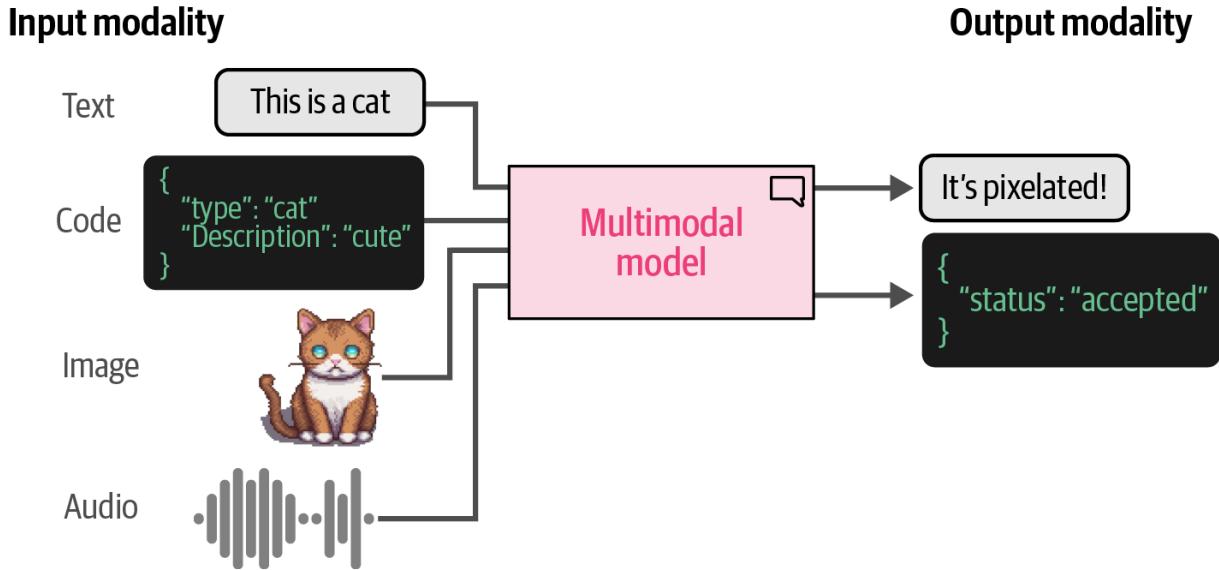


## Chap-9

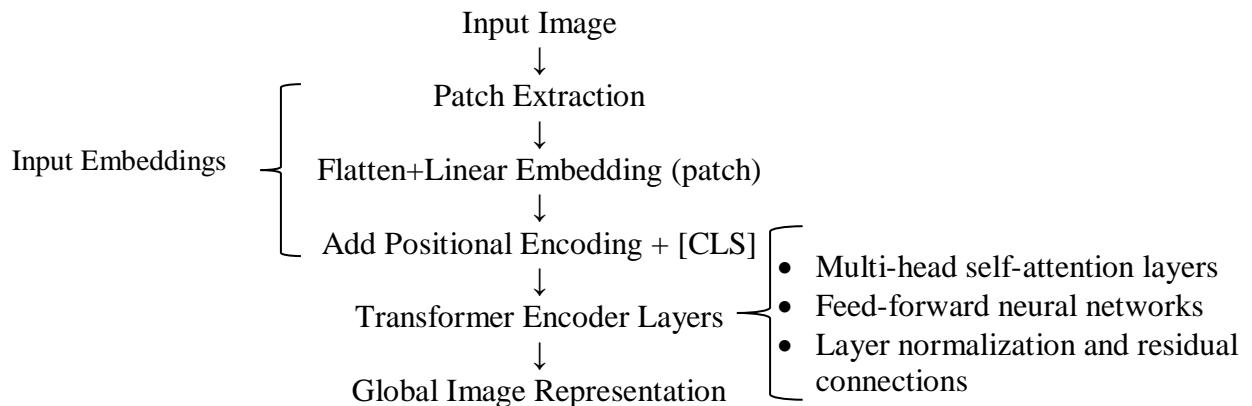
### Multimodal Large Language Models



## Transformers for Vision

**Objective:** How transformer architectures are adapted to handle visual information and align it with text representations.

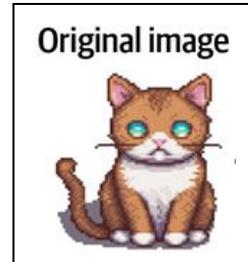
## Vision Transformer



## Representation: Step-by-Step Image Processing

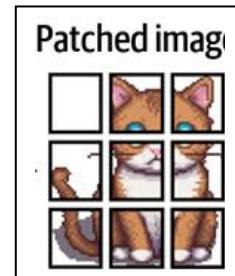
### Step 1: Input Image

An original image containing an object (e.g., a cat).



### Step 2: Patch Extraction

Each patch captures a **local visual region**

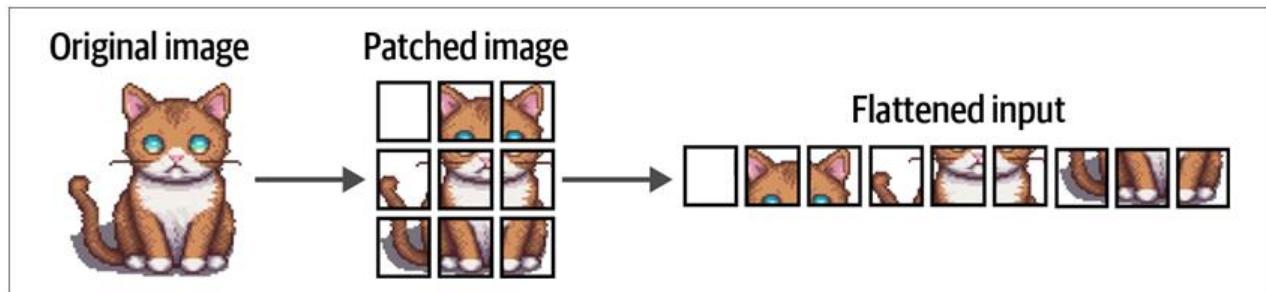


### Step 3: Patch Flattening and Linear Embedding

Each image patch is:

1. **Flattened** into a 1D vector
2. **Linearly projected** into a fixed-dimensional embedding space

This projection maps raw pixel values into **dense visual embeddings**, analogous to word embeddings in text models.



Each patch is converted into a numeric vector (embedding).

```
P1 → [0.21, 0.45, 0.13, ...]  
P2 → [0.67, 0.10, 0.89, ...]  
P3 → [0.34, 0.76, 0.22, ...]  
...
```

## **Step 4: Add Positional Information**

Since spatial order matters, positional embeddings are added.

## **Step 5: Transformer Processing (Global Attention)**

The sequence of patch embeddings is then passed through a **stack of transformer encoder layers**, each consisting of:

- Multi-head self-attention
- Feed-forward neural networks
- Residual connections and layer normalization

## **Step 6: Output Representation**

A global image representation is produced.

Example output:

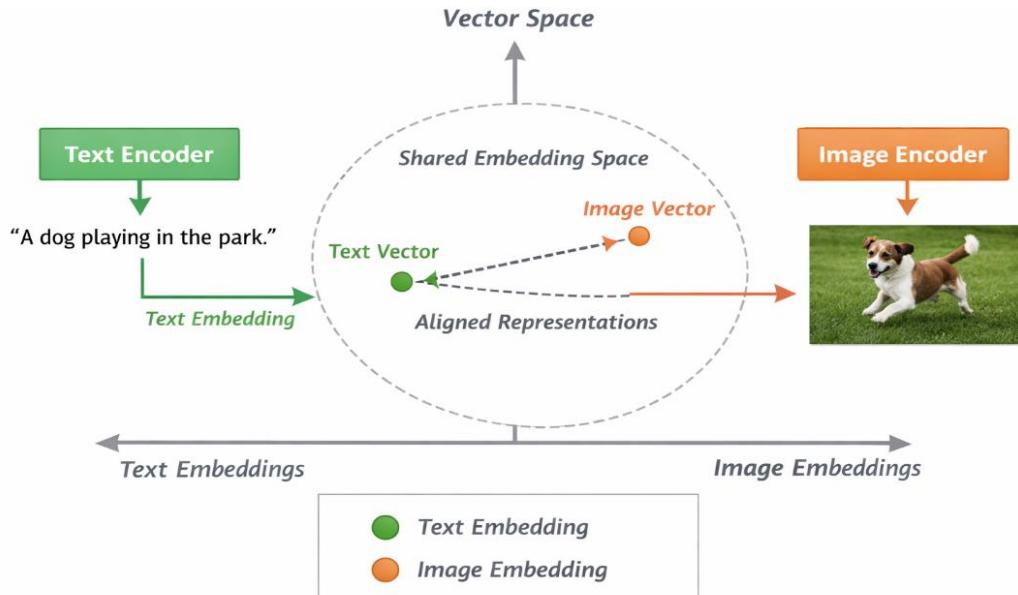
- **Classification:** “Cat”
- **Caption:** “A cat is sitting on the floor”

## **Multimodal Embedding Models**

**Multimodal embedding models** address this by mapping different modalities into a shared vector space, where semantically related inputs—regardless of modality—are positioned close to each other.

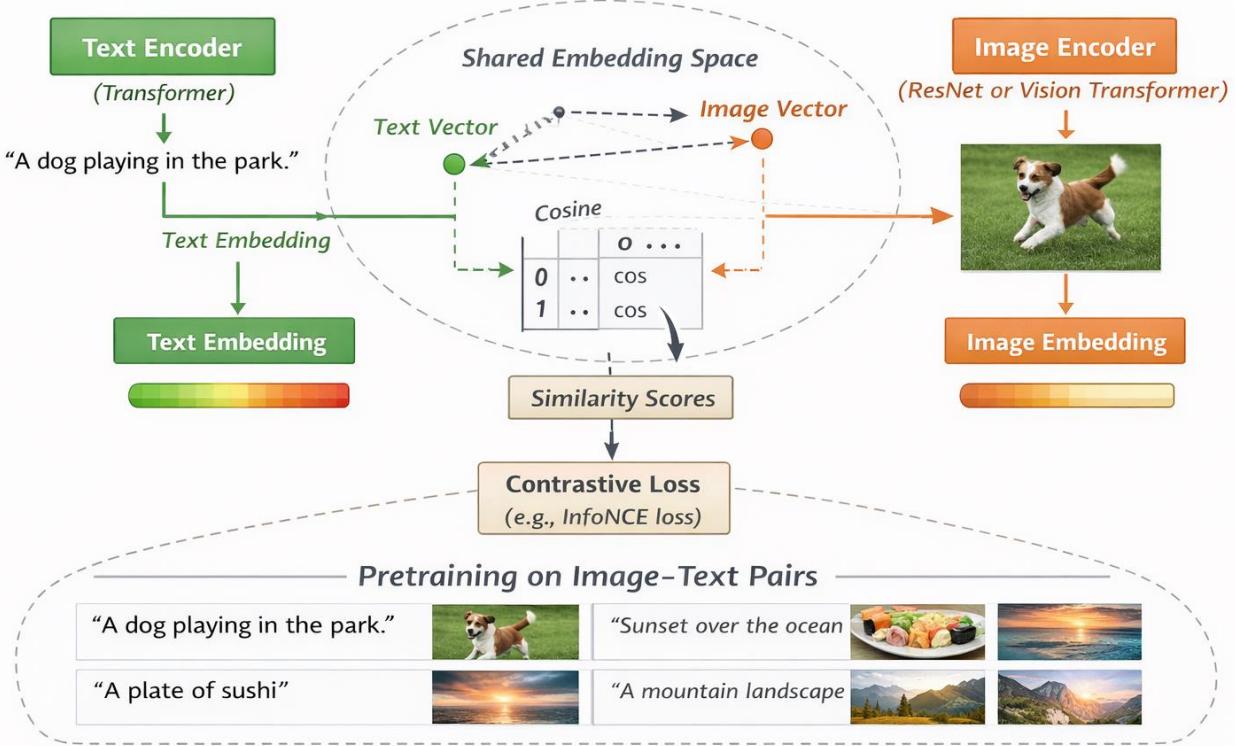
## **Shared Embedding Space Alignment**

**Example:**      Text: “A dog playing in the park”  
                  Image: Photo of a dog running on grass  
                  → Both vectors lie close in the embedding space



## CLIP: Contrastive Language–Image Pretraining

### CLIP: Architecture Overview



# Multimodal Text Generation Models

Multimodal text generation models are AI systems that create text by understanding and combining information from different types of inputs such as images, audio, and text.

## BLIP-2: Bootstrapping Language-Image Pre-training-2

Creating a multimodal language model from scratch requires significant computing power and data. We would have to use billions of images, text, and image-text pairs to create such a model. As you can imagine, this is not easily feasible.

Instead of building the architecture from scratch, BLIP-2 bridges the vision-language gap by building a bridge, named the Querying Transformer (Q-Former), that connects a pretrained image encoder and a pretrained LLM.

By leveraging pretrained models, BLIP-2 only needs to train the bridge without needing to train the image encoder and LLM from scratch

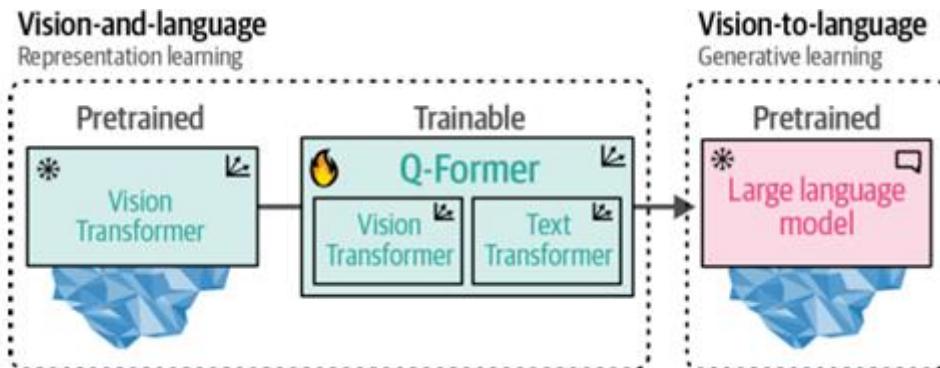


To connect the two pretrained models, the Q-Former mimics their architectures. It has two modules that share their attention layers:

- An **Image Transformer** to interact with the frozen Vision Transformer for feature extraction

- A **Text Transformer** that can interact with the LLM

The Q-Former is trained in **two stages**, one for each modality



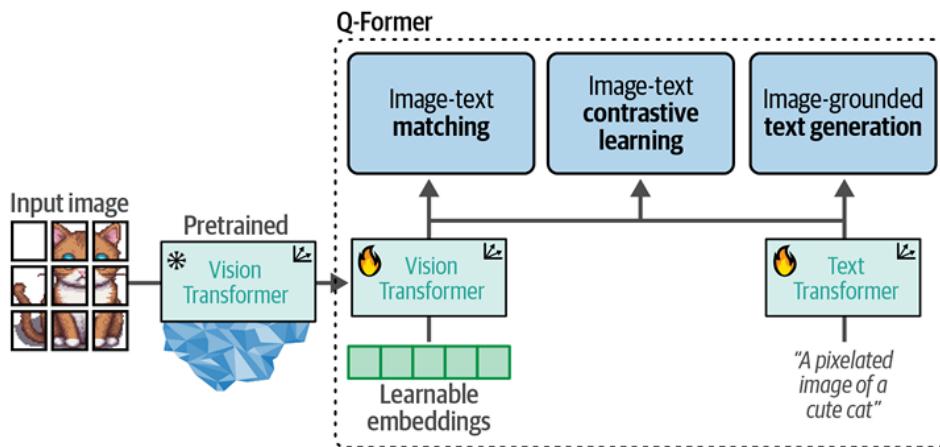
In step 1, image-document pairs are used to train the Q-Former to represent both images and text. These pairs are generally captions of images, as we have seen before when training CLIP.

The images are fed to the frozen ViT to extract vision embeddings. These embeddings are used as the input of Q-Former's ViT. The captions are used as the input of Q-Former's Text Transformer.

With these inputs, the Q-Former is then trained on three tasks:

- **Image-text contrastive learning** This task attempts to align pairs of image and text embeddings such that they maximize their mutual information.
- **Image-text matching** A classification task to predict whether an image and text pair is positive (matched) or negative (unmatched).
- **Image-grounded text generation** Trains the model to generate text based on information extracted from the input image.

These three objectives are jointly optimized to improve the visual representations that are extracted from the frozen ViT.



In step 2, the learnable embeddings derived from step 1 now contain visual information in the same dimensional space as the corresponding textual information. The learnable embeddings are then passed to the LLM. In a way, these embeddings serve as soft visual prompts that condition the LLM on the visual representations that were extracted by the Q-Former.

## Vision-and-language

Representation learning

