Multimodal Image Captioning

Ankit Pratap Singh

1 Introduction

Image captioning is the process of generating textual descriptions for input images. This requires a combination of **Natural Language Processing (NLP)** and **Computer Vision (CV)**. In this project, we implement an **image-to-text model** using **Vision Transformers (ViT) and GPT-2**.

2 Dataset

We use the **COCO (Common Objects in Context) dataset**, a large-scale dataset with over 120K images and their respective captions.

3 Model Architecture

This project uses a **Vision-Text Transformer** pipeline with an **encoder-decoder structure**:

- **Encoder (ViT)**: Extracts visual features from an image.
- **Decoder (GPT-2)**: Generates a sequence of words based on the visual features.
- **Training Process**:
  - The encoder converts the image into a feature representation.
  - The decoder autoregressively generates captions word-by-word.
  - The model is fine-tuned using a sequence-to-sequence (Seq2Seq) approach.

3.1 Encoder-Decoder Representation

Architecture Representation:

```
Image → [ViT Encoder]  Feature Vector  [GPT-2 Decoder] → Generated Caption
```

3.2 What is an Encoder-Decoder Model?

An **Encoder-Decoder model** is a type of neural architecture used in many sequence-to-sequence tasks. It consists of:

- **Encoder**: Processes the input (an image in this case) and generates a latent space representation (feature vector).
- **Decoder**: Takes the feature vector and generates an output sequence (a caption in this case).

This model architecture is commonly used in tasks like machine translation, summarization, and image captioning.

4 Code Files

4.1 `model.py`

- **Function:** `setup_model()`
- Initializes the **Vision Transformer (ViT) and GPT-2** model.
- Loads the **tokenizer** and **feature extractor**.
- Saves the model in `vit-gpt2-model/`.

4.2 `data.py`

- **Function:** `download_and_process_data()`
- Downloads and preprocesses the **COCO 2017 dataset**.
- Converts **images into feature vectors** using ViT.
- Tokenizes **text captions** using GPT-2.

4.3 `train.py`

- **Functions:** `compute_eval_metrics()`, `refine_text()`
- Loads the model and dataset.
- Uses **Hugging Face's Seq2SeqTrainer** for training.
- Computes **ROUGE Score** for evaluation.

5 Training Results

Here are some generated captions from the trained model:

Image	Generated Caption
	"Closeup of bins of food that include broccoli and bread."
	"Various slides and other footwear rest in a metal basket outdoors."

	“A picture of a dog laying on the ground.”