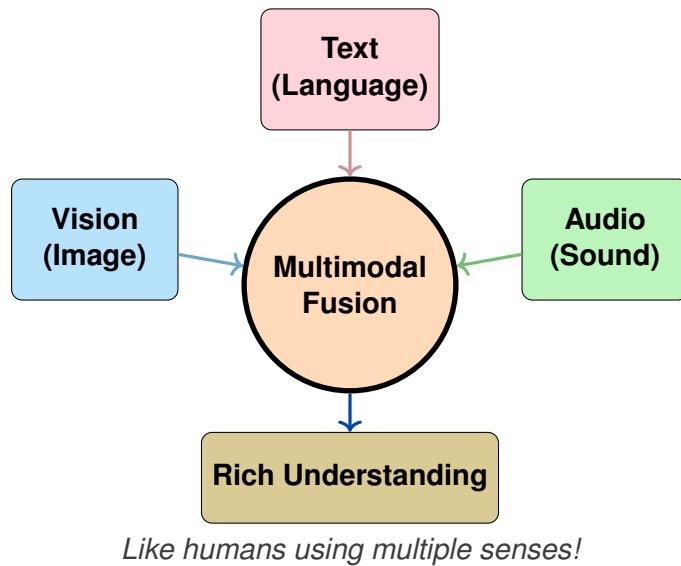


Deep Learning for Perception

Lecture 10: Multimodal Learning & Vision-Language Models



Topics Covered:

- What is Multimodality?
- Why Multimodal Learning?
- Vision-Language Models (VLMs)
- VLM Architectures
- Multimodal Tasks Overview
- Visual Question Answering
- Image Captioning
- Image-Text Retrieval
- Visual Grounding
- CLIP: Contrastive Learning
- Zero-Shot Classification
- Text-to-Image Generation

Contents

1 A Multimodal World	3
1.1 What is a Modality?	3
1.2 Why Multimodality Matters	3
1.3 Key Challenges in Multimodal Learning	5
2 Introduction to Vision-Language Models (VLMs)	6
2.1 What is a Vision-Language Model?	6
2.2 VLM Architecture: The Three Components	6
2.3 VLM Learning Strategies	8
3 Multimodal Tasks and Models	9
3.1 Visual Question Answering (VQA)	9
3.2 Document Visual QA (DocVQA)	10
3.3 Image Captioning	11
3.4 Image-Text Retrieval	11
3.5 Visual Grounding	12
4 CLIP and Relatives: Contrastive Vision-Language Learning	13
4.1 The CLIP Approach	13
4.2 CLIP Architecture	14
4.3 Contrastive Learning Objective	14
4.4 Zero-Shot Classification with CLIP	15
4.5 CLIP Relatives and Text-to-Image	15
5 Summary	16
6 Glossary	17

Advance Organizer — What You'll Learn

The Big Picture: Combining What You've Learned

You've mastered individual modalities:

- **CNNs** — Process images (spatial data)
- **RNNs/LSTMs** — Process sequences (text, audio)
- **Transformers/Attention** — Handle long-range dependencies
- **Autoencoders/VAEs/GANs** — Generate new data

Now: We combine these to build models that understand **multiple modalities together**—just like humans use sight, hearing, and language simultaneously!

Learning Objectives:

1. **Explain** what multimodality means and why it matters
2. **Describe** Vision-Language Model (VLM) architectures
3. **Identify** key multimodal tasks (VQA, captioning, retrieval, grounding)
4. **Understand** CLIP's contrastive learning approach
5. **Apply** concepts to real-world multimodal applications

Real-World Impact: These models power ChatGPT-4V, Google Lens, AI image search, DALL-E, and modern AI assistants!

1 A Multimodal World

Why It Matters

Humans don't understand the world through just one sense. When you watch a movie, you see images, hear dialogue and music, and read subtitles—all simultaneously. This combination gives you **richer understanding** than any single source alone. AI should work the same way!

1.1 What is a Modality?

Definition

A **modality** is a medium, channel, or way of representing/inputting information.

Common modalities in machine learning:

- **Vision:** Images, videos, 3D scans
- **Text:** Natural language, documents, code
- **Audio:** Speech, music, environmental sounds
- **Sensor data:** LiDAR, radar, accelerometer, EEG

Key Terminology:

- **Unimodal:** Uses only ONE modality (e.g., image-only classifier)
- **Multimodal:** Combines TWO OR MORE modalities (e.g., image + text)

Analogy

The Five Senses Analogy

Humans have 5 senses: sight, hearing, touch, smell, taste. We constantly **combine** them:

Scenario: You're in a dark room and hear a sound.

- **Audio only:** "Something moved" (ambiguous—could be a cat, wind, intruder)
- **Audio + Vision:** You turn on the light and see an open window
- **Combined understanding:** "The wind blew through the open window"

Key insight: Adding another modality **resolved ambiguity** and gave richer understanding!

1.2 Why Multimodality Matters

The 7-38-55 Rule of Human Communication

Research by Albert Mehrabian found that in face-to-face communication:

- 7% of meaning comes from **words** (text)
- 38% comes from **tone of voice** (audio)
- 55% comes from **body language/facial expressions** (vision)

Implication: A text-only AI misses 93% of human communication signals! This is why we need multimodal models.

Real-World Data is Multimodal

Most real-world data naturally involves multiple modalities:

Data Source	Modalities Involved
YouTube video	Vision + Audio + Text (captions)
Medical record	Images (X-ray) + Text (doctor's notes)
Social media post	Image + Text + (sometimes video/audio)
Autonomous car	Vision + LiDAR + Radar + GPS
Document scan	Layout + Text + Images/diagrams

Problem: Unimodal models can only process ONE part of this data!

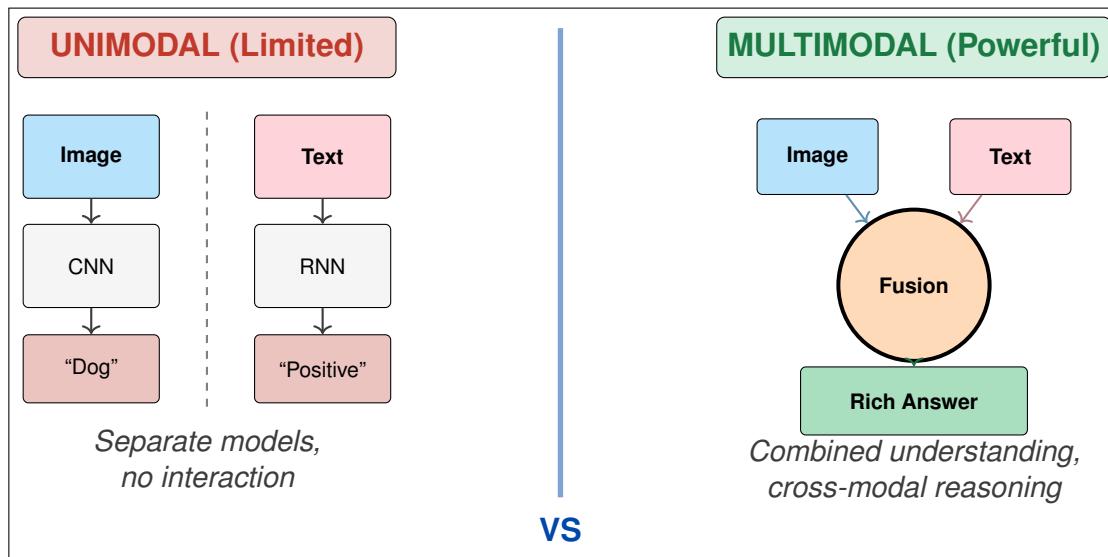


Figure 1: Comparison of Unimodal vs Multimodal approaches. Unimodal processes each modality separately with no interaction. Multimodal combines modalities through fusion for richer understanding.

1.3 Key Challenges in Multimodal Learning

Five Core Challenges

- 1. Representation:** How to encode different modalities into compatible formats?
 - Images: 2D pixel grids → need CNN/ViT
 - Text: Sequences of tokens → need RNN/Transformer
 - Must output embeddings in same vector space!
- 2. Alignment:** How to match elements across modalities?
 - Which image region corresponds to which word?
 - Example: “red car” ↔ specific pixels showing red car
- 3. Modality Dominance:** One modality may overpower others
 - Text often dominates vision in training
 - Model might ignore visual information!
- 4. Missing Modalities:** What if some data lacks one modality?
 - Not all images have captions
 - Model must handle incomplete inputs
- 5. Scale Mismatch:** Different modalities have different data sizes
 - Text: Trillions of tokens available (web text)
 - Images: Fewer high-quality labeled images

2 Introduction to Vision-Language Models (VLMs)

Connection to Prior Learning

Building on Your Knowledge:

What you learned:

- CNNs extract features from images → Image Encoder
- Transformers/RNNs process text → Text Encoder
- Attention aligns different parts of input → Cross-modal Attention
- Encoder-Decoder architecture → VLM structure

VLMs combine all of these! They use image encoders, text encoders, and attention mechanisms to jointly understand vision and language.

2.1 What is a Vision-Language Model?

Definition

A **Vision-Language Model (VLM)** is a neural network that jointly processes **image** and **text** inputs to perform tasks requiring understanding of both modalities.

Key Capabilities:

- Understand images in the context of text
- Generate text descriptions of images
- Answer questions about images
- Match images with relevant text (and vice versa)

Example Models: CLIP, BLIP, ViLT, Flamingo, GPT-4V, LLaVA

2.2 VLM Architecture: The Three Components

Every VLM has three fundamental components:

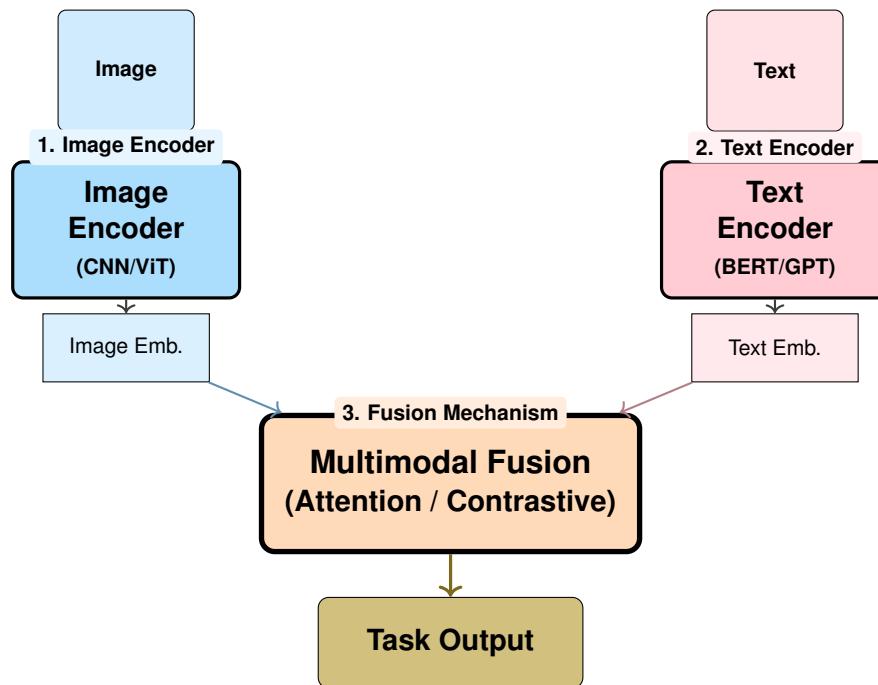


Figure 2: VLM Architecture: Image Encoder + Text Encoder + Fusion Mechanism. Each component plays a specific role in processing multimodal data.

Component 1: Image Encoder

Purpose: Convert raw pixels into meaningful feature vectors

Common architectures:

- **CNN-based:** ResNet, EfficientNet (what you learned in Weeks 6-9)
- **Transformer-based:** Vision Transformer (ViT) — treats image patches as tokens

Output: Image embedding vector (e.g., 512 or 768 dimensions)

Component 2: Text Encoder

Purpose: Convert text into meaningful feature vectors

Common architectures:

- **Transformer encoders:** BERT, RoBERTa
- **Transformer decoders:** GPT-2, GPT-3

Output: Text embedding vector (same dimension as image embedding)

Component 3: Fusion Mechanism

Purpose: Combine image and text embeddings for joint understanding

Approaches:

- **Concatenation:** Simply join embeddings $[e_{img}; e_{txt}]$
- **Cross-attention:** Let text “attend to” image features (and vice versa)
- **Contrastive alignment:** Map both to shared space (CLIP approach)

2.3 VLM Learning Strategies

Four Major Approaches to Building VLMs

1. Embedding Fusion (Image as Tokens)

- Split image into patches → treat as “visual tokens”
- Feed visual + text tokens into single Transformer
- Examples: VisualBERT, SimVLM

2. Image Embedding as Prefix

- Keep language model frozen
- Learn to convert image embedding to format LM understands
- Examples: Frozen, ClipCap

3. Cross-Attention Fusion

- Add special attention layers
- Language model attends to visual features
- Examples: VisualGPT, Flamingo

4. Contrastive Learning

- Map image and text to shared embedding space
- Train with contrastive loss (match pairs, push non-matches)
- Examples: CLIP, ALIGN

3 Multimodal Tasks and Models

Why It Matters

Now that you understand VLM architecture, let's explore what these models can do. Each task requires different kinds of reasoning about vision and language together.

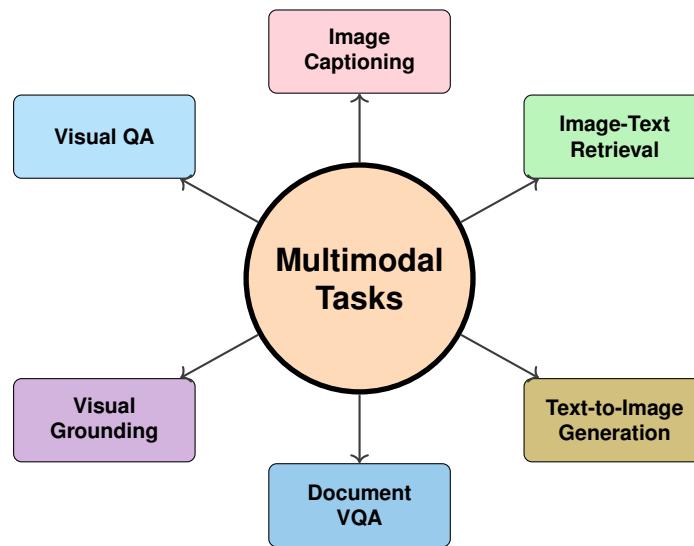


Figure 3: Overview of key multimodal tasks combining vision and language

3.1 Visual Question Answering (VQA)

Definition

Visual Question Answering (VQA) is the task of answering natural language questions about an image.

Input: Image + Question (text)

Output: Answer (text or classification)

Example:

- Image: Photo of a kitchen with a red kettle
- Question: “What color is the kettle?”
- Answer: “Red”

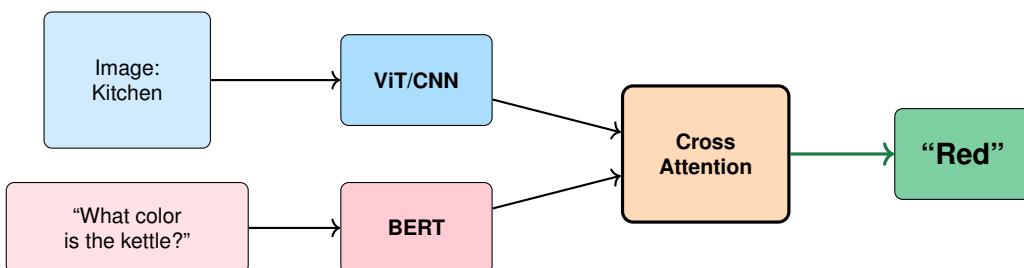


Figure 4: VQA pipeline: Image + Question → Encoders → Fusion → Answer

VQA Requires Multiple Skills

To answer visual questions, the model must master:

1. **Object recognition:** “What objects are in the image?”
2. **Attribute detection:** “What color/size/shape is it?”
3. **Spatial reasoning:** “What is to the left of X?”
4. **Counting:** “How many people are there?”
5. **Reading:** “What does the sign say?”
6. **Common sense:** “Is it raining?” (infer from umbrellas)

Popular models: BLIP-VQA, ViLT, mPLUG

3.2 Document Visual QA (DocVQA)

Definition

Document VQA answers questions about scanned/photographed documents by understanding both visual layout and text content.

Input: Document image + Question

Output: Answer extracted or inferred from document

Applications: Form processing, invoice analysis, contract review

DocVQA Models

LayoutLM: Uses OCR to extract text, then combines text + layout positions

Donut: OCR-free! Directly reads document images end-to-end

- Encoder: Swin Transformer (vision)
- Decoder: BART (text generation)

Nougat: Specialized for academic PDFs, outputs structured markdown

3.3 Image Captioning

Definition

Image Captioning generates a natural language description of an image.

Input: Image

Output: Descriptive text (caption)

Example:

- Image: Beach scene with people
- Caption: “A group of people playing volleyball on a sunny beach”

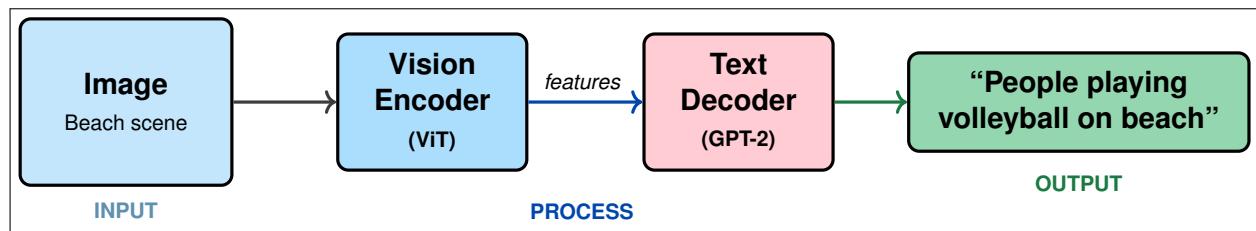


Figure 5: Image Captioning uses an Encoder-Decoder architecture. The Vision Encoder extracts image features, and the Text Decoder generates a natural language description.

Popular Captioning Models

ViT-GPT2: Vision Transformer encoder + GPT-2 decoder, fine-tuned on COCO

BLIP: Bootstraps training with noisy web captions, achieves strong generation

GIT: Generative Image-to-Text model from Microsoft

3.4 Image-Text Retrieval

Definition

Image-Text Retrieval finds matching content across modalities:

Text → Image: Given a text query, find relevant images

Image → Text: Given an image, find relevant captions/descriptions

Key approach: Create a **shared embedding space** where similar image-text pairs are close together.

Concrete Example

Retrieval in Action

Text → Image:

- Query: “A cat sleeping on a red couch”
- Result: Images of cats on red furniture ranked by similarity

Image → Text:

- Query: Upload photo of sunset over ocean
- Result: “Beautiful sunset over the Pacific Ocean”

This is how Google Image Search and Pinterest work!

3.5 Visual Grounding

Definition

Visual Grounding locates specific regions in an image based on a text description.

Input: Image + Referring expression (text phrase)

Output: Bounding box or segmentation mask

Example:

- Image: Room with multiple objects
- Text: “The red ball on the left”
- Output: Bounding box around the specific red ball

Models: OWL-ViT, Grounding DINO

4 CLIP and Relatives: Contrastive Vision-Language Learning

Why It Matters

CLIP (Contrastive Language-Image Pre-training) revolutionized vision-language AI by showing that models trained on massive web data can perform **zero-shot** tasks—without any task-specific training! It's the foundation for DALL-E, Stable Diffusion, and many modern VLMs.

4.1 The CLIP Approach

Definition

CLIP learns to map images and text into a **shared embedding space** where matching pairs are close and non-matching pairs are far apart.

Training data: 400 million (image, text) pairs from the internet

Key innovation: Instead of predicting specific labels, CLIP learns general visual-semantic alignment that transfers to any task!

Analogy

The Party Matching Game

Imagine a party where 100 people each have a photo and a name tag. Your goal is to match photos to name tags.

Traditional approach: Memorize exactly which face goes with which name. Only works for people you've seen!

CLIP approach: Learn what makes a face “look like” a certain name. Then you can match even **NEW** people you've never seen!

This is why CLIP works zero-shot: it learned **general correspondence**, not specific memorization.

4.2 CLIP Architecture

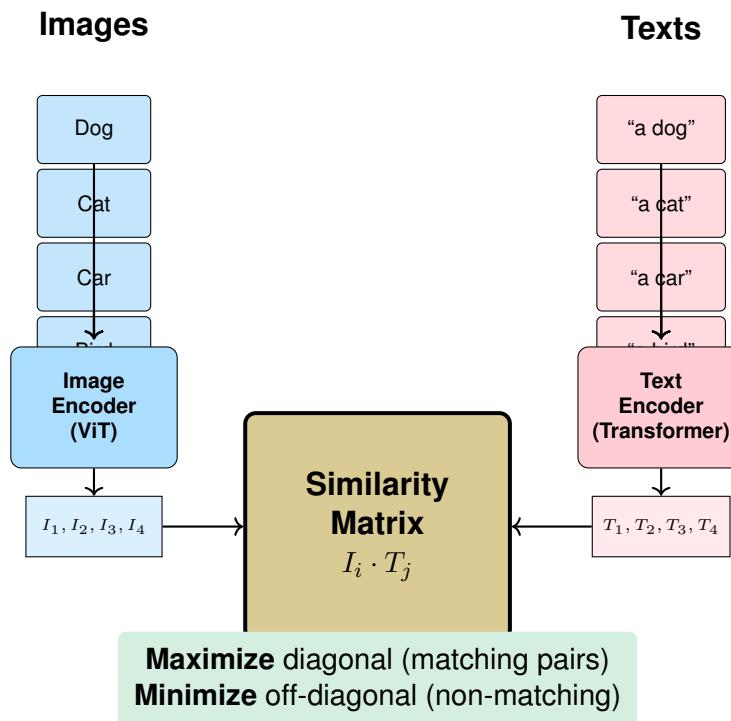


Figure 6: CLIP architecture: Dual encoders produce embeddings that are compared in a similarity matrix. Training pushes matching pairs together.

4.3 Contrastive Learning Objective

Key Formula

CLIP Contrastive Loss (InfoNCE)

For a batch of N image-text pairs:

Step 1: Compute embeddings

- Image embeddings: I_1, I_2, \dots, I_N
- Text embeddings: T_1, T_2, \dots, T_N

Step 2: Similarity matrix: $S_{ij} = I_i \cdot T_j$ (dot product)

Step 3: Contrastive loss

$$\mathcal{L}_{\text{image}} = -\frac{1}{N} \sum_{i=1}^N \log \frac{\exp(S_{ii}/\tau)}{\sum_{j=1}^N \exp(S_{ij}/\tau)}$$

where τ is a temperature parameter.

Intuition: Push matching pairs (diagonal) together, push non-matching pairs apart!

4.4 Zero-Shot Classification with CLIP

Definition

Zero-shot classification means classifying images into categories **without** training on those specific categories.

How CLIP does it:

1. Create text prompts: “A photo of a dog”, “A photo of a cat”, etc.
2. Encode the image and all text prompts
3. Find which text prompt is most similar to the image
4. That’s the predicted class!

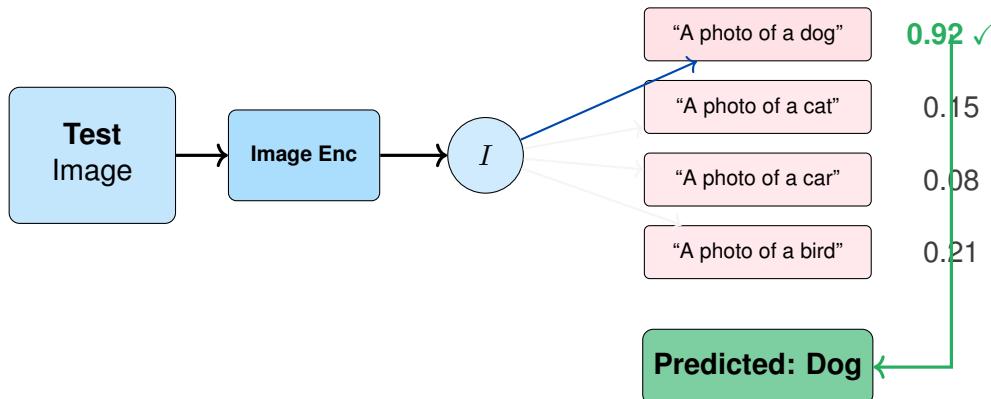


Figure 7: CLIP zero-shot classification: Compare image embedding to text prompt embeddings. Highest similarity wins.

4.5 CLIP Relatives and Text-to-Image

Models Building on CLIP

OpenCLIP: Open-source reproduction with various sizes

ALIGN: Google’s version trained on 1.8B noisy pairs

BLIP: Adds generation capability (can caption, not just match)

SigLIP: Improved training objective (sigmoid vs softmax)

Text-to-Image Generation using CLIP:

Stable Diffusion / DALL-E 2: Use CLIP text encoder

- Text query → CLIP text embedding
- Guide diffusion process with this embedding
- Generate image matching the text description

5 Summary

Key Takeaways

1. Multimodality

- Modality = way of representing information (vision, text, audio)
- Combining modalities gives richer, less ambiguous understanding
- Real-world data is inherently multimodal

2. Vision-Language Models (VLMs)

- Three components: Image Encoder + Text Encoder + Fusion
- Learning strategies: Embedding fusion, cross-attention, contrastive

3. Key Multimodal Tasks

- **VQA:** Answer questions about images
- **Captioning:** Generate text descriptions
- **Retrieval:** Find matching images/text
- **Grounding:** Locate regions from text

4. CLIP

- Contrastive learning on 400M image-text pairs
- Shared embedding space for vision and language
- Enables zero-shot classification
- Foundation for text-to-image generation

Self-Test

Q1: What is a modality? Give three examples.

A: A medium for representing information. Examples: vision, text, audio.

Q2: What are the three main components of a VLM?

A: Image encoder, Text encoder, Fusion mechanism.

Q3: How does CLIP enable zero-shot classification?

A: Compare image embedding to text prompt embeddings; highest similarity = predicted class.

Q4: What is the difference between VQA and Image Captioning?

A: VQA: image + question → answer; Captioning: image only → description.

Q5: What is contrastive learning?

A: Training that pushes matching pairs together and non-matching pairs apart.

6 Glossary

Term	Definition
Modality	A medium or channel for information (vision, text, audio)
Multimodal	Combining two or more modalities
VLM	Vision-Language Model—processes images and text jointly
VQA	Visual Question Answering—answer questions about images
Captioning	Generate text descriptions of images
Retrieval	Find matching content across modalities
Grounding	Locate image regions from text descriptions
CLIP	Contrastive Language-Image Pre-training
Zero-shot	Performing tasks without task-specific training
Contrastive	Learning by comparing positive and negative pairs
Embedding	Vector representation of data in a continuous space
Fusion	Combining information from multiple modalities