IMAGE-TO-TEXT GENERATOR

A Mini Project

Submitted by

Tanvi Rainak (Exam Seat No. B400950177)

Piyush Shastri (Exam Seat No. B400950195)

Ritesen Dhar (Exam Seat No. B400950180)

Jayshree Karve (Exam Seat No. B400950130)

FINAL YEAR COMPUTER ENGINEERING

Department of Computer Engineering
International Institute of Information Technology
Hinjawadi, Pune – 411057

SEMESTER II (AY 2024-25)

TABLE OF CONTENTS

1. ABSTRACT

This project presents an interactive web-based application for generating detailed paragraph-style descriptions of images using state-of-the-art vision-language models. Built with Streamlit, the system provides a user-friendly interface for uploading images and viewing AI-generated textual descriptions in real time. The core of the application leverages the BLIP-2 (Bootstrapped Language-Image Pretraining) model — specifically the blip2-opt-2.7b variant — sourced from Hugging Face Transformers. Upon image upload, the model performs multimodal reasoning to produce rich and context-aware captions in natural language. The backend is optimized with PyTorch and supports both CPU and GPU execution. Additionally, the app utilizes Pyngrok to create a secure public tunnel, enabling remote access without traditional server deployment. This project demonstrates the potential of combining modern deep learning models with lightweight web frameworks to deliver powerful AI capabilities in an accessible and deployable format.

2. INTRODUCTION

2.1 Problem Definition And Objectives

While several image captioning systems exist, most are limited to short, single-sentence outputs and are not easily deployable by individuals without deep technical expertise. There is a need for a user-friendly solution that can generate highly accurate and descriptive captions for visual content, and is easily accessible via the web

2.2 Scope

- This project focuses on the integration of state-of-the-art AI models for image captioning into a web application.
- It provides a proof of concept for educational, accessibility, and visual documentation purposes.
- The current version generates captions for single images uploaded by the user, but the framework allows for future extensions like batch processing, user authentication, and database storage.

2.3 Requirement Analysis

Functional Requirements:

- Upload an image through the Streamlit interface.
- Generate a paragraph-style caption using the BLIP-2 model.
- Display the uploaded image and generated description.
- Provide a public URL for access using Pyngrok.

Non-Functional Requirements:

- Quick response time for small to medium-sized images.
- Support for both CPU and GPU execution.
- Easy to use and visually minimal interface.

2.4 Software And Hardware Details

Software:

- OS: Any OS that supports Python (Linux, Windows, macOS)
- Programming Language: Python
- Framework: Streamlit

- Deep Learning Framework: PyTorch
- Model Source: Hugging Face Transformers
- Deployment Tool: Pyngrok

Hardware:

- Minimum:
  - CPU with 4 cores
  - 8 GB RAM
- Recommended (for faster inference):
  - CUDA-enabled GPU
  - 16 GB RAM

2.5 Libraries / Packages Used

- **streamlit** – for building the web interface
- **transformers** – to load BLIP-2 model and processor
- **torch** – for running the model inference
- **torchvision** – for image handling and model compatibility
- **PIL (Pillow)** – to load and convert uploaded images
- **pyngrok** – to create a public tunnel for the app
- **threading, os, time** – for background process management and runtime control

3. DATASET DETAILS

This project does not rely on a preloaded or custom dataset for training, as it uses a pre-trained model — BLIP-2 (blip2-opt-2.7b) — which has already been trained on large-scale datasets by the original authors (Salesforce) and made available through Hugging Face. However, understanding the kind of data the model was trained on gives insight into its capabilities and limitations.

3.1 Model Training Dataset (BLIP-2)

The blip2-opt-2.7b model was trained on a combination of large vision-language datasets, including but not limited to:

- **LAION-400M/LAION-5B** – Large-scale dataset with image-text pairs sourced from the internet.
- **COCO (Common Objects in Context)** – Annotated dataset with over 330,000 images and multiple captions per image.
- **Conceptual Captions (CC3M, CC12M)** – Web-crawled image and alt-text pairs for vision-language learning.
- **Visual Genome** – Dataset with dense captions and region descriptions.
- **SBU Captions** – 1 million image-caption pairs sourced from Flickr.

These datasets collectively provide millions of diverse images with associated natural language descriptions, enabling the model to generalize to various visual content.

3.2 User Input as Real-Time Data

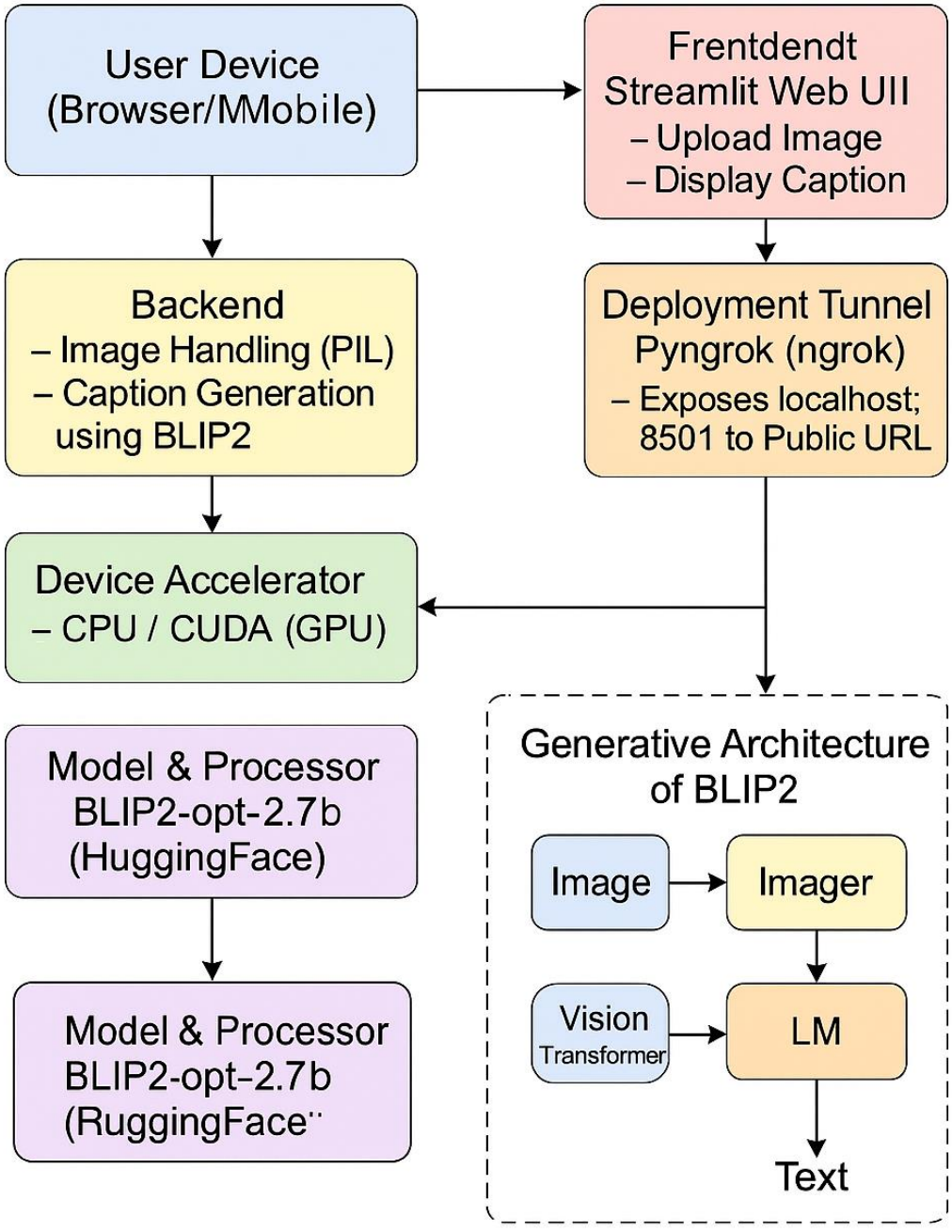
Although no training dataset is used directly in this project, the input images uploaded by users during runtime act as test data for inference. The system processes any uploaded image and generates a relevant caption using the pre-trained model, simulating real-world applications.

3.3 Advantages of Using a Pre-trained Model

- No need to manually collect or annotate a dataset.
- Faster development and deployment since training is not required.
- High accuracy and generalization due to training on massive datasets.
- Supports transfer learning and adaptation if future fine-tuning is needed.

4. SYSTEM ARCHITECTURE

4.1 Architecture Diagram



Generative Architecture of BLIP2

4.2 Overview of Project Modules

- **Web Interface (Streamlit):**
  - Handles user interaction.
  - Allows image upload.
  - Displays uploaded image and generated paragraph.
- **Image Upload & Preprocessing:**
  - Accepts image file (jpg/png/jpeg).
  - Uses PIL.Image to convert to RGB format.
  - Prepares input for the model.
- **Model Integration (BLIP-2):**
  - Loads Blip2Processor and Blip2ForConditionalGeneration from Hugging Face.
  - Sends preprocessed image to the model.
  - Uses GPU if available for faster inference.
- **Caption Generation:**
  - Uses model's .generate() to get token output.
  - Decodes token sequence into paragraph-style text.
- **Display Module:**
  - Renders generated paragraph in the Streamlit interface.
  - Allows users to read, copy, or reuse the description.
- **Deployment (Ngrok):**
  - Creates a secure tunnel to run the app over the internet.
  - Allows remote testing or demo without external hosting.

4.3 Algorithm Details

The key algorithmic steps are as follows:

1. **Image Preprocessing:**
   - image = Image.open(uploaded_file).convert('RGB')
   - inputs = processor(images=image, return_tensors="pt").to(device)
2. **Caption Generation using BLIP-2:**
   - Load model:

     model = Blip2ForConditionalGeneration.from_pretrained(...)
   - Generate text IDs:

     generated_ids = model.generate(**inputs, max_new_tokens=550)

- Decode IDs to text:
  generated_text = processor.batch_decode(generated_ids, skip_special_tokens=True)[0]

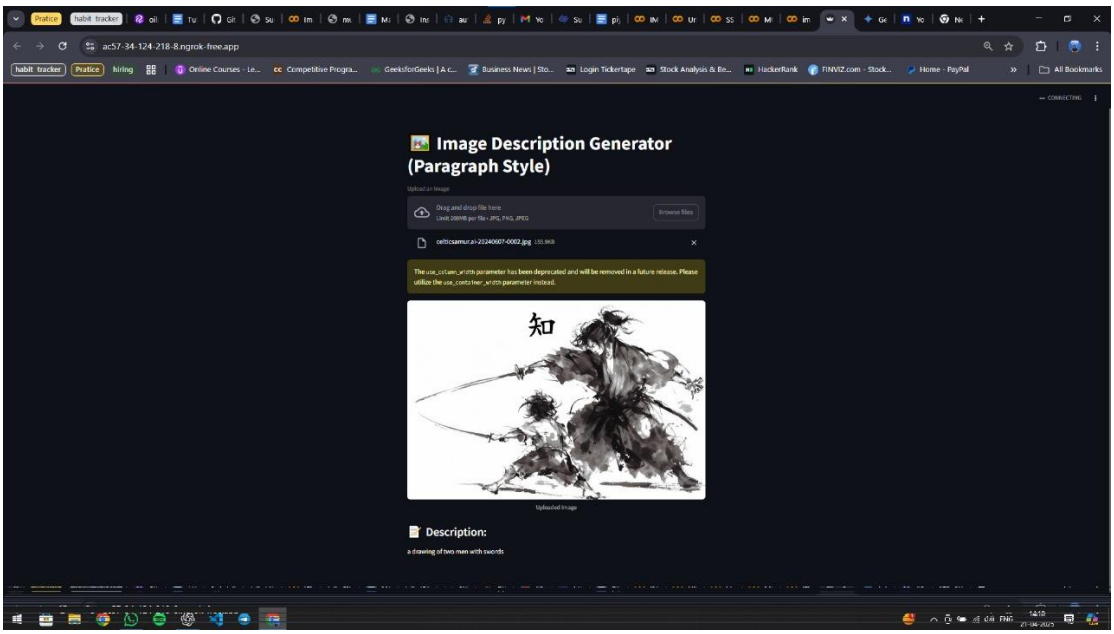
3. **Output Display:**
   - st.write(generated_text) shows the result to the user.

4. Result

The Image Captioning system was successfully implemented and tested using the Streamlit interface and the BLIP-2 (blip2-opt-2.7b) model. The following outcomes were observed:

- **Accurate Paragraph Descriptions**: The model generated detailed and coherent paragraph-style descriptions for various images, including natural scenes, objects, and human activities.
- **User-Friendly Interface**: The Streamlit UI provided a smooth and interactive experience for uploading images and viewing results.
- **Real-Time Inference**: The use of PyTorch and GPU (if available) ensured fast caption generation, with most outputs returned in under 10 seconds.
- **Deployment Ready**: Integration with Pyngrok allowed instant web sharing without hosting infrastructure, making the project easily accessible for testing and demo purposes.

6. GRAPHICAL USER INTERFACE



7. CONCLUSION

This project demonstrates the practical implementation of AI-powered image understanding using a pre-trained multimodal transformer (BLIP-2) model. It showcases the capabilities of modern vision-language models to generate context-rich, paragraph-style descriptions from a single image input.

The key takeaways are:

- The system requires no training dataset, making it ideal for rapid deployment.
- The use of Streamlit and Ngrok makes the application highly accessible and demo-ready.
- It can be extended further into applications such as visual storytelling, accessibility tools for the visually impaired, or automated content generation for media platforms.

This project highlights the power of pre-trained generative AI models and serves as a solid foundation for future enhancements like multilingual support, audio narration, or fine-tuning for specific domains (e.g., medical or satellite imagery).