引言 (Introduction)

隨著即時通訊工具（如 LINE 和 Messenger）的普及，圖片分享已成為日常對話中不可或缺的一部分。本次作業的目標是基於對話內容檢索相關的圖片，這是一項多模態檢索任務，結合了自然語言處理（NLP）和計算機視覺（CV）技術。本報告將詳細描述整個系統的架構設計與實作細節。

方法 (Methods)

1. 資料處理 (Data Processing)

資料處理的主要目標是從提供的 `.jsonl` 資料中提取對話文字與圖片，並對其進行適當的預處理。

1. **資料讀取**
    - 使用 `load_jsonl` 函式讀取 `train.jsonl`、`test.jsonl` 和 `test_images.jsonl` 資料檔案。
    - 每一行是一個 JSON 字典，訓練資料包含 `dialogue` 和 `photo_id`，而測試資料僅包含對話。
2. **資料結構與預處理**
    - **對話文字：**
        - 合併多輪對話的 `message` 欄位，生成一段完整文字描述，方便後續的 NLP 模型處理。
        - 使用 BERT 分詞器將文字轉換為數值向量。
    - **圖片：**
        - 圖片被調整為 224x224 的固定大小，並進行標準化處理。
        - 加入隨機水平翻轉作為資料增強策略。

程式碼範例：

```
def merge_dialogue_messages(dialogue):
    """將 dialogue 列表中所有 message 合併為單一字串"""
    messages = [turn['message'] for turn in dialogue if turn['message'].strip() !=
'']
    return " ".join(messages) if messages else ""
```

---

2. 模型設計 (Model Design)

本次作業採用雙編碼器模型（Dual Encoder），設計目的是將文字與圖片嵌入到相同的特徵空間中，以便計算相似度。

1. **文字編碼器 (Text Encoder)**
    - 使用 BERT 模型將對話文字轉換為特徵向量，並通過線性層進一步投影到圖片特徵空間的維度。
    - 特徵空間維度為 2048。

程式碼範例：

```
def encode_text(self, input_ids, attention_mask):
    outputs = self.text_encoder(input_ids=input_ids,
attention_mask=attention_mask)
    cls_emb = outputs.last_hidden_state[:, 0, :]  # 提取 [CLS] token 的嵌入
    return self.text_proj(cls_emb)
```

2. **圖片編碼器 (Image Encoder)**
   - 使用 ResNet50 模型提取圖片的特徵，最後一層（分類層）被替換為一個線性投影層，將特徵對齊到相同的 2048 維嵌入空間。

程式碼範例：

```
def encode_image(self, image):
    img_emb = self.image_encoder(image)
    return self.image_proj(img_emb)
```

---

3. 訓練策略 (Training Strategy)

1. **損失函數**
   - 採用對比學習的交叉熵損失函數（Cross-Entropy Loss）。
   - 損失函數鼓勵正確的圖片-對話配對向量靠近，並推開不正確的配對。

2. **學習率調整**
   - 使用餘弦退火學習率排程器（CosineAnnealingLR）動態調整學習率。

程式碼範例：

```
logits = torch.matmul(text_emb_norm, image_emb_norm.T) / TEMPERATURE
labels = torch.arange(logits.size(0)).to(device)
loss = criterion(logits, labels)
```

---

4. 推論階段 (Inference)

1. **嵌入生成**
   - 在測試階段，為每張圖片生成嵌入向量，並將其儲存。
   - 對於每段對話，生成其嵌入向量並計算與所有圖片的相似度。

2. **檢索策略**
   - 根據相似度，選出對話對應的 Top-30 圖片，作為檢索結果。

程式碼範例：

```
similarity = torch.matmul(text_emb_norm, all_img_embs.T).squeeze(0)
topk_vals, topk_indices = torch.topk(similarity, TOP_K)
top_photo_ids = [all_img_ids[i] for i in topk_indices.cpu().tolist()]
```

結果與討論 (Results and Discussion)

模型效能

- 本次作業的檢索性能指標為 Recall@30，該指標衡量正確圖片是否在 Top-30 的檢索結果中出現。
- 檢索性能未顯示具體數值，以待模型進一步調整。

分析

1. **模型優勢**
    - 雙編碼器模型有效地對齊圖片與文字的特徵空間。
    - 對比學習策略顯著提升檢索準確率。
2. **改進空間**
    - 增加多模態資料（如場景標籤）進一步提升模型的語意捕捉能力。
    - 考慮使用更強的視覺語言模型（如 CLIP）。

結論 (Conclusion)

本次作業成功構建了一個基於對話的圖片檢索系統，採用雙編碼器模型實現文字與圖片的特徵對齊，為未來進一步的多模態檢索研究提供了良好的基礎。未來的改進方向包括增強數據集的多樣性與採用更先進的模型架構。
