001.jpg", "captions": ["a dog running on grass", "a brown dog in a field", "..."]},
  {"image": "0002.jpg", "captions": ["two people riding bicycles", "cyclists on a road", "..."]}
]
```

- 数据划分
  - **训练集**：约 6000 张图像
  - **验证集**：约 1000 张图像
  - **测试集**：约 1000 张图像

4. 实验任务

- **图文检索（双向）**
  - *Text→Image*：给定一句描述，在图片库中检索最相关的图片。
  - *Image→Text*：给定一张图片，在全部候选文本中检索最相关的描述。
- **评价指标**
  - Recall@K（K ∈ {1,5,10}）：正确目标命中前 K 的比例。
- 要点
  - 归一化：CLIP 特征在计算余弦相似度前做 L2 归一化效果更好。
  - 缓存：编码后文本特征和图像特征可保存复用。
  - **多正样文本**（Image→Text）：应将同一图对应的 **所有文本描述** 都当作正确答案。
  - 对文本编码器输入中的 **prompt** 进行精心设计是有效的。

5. 实验报告要求

1. **摘要（100–150字）**：说明问题、方法、数据、主要结果。
2. **方法**：
   - CLIP 简要原理（共享嵌入空间、对比学习、余弦相似度）。
   - 本实验的流程图（可画“图像/文本 → Encoder → 特征 → 相似度 → 排序”）。
3. **数据与实现细节**：
   - 数据集与规模（数据划分方式）。
   - 预处理与超参（归一化策略、是否缓存）。
   - 计算资源（CPU/GPU）、运行时长。
4. **结果**：
   - 指标表：Text→Image 与 Image→Text 的 Recall@1/5/10。
   - 前 K 检索可视化若干（成功+失败样例）。
5. **分析与讨论**：
   - 错误案例原因、类别/文本长度对性能的影响。
   - （可选）不同 Prompt 或不同子集规模对结果的影响（画简单曲线/柱状图）。
   - 可扩展优化的方向（如加入 FAISS、融合视觉特征等）。
6. **附录**：关键代码片段、环境版本、可复现实验指令。

6. 评分方式

- **实现正确性（35）**：能完成双向检索与指标计算；代码结构清晰、可复现。
- **结果与可视化（25）**：指标完整、Top-K 示例清晰、图表规范。
- **分析深度（20）**：对成功/失败案例的解释到位；有对 Prompt/规模等的实证分析更佳。
- **报告质量（10）**：表达清楚、结构完整、图表标注规范。
- **加分项（10）**：自行探索优化方法或技术，并在报告中详细说明，并加粗显式标出来