Automatic Image Captioning with Robustness Analysis and Model Identification

Course: CS60010 – Deep Learning Assignment 2
Team ID: 19
Team Members: Shshank Mittal(22CS30051), Sujan Jain(22CS10075), Gaurang Gupta(22CS30025)

---

1. Introduction

This project presents an end-to-end solution for automatic image captioning comprising three parts:

- **Part A:** Implementation and benchmarking of a custom transformer-based encoder-decoder model for image captioning, compared with a zero-shot baseline (SmolVLM).
- **Part B:** Robustness analysis by evaluating model performance under image occlusion.
- **Part C:** A BERT-based classifier to distinguish between captions generated by SmolVLM and our custom model.

The overall goal is to develop accurate and robust models while gaining insights through comprehensive evaluation metrics.

---

2. Methodology

2.1 Part A: Custom Encoder-Decoder Model & Zero-shot Baseline

2.1.1 Zero-shot Captioning with SmolVLM

We implemented the function `zero_shot_captioning(image_path, model_name="SmolVLM")` that uses the pre-trained SmolVLM model to generate captions from input images without further training. Evaluation of the test set is done using BLEU,

ROUGE-L, and METEOR metrics. This baseline provides a reference point for our custom model's performance.

2.1.2 Custom Encoder-Decoder Model

Encoder:

- **Model:** We use ViT-Small-Patch16-224 to extract rich visual features. The input image is divided into fixed-size patches (16×16 pixels) and then projected into an **embedding space of 768 dimensions (16×16×3)**.

- **Transformer Layers & Positional Encoding:** The patches are processed through transformer layers using self-attention, with added positional encodings to retain spatial information. This results in a robust representation of the visual content. `encoder_outputs = self.encoder(images).last_hidden_state.` This line does all of the above. **The output vector has dimension 197 (196 patches + [CLS] token).**

Decoder:

- **Architecture:** This architecture uses a GPT-2 transformer decoder to generate captions autoregressively, conditioned on image features from a ViT encoder. The ViT's output **(197×768) is projected to match GPT-2's embedding dimension (768)**. The decoder processes combined image-caption embeddings (217×768), then predicts logits **(seq_len×50,257)** for next-token prediction.

Architecture Diagram: [Custom Model]

- **Training Strategy:**
  The `train_model()` function implements training using an Adam optimizer and cross-entropy loss. In forward pass, the output of Vit encoder is concatenated with embeddings of GPT-2 decoder and then passed into decoder using teacher forcing.
- **Why it works?** Concatenating image features (from ViT) with text embeddings (from GPT-2) before feeding them into the decoder enables **implicit cross-modal attention** in transformer-based models like GPT-2, because of **Position Embeddings and Unified Sequence Processing**

2.1.3 Evaluation

The `evaluate_model()` function computes BLEU, ROUGE-L, and METEOR scores on the test set, which we use to compare our custom model against the zero-shot SmolVLM baseline.

2.2 Part C: BERT-based Caption Classifier

2.2.1 Data Preparation

For each image, we construct a dataset entry of the format `f"{orig_cap} {tokenizer.sep_token} {gen_cap} {tokenizer.sep_token} {occlusion_level}"` as mentioned in the assignment.

2.2.2 Classifier Architecture: Custom Bert

- **BERT Layer**: A pre-trained BERT model processes input text (tokenized as input_ids with an attention_mask).
- **Dropout Layer**: Reduces overfitting by randomly zeroing 10% of activations.
- **Linear Layer**: Maps BERT's pooled output (768-dimensional by default) to num_classes logits.

2.2.3 Training and Evaluation

The classifier is trained using a **70:10:20 train-validation-test split** (ensuring no image overlap across splits). The model is trained using **AdamW, Linear Schedule with Warmup, and the loss function is CrossEntropyLoss**. Then, the model is evaluated on the Test set. We trained the model for **three epochs as further training led to overfitting**.

---

3. Results and Analysis

3.1 Part A: Image Captioning Performance (Occlusion 0%)

Model	BLEU	ROUGE-L	METEOR
SmolVLM(Zero-Shot)	0.026	0.236	0.175
Custom Encoder-Decoder	0.034	0.263	0.202

***Observation**:* Our custom model performs better than the zero-shot baseline, achieving higher scores across all three key metrics.

3.2 Part B: Robustness Under Occlusion

SmolVLM Performance Under Occlusion:

Occlusion Level	BLEU	ROUGE-L	METEOR
10%	0.033891	0.265303	0.205605
50%	0.028987	0.259095	0.196746
80%	0.021467	0.238696	0.179335

Custom Encoder-Decoder Performance Under Occlusion:

Occlusion Level	BLEU	ROUGE-L	METEOR
10%	0.022232	0.232318	0.173778
50%	0.015276	0.204462	0.150298
80%	0.008215	0.166953	0.116629

Observation: Both models experience performance degradation as occlusion increases; however, the custom model has slightly better results at each occlusion level than SmolVLM.

3.3 Part C: Caption Classifier Performance

Validation Set Accuracy: 97.98%
Test Set Accuracy: 98.38%

	Precision	Recall	F1-Score
0	0.98	0.99	0.98
1	0.99	0.98	0.98
Accuracy	-	-	0.98
Macro avg	0.98	0.98	0.98
Weighted avg	0.98	0.98	0.98

Observation:
The BERT-based classifier successfully distinguishes between captions generated by SmolVLM and our custom model, even under varying occlusion levels, demonstrating that the generated captions have distinguishable stylistic and semantic features.

---

4. Conclusion

This project developed an effective pipeline for automatic image captioning with robust performance analysis and model identification. Our custom transformer-based encoder-decoder model outperformed the zero-shot SmolVLM baseline, and the occlusion experiments confirmed its relative robustness. Furthermore, the BERT-based caption classifier effectively discriminates between the outputs of the two models.