Lecture 12.1: Multimodality

Bridging Vision and Language

Heman Shakeri

Where We've Been: Separate Modalities

Throughout this course, we've treated modalities separately:

Vision (Modules 4-6):

- CNNs learned spatial features
- Translation invariance
- Hierarchical representations
- Vision Transformers (ViT) [Module 10]

Language (Modules 7-10):

- RNNs/LSTMs for sequences
- Attention mechanisms
- Transformers dominate
- Self-supervised pretraining

But these lived in *separate worlds*...

Human Intelligence is Multimodal

Consider how you understand the world:

- You see a cat $\rightarrow$ You think "that's a cat"
- Someone says "look at that fluffy cat" $\rightarrow$ You know what to look for
- You read "a red apple on a wooden table" $\rightarrow$ You can visualize it

Photo of a cat

Your brain seamlessly integrates:

- Visual perception
- Language understanding
- Semantic knowledge

Red apple on wooden table

Can AI do the same?

What is Multimodal AI?

Definition: A multimodal model processes and/or generates content across multiple data modalities (text, images, audio, video) to perform a task

Key Difference from Single-Modal:

- **Before:** Separate CNNs for images, separate Transformers for text
- **Multimodal:** Learn shared structure and alignment between modalities

Learning Analogy (VARK):

Just as humans learn better with Visual, Auditory, Reading/Writing, and Kinesthetic cues, combining modalities improves:

- Robustness
- Sample efficiency
- Generalization

The Core Challenge

Images and text live in completely different spaces:

- **Image:** $512 \times 512 \times 3$ tensor of pixel values
- **Text:** Sequence of discrete tokens

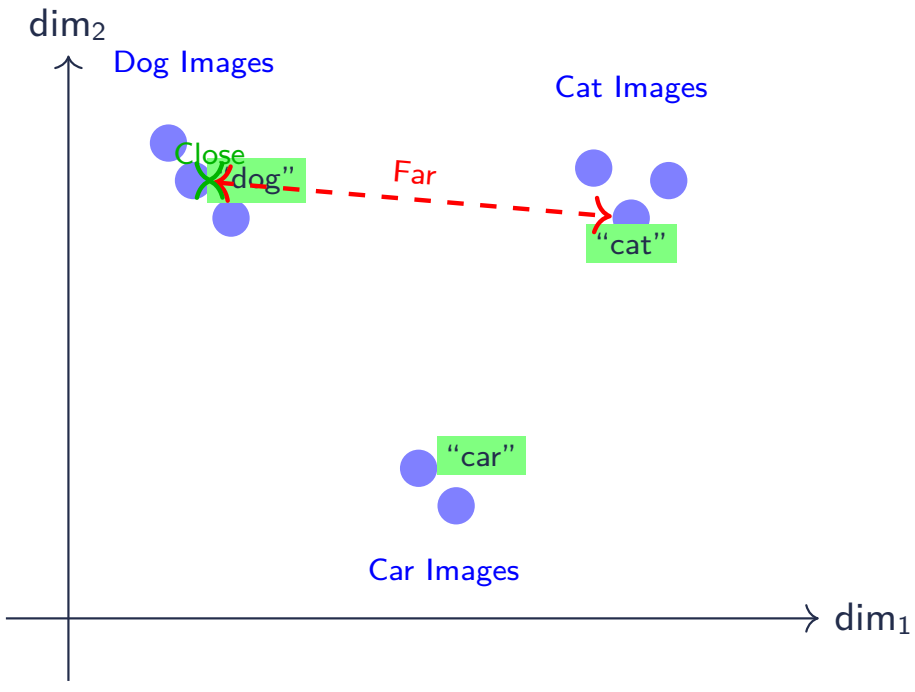
Fundamental Questions:

1. How do we represent both in a common space?
2. How do we learn meaningful correspondences?
3. How do we leverage this for downstream tasks?

The Answer: Joint Embeddings

The Core Idea: Joint Embedding Space

Key Insight: Map different modalities into a shared embedding space where semantically similar concepts are close together



This enables:

- Search images using text queries
- Generate captions for images
- Zero-shot classification

How to Learn Joint Embeddings?

The Challenge: No explicit supervision saying "this image region corresponds to this word"

We only have:

- Images
- Captions/descriptions that *somehow* describe the image
- No pixel-level or region-level annotations

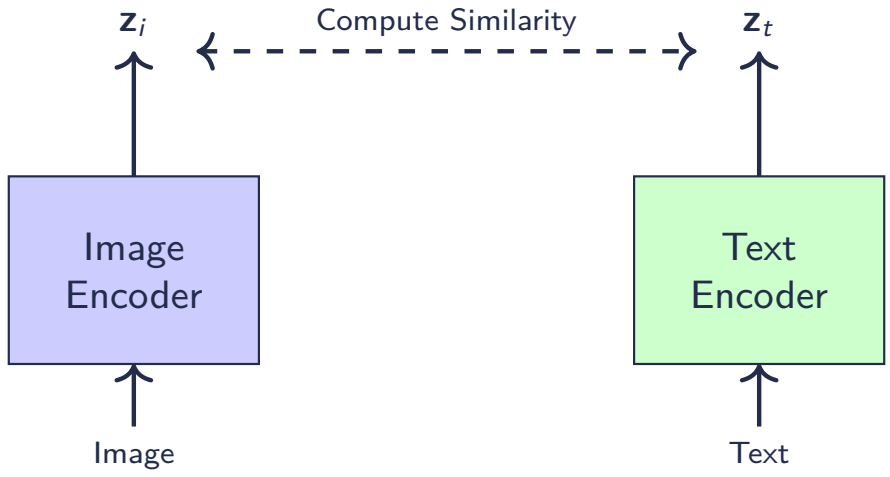
The Solution: Contrastive Learning

"Pull similar things together,
push dissimilar things apart"

Contrastive Learning Framework

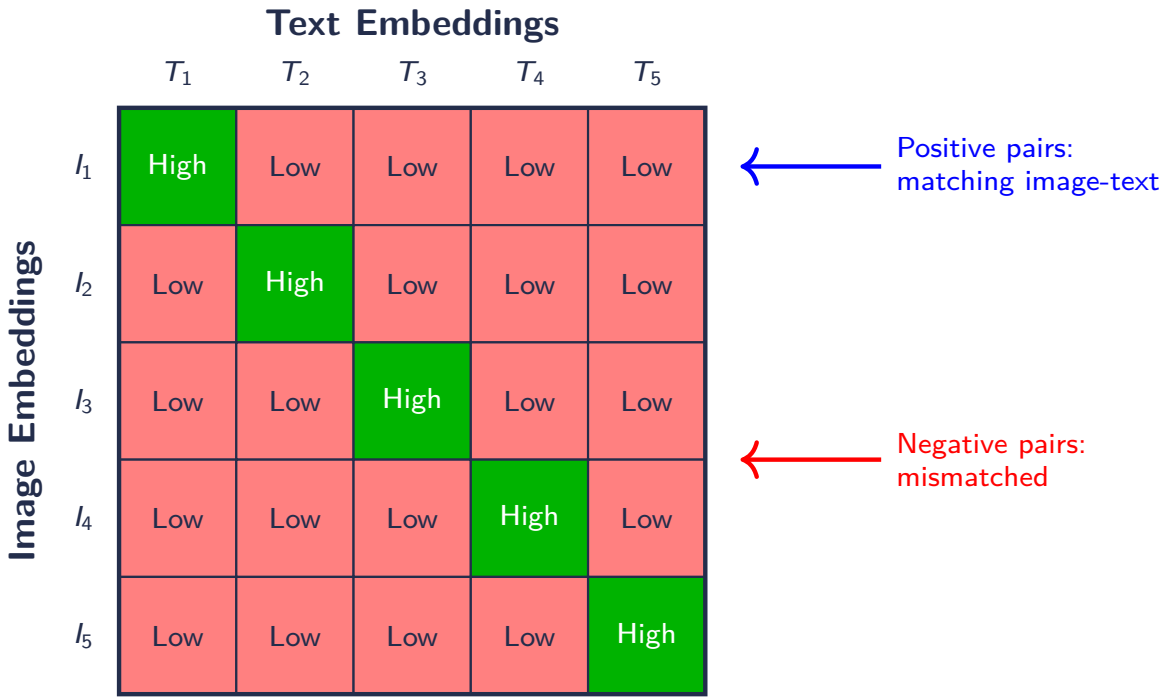
Training Signal:

- Maximize similarity for matching pairs
- Minimize similarity for non-matching pairs



Visualizing the Contrastive Batch

In a batch of N pairs, we compute all pairwise similarities:



Goal: Maximize the diagonal, minimize the off-diagonal

- The InfoNCE loss treats each row as a multi-class classification
- This explains why PyTorch uses `F.cross_entropy`!
- Batch size matters: $N = 1024 \rightarrow 1023$ negatives per positive

The Contrastive Loss (InfoNCE)

Given a batch of N image-text pairs:

For each image i, we have:

- One positive text t^+ (the correct caption)
- $N - 1$ negative texts (captions for other images)

Loss Function:

$$\mathcal{L}_{\text{contrastive}} = -\log \frac{\exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_{t^+})/\tau)}{\sum_{j=1}^{N} \exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_{t_j})/\tau)}$$

where:

- $\mathbf{z}_i$ is the image embedding
- $\mathbf{z}_{t^+}$ is the matching text embedding
- $\text{sim}(\cdot, \cdot) = \frac{\mathbf{z}_i^\top \mathbf{z}_t}{\|\mathbf{z}_i\| \|\mathbf{z}_t\|}$ (cosine similarity)
- τ is temperature parameter (learnable)

Understanding the Loss

This is a softmax over similarity scores!

$$\mathcal{L}_{\text{contrastive}} = -\log \frac{\exp(s^+/\tau)}{\sum_{j=1}^{N} \exp(s_j/\tau)}$$

Intuition:

- Encourage high probability for correct match
- Treat all negatives in batch as contrastive examples
- Temperature τ controls the "peakiness" of the distribution
  - Small τ: sharp, peaked distribution (hard negatives matter more)
  - Large τ: smooth distribution (easier training)

Symmetric Loss:

In practice, we also compute the reverse (text-to-image) loss:

$$\mathcal{L}_{\text{total}} = \frac{1}{2}(\mathcal{L}_{i \to t} + \mathcal{L}_{t \to i})$$

Why Contrastive Learning Works

Key Properties:

1. **Self-Supervised:** No manual annotations needed
   - Just image-text pairs from the internet!
   - Alt-text, captions naturally occur with images
2. **Scalable:** Can use massive datasets
   - Billions of image-text pairs available online
   - More data $\rightarrow$ better representations
3. **Flexible:** Learn general-purpose representations
   - Not tied to specific classification tasks
   - Transfers to many downstream applications
4. **Efficient:** Batch provides many negatives
   - Batch size 1024 $\rightarrow$ 1023 negatives per positive!
   - No need to explicitly mine hard negatives

Seminal paper: CLIP: Contrastive Language-Image Pre-training

CLIP (Radford et al., 2021) demonstrated the power of contrastive learning at scale

Key Innovation: Train on 400 million (image, text) pairs from the internet

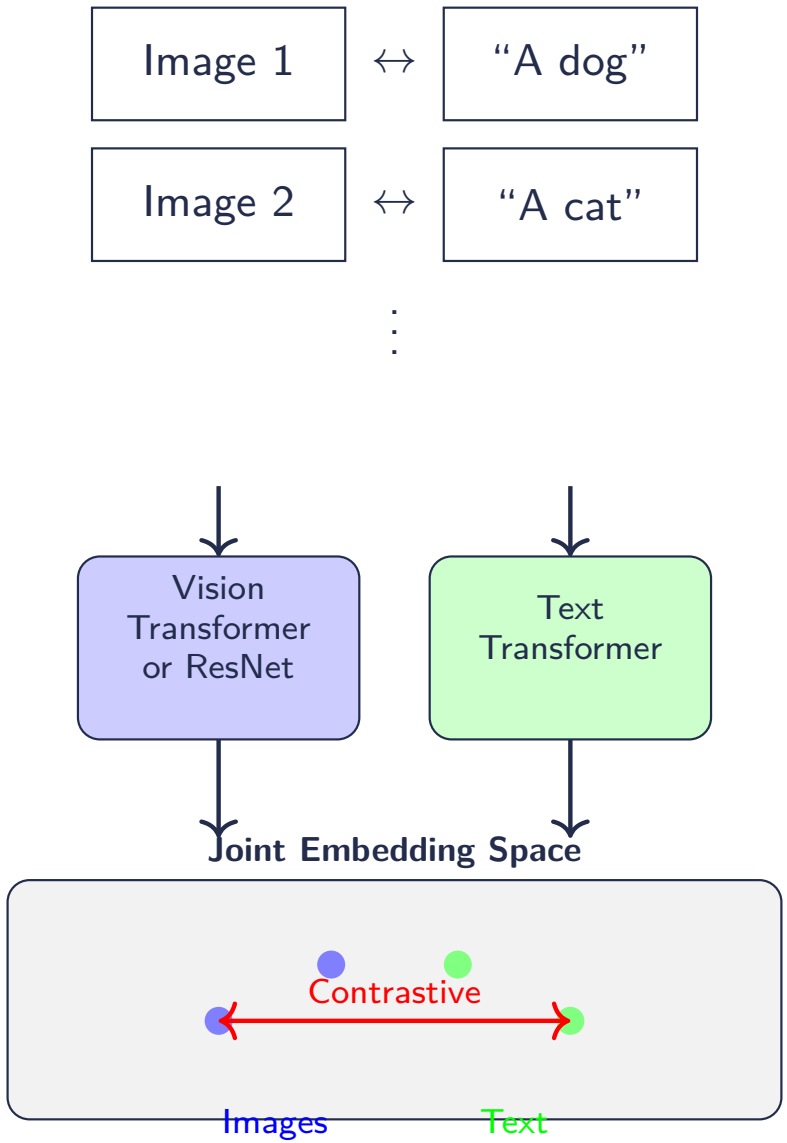
Training Data:

- Scraped from the web
- Alt-text from images
- Captions from various sources
- Naturally noisy but incredibly diverse

No carefully curated dataset needed!

This is the key to scale—use the internet itself as the dataset

CLIP Architecture



CLIP Training Process

Architecture Components:

- **Image Encoder:** Vision Transformer (ViT-B/32, ViT-L/14) or ResNet-50
  - Output: 512-dim or 768-dim embedding
- **Text Encoder:** Transformer (similar to GPT-2)
  - 12 layers, 8 attention heads
  - Output: 512-dim or 768-dim embedding (same as image)
- **Training:** Contrastive loss on batches of 32,768 pairs!
  - Large batch size crucial for performance
  - Each example has 32,767 negatives

Result: Joint embedding space where semantically related images and texts are close

CLIP in 6 Lines (PyTorch)

Simplified Implementation:

```
# z_i, z_t: (N, d) L2-normalized image/text embeddings
# tau is learnable temperature parameter
logits = (z_i @ z_t.t()) / tau
labels = torch.arange(len(z_i), device=logits.device)
loss = 0.5 * (F.cross_entropy(logits, labels)
            + F.cross_entropy(logits.t(), labels))
loss.backward()
optimizer.step()
optimizer.zero_grad()
```

That's it! The simplicity is deceptive—the power comes from:

- Scale (400M pairs)
- Large batch size (32K)
- Good encoders (ViT, Transformer)

Zero-Shot Classification with CLIP

Classify images into categories never explicitly seen!

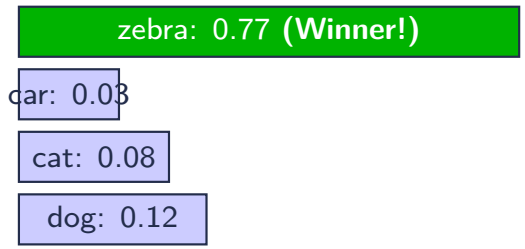
Input Image:

Photo of zebra

Text Prompts:

- "A photo of a dog"
- "A photo of a cat"
- "A photo of a car"
- "A photo of a zebra"

Similarity Scores:



Prediction: Zebra

Even though CLIP never saw "zebra" labels during training, it can match the visual concept to the text!

Zero-Shot Procedure: Step by Step

For N classes, here's how it works:

1. **Encode the image:**

$$\mathbf{z}_{\text{img}} = \text{ImageEncoder}(\text{image}) \in \mathbb{R}^d$$

2. **Create text prompts:** "A photo of a [class]"
   - E.g., "A photo of a dog", "A photo of a cat", etc.
3. **Encode each text prompt:**

$$\mathbf{z}_{\text{txt}}^{(c)} = \text{TextEncoder}(\text{prompt}_c) \in \mathbb{R}^d$$

4. **Compute cosine similarities:**

$$s_c = \frac{\mathbf{z}_{\text{img}}^\top \mathbf{z}_{\text{txt}}^{(c)}}{\|\mathbf{z}_{\text{img}}\| \|\mathbf{z}_{\text{txt}}^{(c)}\|}$$

5. **Predict the class with highest similarity:**

$$\hat{y} = \arg\max_c s_c$$

Why Zero-Shot Works

Traditional supervised learning:

- Train on 1000 ImageNet classes
- Can only predict those exact 1000 classes
- To add new class: need labeled data + retrain

CLIP's zero-shot:

- Learns general visual concepts ("furry", "four legs", "pointed ears")
- Learns language understanding ("cat", "feline", "tabby")
- At test time: matches image concepts to text descriptions
- Works for classes never seen during training!

Example:

- CLIP never trained on "traffic cone" images
- But knows what "traffic" and "cone" look like
- Can recognize traffic cones by matching to text "traffic cone"

CLIP Performance

Zero-shot CLIP matches or exceeds supervised models:

Dataset	ResNet-50 (supervised)	CLIP (zero-shot)
ImageNet	76.5%	76.2%
ImageNet-V2	67.8%	70.1%
ObjectNet	47.3%	52.5%

Key Observations:

- Matches supervised on ImageNet (trained on!)
- Better generalization to new distributions (ImageNet-V2, ObjectNet)
- More robust to distribution shift
- No task-specific training needed!

Why CLIP Works So Well

Four Key Factors:

1. **Scale:** 400M image-text pairs $\gg$ any supervised dataset
   - ImageNet has 1.2M images
   - CLIP: 333$\times$ more data
2. **Natural Supervision:** Alt-text provides weak but diverse labels
   - Rich, natural language descriptions
   - Much more information than single class label
3. **Flexibility:** Text encoder can handle any description
   - Not limited to predefined categories
   - Can specify fine-grained distinctions
4. **Transfer:** Joint space enables zero-shot and few-shot learning
   - No retraining needed for new tasks
   - Just change the text prompts!

Beyond CLIP: Vision-Language Models

CLIP opened the door to a new generation of models

Vision-Language Models (VLMs): Models that can both understand and generate language about images

Examples:

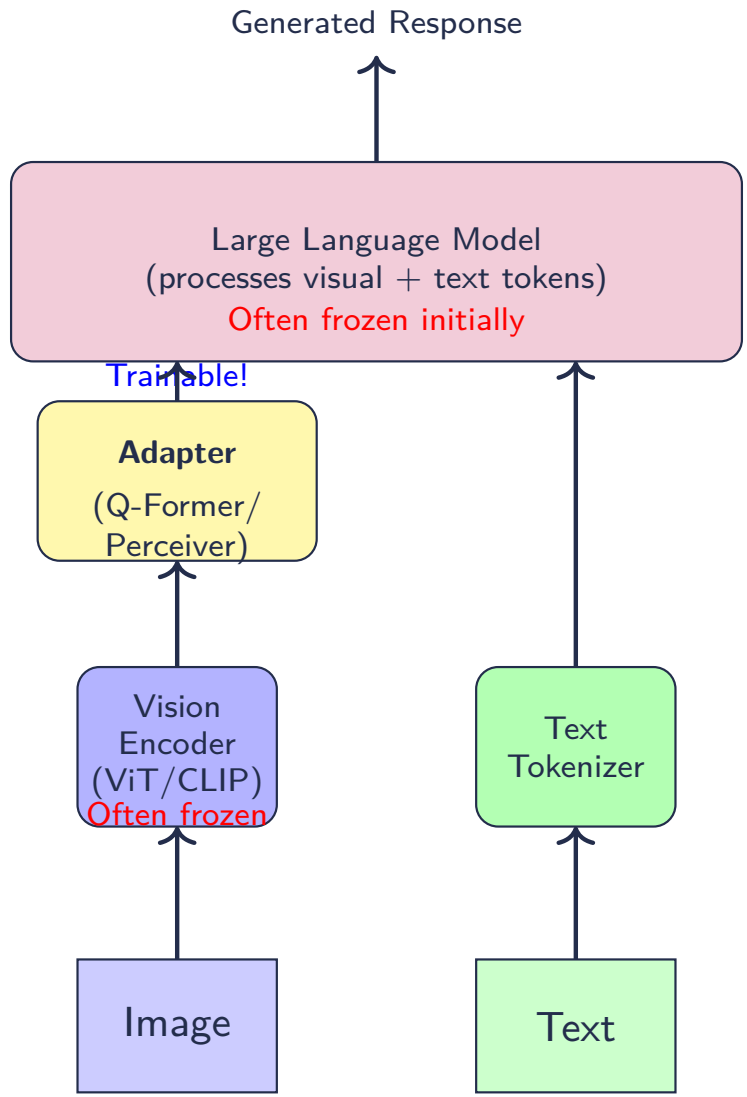
- **Flamingo** (DeepMind): Few-shot learning for vision-language tasks
- **BLIP/BLIP-2:** Unified vision-language understanding and generation
- **GPT-4V:** GPT-4 with vision capabilities
- **Gemini:** Google's multimodal model

These go beyond classification:

- Detailed image captioning
- Visual question answering (VQA)
- Visual reasoning
- Reading text in images (OCR)

Typical VLM Architecture

Common Pattern: Perception → Adapter → Language



The Adapter: Bridging Vision and Language

Why do we need an adapter?

- **Problem:** ViT outputs 196+ patch tokens (14×14 grid)
  - Too many tokens for LLM efficiency
  - Different "language" than text tokens
- **Solution:** Adapter compresses and translates
  - Reduces 196 tokens → 32-64 "visual tokens"
  - Maps visual features to LLM's input space
  - Acts as a learned compression + translation layer

Common Adapter Architectures:

- **Linear Projection:** Simple learned matrix (early VLMs)
- **Q-Former (BLIP-2):** Query-based cross-attention
  - 32 learnable query tokens attend to all image patches
  - Queries become the compressed visual representation
- **Perceiver Resampler (Flamingo):** Similar to Q-Former

Training Strategy: Freeze and Focus

Efficient VLM Training:

Stage 1: Adapter Pre-training

- Freeze both Vision Encoder and LLM
- Train only the adapter (1-2% of total params)
- Objective: Align visual and text representations
- Dataset: Large image-caption pairs (millions)

Stage 2: Instruction Fine-tuning (Optional)

- Freeze Vision Encoder
- Train Adapter + LLM (or just adapter)
- Objective: Follow instructions, VQA, detailed captioning
- Dataset: High-quality instruction-following data

Why this works:

- Leverages strong pre-trained components
- Much cheaper than training from scratch
- Typical: Train 10M params instead of 7B+ params

Training Paradigms for VLMs

Common Training Strategies:

1. **Contrastive (ITC):** Image-Text Contrastive (like CLIP)
   - Align global image and text representations
2. **Matching (ITM):** Image-Text Matching
   - Binary classification: does this text describe this image?
   - Finer-grained than contrastive loss
3. **Generative (Captioning):** Language Modeling
   - Given image, generate caption autoregressively
   - Standard next-token prediction loss
4. **Instruction Following:** Task-specific fine-tuning
   - VQA: answer questions about images
   - Instruction following: "describe the objects in this image"

Most VLMs use a "training cocktail" combining multiple objectives

Evaluation: How Good are VLMs?

Standard Benchmarks:

Task	Metric
Image-Text Retrieval	Recall@K (R@1, R@5, R@10)
Image Captioning	CIDEr, BLEU, METEOR
VQA	Accuracy on question-answer pairs
Visual Reasoning	Compositional understanding (GQA, CLEVR)

Recent Focus: Compositional Understanding

- Can model understand "a red car next to a blue house"?
- Not just objects, but relationships and attributes
- Challenging—requires fine-grained grounding

What Can Modern VLMs Do?

Example 1: Detailed Visual Understanding

Complex outdoor scene

Model Output:

"The image shows a grey tabby cat with green eyes sitting on a weathered wooden fence. The fence appears to be made of cedar planks. In the background, there's a garden with blooming roses and what looks like a shed. The lighting suggests it's golden hour, with warm sunlight casting long shadows."

Beyond simple labels:

- Fine-grained attributes ("grey tabby", "green eyes")
- Material recognition ("weathered wooden", "cedar planks")
- Spatial reasoning ("in the background")

What Can Modern VLMs Do?

Example 2: Meme Understanding

Distracted boyfriend meme

User: "Explain this meme"

Model:

"This is the 'Distracted Boyfriend' meme. It shows a man walking with his girlfriend but turning to look at another woman. The girlfriend looks upset. The meme is used to represent being distracted by something new/attractive while neglecting something you already have. The format typically labels the three people to represent different concepts in a humorous comparison."

Capabilities demonstrated:

- Cultural knowledge (recognizing meme format)
- Emotional recognition (girlfriend is upset)
- Abstract reasoning (metaphorical meaning)

What Can Modern VLMs Do?

Additional Capabilities:

- **OCR + Understanding:** Read text and understand context
  - "What does this sign say and what does it mean?"
- **Spatial Reasoning:** "The ball is to the left of the box"
  - Understand 3D layouts from 2D images
- **Counting:** "How many objects are in this image?"
  - Still challenging but improving rapidly
- **Visual Chain-of-Thought:** Multi-step reasoning
  - "First I see the car is damaged. The damage appears on the driver's side. Based on the angle, it was likely a side-impact collision..."
- **Multimodal Dialogue:** Back-and-forth conversation
  - "Can you describe the person on the left? Now compare them to the person on the right."

From classification to genuine visual understanding!

Real-World Applications

Multimodal AI is already deployed:

- **Healthcare:**
  - Radiology reports + medical images for diagnosis
  - Clinical notes + scans for treatment planning
- **E-commerce:**
  - Visual search: upload image, find similar products
  - Automatic product tagging from images
  - Price prediction from images + descriptions
- **Accessibility:**
  - Image descriptions for visually impaired users
  - Real-time scene understanding for navigation
- **Content Moderation:**
  - Understanding context (image + text) for harmful content
  - Better than either modality alone

Scientific Applications

Multimodal models in research:

- **Microscopy:** Cell images + gene expression data
- **Astronomy:** Telescope images + spectral data
- **Climate Science:** Satellite imagery + meteorological data
- **Materials Science:** Microscope images + chemical composition
- **Robotics:** Visual perception + natural language commands

The Pattern:

Most real-world problems involve multiple modalities!

Unimodal models leave information on the table

Challenges and Limitations

1. Counting is Hard

Model: "The person is showing 6 fingers"
Actual: 5 fingers

2. Fine-grained Spatial

User: "Is the red cube on the blue cube?"
Model struggles with precise spatial relationships

3. Compositionality

Distinguishing "red apple + green banana" from "green apple + red banana" is still challenging

4. Small Text Recognition

Models struggle with fine-grained OCR (Optical Character Recognition)

Challenges: Beyond Technical

Data and Bias Issues:

1. **Data Quality:**
   - Web-scraped captions are often noisy or incorrect
   - Many captions incomplete ("a dog" vs detailed description)
   - Harmful/toxic content in training data
2. **Bias and Fairness:**
   - Training data biases get amplified
   - Stereotypical associations (gender, race, occupation)
   - Example: "CEO" query $\rightarrow$ predominantly male images
   - Western-centric representations dominate
3. **Robustness:**
   - Models can be fooled by adversarial patches
   - Performance degrades on out-of-distribution images
   - Sensitive to image quality and resolution

Ethical Considerations

Key Concerns:

- **Privacy:**
  - Models trained on billions of images from the web
  - Many contain people without consent
  - Can models "remember" training data?
- **Copyright:**
  - Training on copyrighted images and text
  - Who owns the outputs?
  - Legal frameworks still evolving
- **Dual Use:**
  - Surveillance applications
  - Deepfake creation
  - Misinformation at scale
- **Bias Amplification:**
  - Models perpetuate societal biases
  - Example: "CEO" $\rightarrow$ mostly male
  - Can reinforce harmful stereotypes

What We've Learned: Multimodality

Core Concepts:

1. **Joint Embeddings:** Map different modalities to shared space
   - Enables cross-modal understanding
2. **Contrastive Learning:** Self-supervised training
   - Pull similar pairs together, push apart dissimilar
   - InfoNCE loss: softmax over similarities
3. **CLIP:** Breakthrough at scale
   - 400M image-text pairs
   - Zero-shot classification
   - Strong transfer learning
4. **VLMs:** Beyond classification
   - Captioning, VQA, reasoning
   - Combining vision encoder + LLM

Key Takeaways

1. **Multimodal AI mimics human cognition**
   - We naturally integrate visual and linguistic information
   - Combining modalities improves robustness and generalization
2. **Contrastive learning is powerful and scalable**
   - Self-supervised training on massive datasets
   - No need for expensive manual annotations
3. **Zero-shot capabilities are transformative**
   - Models can handle tasks they weren't explicitly trained for
   - More flexible and adaptable than supervised models
4. **The future is multimodal**
   - Most real-world problems involve multiple modalities
   - Single-modality models are increasingly limiting

The Multimodal Future

Emerging Trends:

- **More Modalities:** Vision + Language + Audio + Video
- **Unified Architectures:** Single model for all modalities
- **Embodied AI:** Robots that see, hear, and understand language
- **Interactive Systems:** Back-and-forth visual dialogue

But alongside multimodality,
another revolution is happening:

Generative AI

Next: Lecture 12.2 - Generative AI and Diffusion Models