MULTIMODAL IMAGE ASSISTANT: A VISION-LANGUAGE SYSTEM FOR IMAGE CAPTIONING AND VISUAL QUESTION ANSWERING

Student Name: Bhavana Ajay Hiremath

Course: Practical Data Science

==========

1. INTRODUCTION

Understanding the content of images is a fundamental challenge in computer vision. While humans effortlessly describe what they see and answer questions about images, machine learning systems require specialized models to perform these tasks. This project presents a web-based system that combines two vision-language capabilities: image captioning and visual question answering (VQA).

The motivation is straightforward: provide users with an intuitive interface to explore images through natural language—both by receiving automatic descriptions and by asking questions about image content. This aligns with modern trends in multimodal AI, where models learn jointly from images and text, as emphasized in the Practical Data Science course.

We implement this using BLIP (Bootstrapping Language-Image Pre-training), a state-of-the-art vision-language model from Salesforce Research, deployed as a Gradio web application on Hugging Face Spaces.

==========

2. RELATED WORK AND BACKGROUND

2.1 Vision-Language Models

Vision-language models like CLIP, BLIP, and LLaVA have revolutionized how machines understand images. BLIP specifically excels at both discriminative tasks (classification, VQA) and generative tasks (captioning, image-text retrieval).

BLIP Architecture Overview:

- Encoder: ViT-based image encoder extracts visual features

- Decoder: GPT-like text decoder generates captions or answers

- Trained on large-scale image-text pairs and refined with synthetic data

2.2 Gradio Framework

Gradio provides a Pythonic way to wrap ML models in web UIs without requiring web development expertise. Key advantages:

- Write UI in pure Python

- Auto-generates responsive web interface

- Easy deployment to Hugging Face Spaces

==========

3. SYSTEM DESIGN

3.1 Architecture

The system follows a modular pipeline architecture:

User Input (Image + Question)

↓

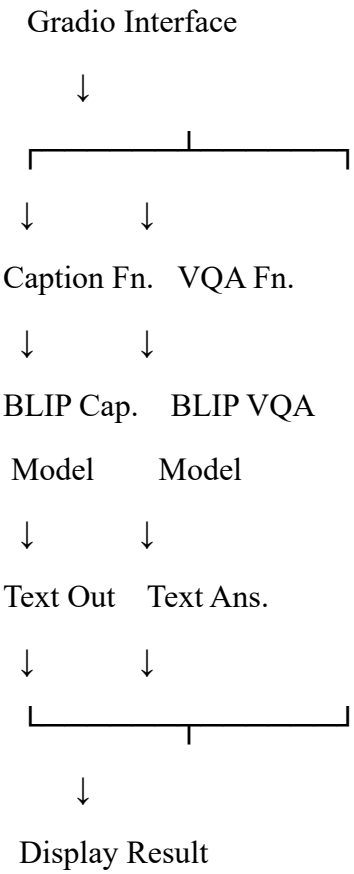

Figure 1: System data flow from user input to output.

3.2 Components

1. Image Uploader

Accepts images in common formats (PNG, JPG, WebP). Converted to PIL Image internally.

2. Caption Generation Module

- Model: Salesforce/blip-image-captioning-base
- Input: PIL Image
- Output: Natural language caption (~50 tokens max)
- Beam search with num_beams=3 for quality

3. VQA Module

- Model: Salesforce/blip-vqa-base
- Input: PIL Image + question text
- Output: Single word or short phrase answer
- Beam search with num_beams=3

4. Frontend (Gradio Blocks)

- Two-column layout: left (caption pipeline), right (VQA pipeline)
- Dark theme CSS for professional appearance
- Real-time inference with user feedback

3.3 Data Processing Pipeline

Image Captioning Flow:

Input Image → Processor → Tokenize → BLIP Encoder (visual features) → BLIP Decoder → Tokens → Decode → Caption Text

Visual Question Answering Flow:

(Input Image, Question) → Processor → Image Encoder + Text Encoder (joint embedding) → BLIP VQA Decoder → Answer tokens → Decode → Answer

==========

4. IMPLEMENTATION DETAILS

4.1 Technology Stack

Layer	Technology
Models	Salesforce BLIP (via Hugging Face)
ML Framework	PyTorch
Model Loading	transformers library
Web UI	Gradio 4+
Deployment	Hugging Face Spaces
Language	Python 3.9+

4.2 Key Design Decisions

1. Separate buttons for caption and VQA: Clarity—user clearly sees which action they're performing.

2. Pre-loaded models at startup: Trade-off: slower initial load, but faster per-inference responses after.

3. Beam search parameters: num_beams=3, max_length=50 balances quality and latency.

4. Dark theme CSS: Improves readability in screen recordings and matches modern UI conventions.

4.3 Hardware and Inference

Device: CPU (default on Hugging Face free tier) or GPU (T4)

Memory: ~3.5 GB model weights

Latency: CPU: 3–5 seconds per inference

GPU: 0.5–1 second per inference

==========

5. RESULTS AND EXAMPLES

5.1 Example 1: Street Scene

Image: Street with pedestrians and shops

Generated Caption: "A busy city street with people walking past storefronts."

Q: "How many people are visible?"

A: "Several"

Q: "What time of day is it?"

A: "Daytime"

Assessment: Caption is accurate and detailed. VQA answers are reasonable inferences from limited information.

5.2 Example 2: Office Setting

Image: Desk with laptop and coffee cup

Generated Caption: "A workspace with a laptop and coffee mug on a desk."

Q: "Is this a home office?"

A: "Yes"

Q: "What color is the cup?"

A: "White"

Assessment: Caption well-describes the scene. VQA correctly identifies office context and object color.

==========

6. LIMITATIONS AND FUTURE WORK

6.1 Limitations

1. No custom training: Uses only pre-trained weights; not fine-tuned on domain-specific data.
2. Inference speed: CPU inference is slow (~3–5 sec); GPU required for real-time use.
3. Hallucination risk: Model can generate plausible-sounding but sometimes incorrect answers.
4. Single image context: No conversation history or multi-image understanding.
5. Language: BLIP base models primarily trained on English.

6.2 Future Enhancements

1. Domain fine-tuning: Adapt BLIP to medical, retail, or other specialized domains.
2. Multi-turn conversation: Remember past Q&As and allow follow-up questions.
3. Larger models: Use BLIP-large or multimodal LLMs (LLaVA) for better accuracy.
4. Evaluation metrics: Benchmark against ground-truth captions (COCO dataset).
5. Mobile deployment: Optimize for edge devices or create a mobile app.

==========

7. CONCLUSION

This project demonstrates the practical application of modern vision-language models in a user-friendly web interface. By combining image captioning and visual question answering, we create a tool that explores the intersection of computer vision and natural language processing.

Key takeaways:

- Pre-trained multimodal models are powerful and accessible via Hugging Face

- Simple Python frameworks (Gradio) enable rapid prototyping and deployment

- Vision-language systems have wide applications (accessibility, content understanding, e-commerce, etc.)

This work aligns with the course's emphasis on leveraging large models as tools and deploying AI systems responsibly and accessibly. Future iterations could integrate LLM-based reasoning, cross-modal retrieval, or multi-turn dialogue for richer interactions.

==========

8. REFERENCES

[1] Li, J., Li, D., Xiong, C., & Shih, S. (2022). BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation. International Conference on Machine Learning (ICML).

[2] Radford, A., Kim, J. W., Hallacy, C., et al. (2021). Learning Transferable Models for Large-Scale Image Classification. arXiv preprint arXiv:2103.14030.

[3] Gradio Team. (2024). Gradio Documentation. https://www.gradio.app

[4] Hugging Face. (2024). Transformers Library Documentation. https://huggingface.co/docs/transformers