Basic Multimodal AI with Ollama

CS450: Modern Software Engineering

Beyond Text: Vision and Language Together

Today's Topics

1. What is Multimodal AI?
2. Vision-Language Models (VLMs)
3. Image Understanding Tasks
4. Prompting for Vision Models
5. Practical Applications & Limitations

What is Multimodal AI?

Unimodal AI: Works with one type of data

- Text only (GPT, Claude)
- Images only (traditional CNNs)
- Audio only (speech recognition)

Multimodal AI: Combines multiple data types

- Text + Images (LLaVA, GPT-4V)
- Text + Audio (Whisper + LLM)
- Text + Video (Video understanding models)

Why Multimodal AI?

Humans are naturally multimodal:

- We see, hear, read, and speak
- Understanding requires multiple senses

New capabilities:

- "What's in this image?"
- "Is this safe for work?"
- "How many people are in the photo?"
- "Describe the scene for accessibility"

Vision-Language Models (VLMs)

Architecture Overview:

1. **Vision Encoder:** Converts image to embeddings
2. **Projection Layer:** Aligns vision and language spaces
3. **Language Model:** Generates text responses

```
Image → Vision Encoder → Projection → LLM → Text Response
         (visual tokens)   (alignment)  (generation)
```

Popular Vision Models

LLaVA (Large Language and Vision Assistant)

- Open source
- Good balance of speed and accuracy

Gemma3:

- Ollama version of Google's Gemini
- What we'll use in lab

How Vision Models "See"

Image Processing:

1. Image divided into patches (e.g., 16×16 pixels)
2. Each patch becomes a token
3. Tokens processed like text tokens
4. Model learns relationships between patches

Result: The model builds understanding from visual patterns

Image Understanding Tasks

1. Image Description (Captioning)

- Generate natural language descriptions
- Identify objects, scenes, actions

2. Visual Question Answering (VQA)

- Answer specific questions about images
- "Does something seem off with this image?"

3. Object Detection

- Identify and locate objects
- Count specific items

Image Understanding Tasks (cont.)

4. Image Classification

- Categorize images into predefined classes
- "Is this indoors or outdoors?"

5. Content Moderation

- Assess image appropriateness
- Detect unsafe content

6. OCR (Text Extraction)

- Read text from images
- Process documents, signs, screenshots

Temperature for Vision Tasks

Lower temperature (0.0-0.3):

- Factual descriptions
- Classification tasks
- Counting objects ("How many dogs are in this image?")

Higher temperature (0.5-0.8):

- Creative descriptions
- Storytelling about images ("Write a story inspired by this scene")

Zero-Shot Classification

No training examples needed!

```
prompt = """Classify this image into ONE category:
dog, cat, bird, or other animal

Return ONLY the category name."""

response = analyze_image(image_url, prompt)
```

Works by:

- Leveraging pre-trained knowledge
- Following natural language instructions

Few-Shot Style Prompting

Provide guidelines without actual image examples:

```
prompt = """Classify as INDOOR or OUTDOOR.

Guidelines:
- INDOOR: inside buildings, rooms, enclosed spaces
- OUTDOOR: natural settings, streets, open areas

Classification:"""
```

Helps by:

- Clarifying category boundaries
- Reducing ambiguity

Structured Output for Vision

Get consistent, parseable responses:

```
prompt = """List all objects in this image.
Format: object1, object2, object3"""
```

Or more structured:

```
prompt = """Analyze this image.
Return format:
Objects: [list]
Colors: [list]
Setting: [indoor/outdoor]"""
```

Visual Question Answering

Multiple question strategies:

Sequential (one at a time):

- More accurate
- Better context per question
- Slower (multiple API calls)

Batch (all at once):

- Faster
- May miss details
- Use for related questions

Content Moderation Example

```
prompt = """Analyze this image for content safety.

Check for:
1. Appropriate for general audience
2. No violent content
3. No explicit material
4. No harmful activities

Provide:
- Safety rating (Safe/Warning/Unsafe)
- Brief explanation
- Detected concerns (if any)"""
```

Common Challenges

1. Hallucination

- Models may "see" things that aren't there
- More common than text-only models
- Solution: Ask for confidence levels

2. Spatial Reasoning

- Difficulty with precise positioning
- "Left vs right" can be unreliable
- Solution: Use relative terms carefully

Common Challenges (cont.)

3. Small Text/Details

- May miss fine details
- Small text can be difficult
- Solution: Ask about specific regions

4. Abstract Concepts

- Better at concrete objects than emotions/mood
- Solution: Be specific about what to look for

Best Practices

1. Clear Instructions

- Specify what information you need
- Define output format

2. Verify Critical Information

- Don't trust counts/details blindly
- Use multiple questions for validation

Best Practices (cont.)

3. Image Quality Matters

- Clear, well-lit images work best
- Avoid excessive blur or compression

4. Iterative Refinement

- Start with broad questions
- Follow up with specific details

Real-World Applications

Accessibility:

- Image descriptions for visually impaired
- Alt text generation
- Scene understanding

E-commerce:

- Product categorization
- Quality control
- Inventory management

Real-World Applications (cont.)

Healthcare:

- Medical image analysis
- Diagnostic assistance
- Patient monitoring

Computer Security:

- Surveillance analysis
- Anomaly detection
- Safety monitoring

Limitations to Consider

1. Privacy & Ethics

- Be careful with personal images
- Consider consent and data protection

2. Accuracy

- Not perfect, especially for critical tasks
- Requires human oversight for important decisions

3. Cost & Speed

- Vision models are slower than text-only (larger computational requirements)

Lab time! 🖼️