Get unlimited access to the best of Medium for less $5/week. Become a member

What are Visual Language Models and how do they work?

Karem Aydin Follow
Published in Towards AI · Feb 23, 2026 · 10 min read

In this article, we will delve into Visual Language models and understand how they work.

Introduction

Multimodal models are described as the machine learning models that can more than one type of input such as vision + text models. In this article we are going to dive into the vision language models which is only a type of multimodal models. These models are designed to understand and generate content that involves both images and text. Their tasks consist of image captioning, LLM's, visual question answering(VQA) + many more to imagine. LLM's for text to video (e.g. Sora) generation.

A video generated by OpenAI's Sora

Architecture of Visual Language Models

The architecture of Vlm changes according to the way they fuse the visual and textual modalities. Some architectures prefer to fuse or align the modalities in the early stage, some at intermediate stage and finally some at late stage

We can think of VLMs as large language models that understand both visual and textual inputs because that is simply what it is. Even though some details of the architecture vary, the main intuition is to convert the visual and textual into the same format such as embeddings and perform inference on the combination of the inputs. Embeddings are just vectors that represent the meaning of a token so they carry the semantic information in a numerical form.

There are many different architectures so we will delve into one of the most popular open source VLMs in the sector today which is LLaVA.

LLaVA(Large Language and Vision Assistant)

The LLaVA model, as described in the paper, is a sophisticated multimodal model that combines a pre-trained large language model with a pre-trained visual vision encoder for multimodal chat capabilities, specifically inspired by the GPT-4 framework. This model was initially introduced in the paper "Visual Instruction Tuning" authored by Liu from Columbia University.

In essence, LLaVA comprises two key components: a vision encoder and a language model called Vicuna, which is a fine-tuned version of the Llama 2 architecture. The vision encoder is built upon a pre-trained CLIP ViT-L/14 model, which excels in understanding visual content. Meanwhile, Vicuna serves as the language processing backbone, adept at handling textual data.

Architecture of LLaVA(Image by author)

During operation, the vision encoder processes images and converts them into embeddings, capturing the essential visual information in a format meaningful visual features. Simultaneously, the language model Vicuna processes textual inputs, generating corresponding embedding vectors. Notably, these embedding vectors share the same dimensional space, facilitating seamless integration of visual and textual information within the model. This integration enables LLaVA to effectively understand and interpret multimodal inputs, bridging the gap between visual and linguistic understanding.

There are two stages in the training of the LLaVA model:

1. **Pre-training for Feature Alignment.** Only the projection matrix is updated based on CC3M to improve the connection between textual and visual inputs.

Illustration from Unsplash

2. **Fine-tuning End-to-End.** Both the projection matrix and LLM are updated for two different use scenarios:
   - **Visual Chat**: LLaVA is fine-tuned on our generated multimodal instruction-following data for daily user-oriented applications.
   - **Science QA**: LLaVA is fine-tuned on this multimodal reasoning dataset for the science domain.

Architecture of Visual Reasoning

Components of LLaVA model

LLaVA consists of various different modules that work together to understand both modalities. As we said, visual encoders create a representation of an image and embedding model creates a representation of text in a numerical form.

1. **Vision Encoder**

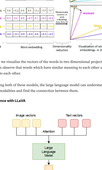
Pictures in an image

Visual encoders typically operate by dividing an image into smaller patches, often organized in grids of 14 x 14 pixels each. Within each patch, the individual pixels are processed and transformed into numerical vectors. These vectors, which capture various features of the image, are then passed through the image encoder.

The image encoder consists of multiple blocks, each containing layers designed to enhance the representation of the image. These blocks typically include feedforward layers, which perform computations to extract higher-level features, and attention layers, which help the model focus on relevant parts of the image. These blocks are stacked sequentially to progressively refine the image representation.

Ultimately, the output of the process is a condensed and enriched representation of the image, encoding its visual content in a format that can be effectively utilized by downstream tasks or models for further analysis or processing.

Vision encoders undergo training through a process of minimizing the disparity between the vector representations of images and their corresponding text descriptions. Both images and texts are converted into numerical embeddings, which are essentially compact representations in a vector format. The goal of this training process is to develop a model adept at transforming the embeddings of an image into a representation that closely aligns with its textual counterpart.


Pre-training
Fine-tuning
Image Encoder
Text Encoder
Image Encoder
Text Encoder
Image Encoder
Text Encoder

See more recommendations