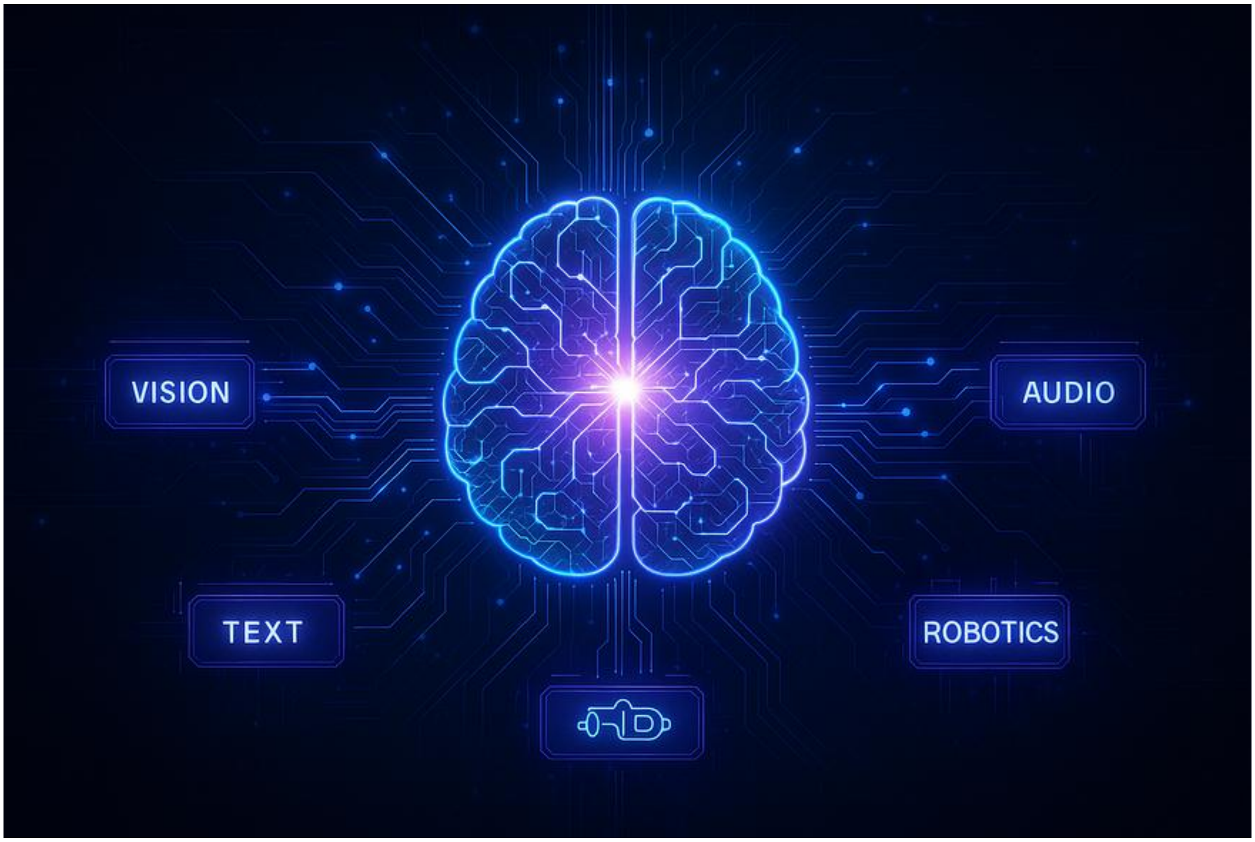
Not Everything Is an LLM: 8 AI Model Types You Need to Know in 2025

Beyond ChatGPT, A beginner's guide to today's essential AI models



In 2023, if you said **"AI"**, most people thought of **ChatGPT.**

Fast-forward to 2025, and the landscape looks very different. LLMs (Large Language Models) may have ignited the AI revolution, but now we're deep into an era of **specialized AI models**, each designed with a specific superpower.

Yet, somehow, **everyone still calls them LLMs.**

It's like calling every vehicle a **"car"**, whether it's a bicycle, a truck, or a plane. Sure, they all move, but they're built for very different purposes.

If you're an AI researcher, startup founder, PM, or just someone trying to keep up, understanding the difference between an **LLM, LAM, SLM, MoE, and more** is no longer a nice-to-have.

It's a **competitive edge**.

So, Let's break down 8 powerful AI model types and what they're ***really*** built to do.

1. LLM — Large Language Model

What Is an LLM, Really?

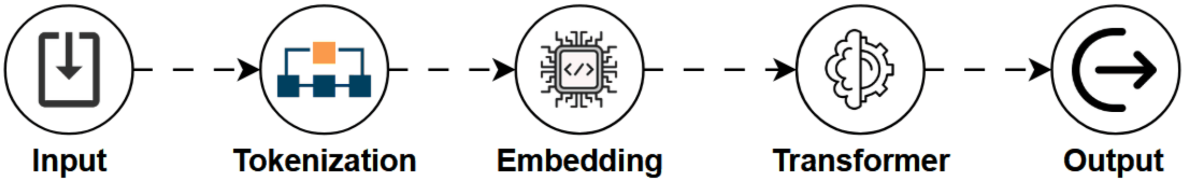


Diagram drawn using **Draw.io** (by the author)

Imagine you're texting a super-intelligent friend who can complete your sentences, write essays, debug code, and even pretend to be Shakespeare, all in one breath.

That's essentially what an **LLM (Large Language Model)** does.

LLMs are trained on **massive amounts of text** from the internet, books, articles, code, tweets to learn how language works.

Their goal? To **predict the next word (or token)** in a sequence, based on everything that came before.

Think of it like supercharged **autocomplete,** but instead of just finishing your sentence, it can write an entire book, answer philosophical questions, or build a working website.

Why Are LLMs So Popular?

They became the ***poster child*** of AI in recent years for a few reasons,

- **Conversational Power**: ChatGPT, Claude, Gemini — all powered by LLMs.
- **Code + Content**: From blog articles to Python scripts, LLMs handle creative and technical tasks.
- **General Knowledge**: They **"know"** a bit about almost everything, making them great general-purpose tools.

Real-World Use Cases

- Writing and rewriting content

- Programming assistance and code generation
- Customer service chatbots
- Brainstorming ideas
- Language translation
- Education and tutoring

In short, if it involves **words**, LLMs are likely involved.

But There's a Catch…

While LLMs seem magical, they have limitations,

- They can **hallucinate** (make things up confidently)
- They're **computationally expensive** to run
- They lack **true understanding or reasoning**, they're guessing based on patterns

That's why new model types, built for **speed, specialization,** or **deeper reasoning** are emerging fast.

2. LCM — Latent Consistency Model

What Is an LCM, and Why Should You Care?

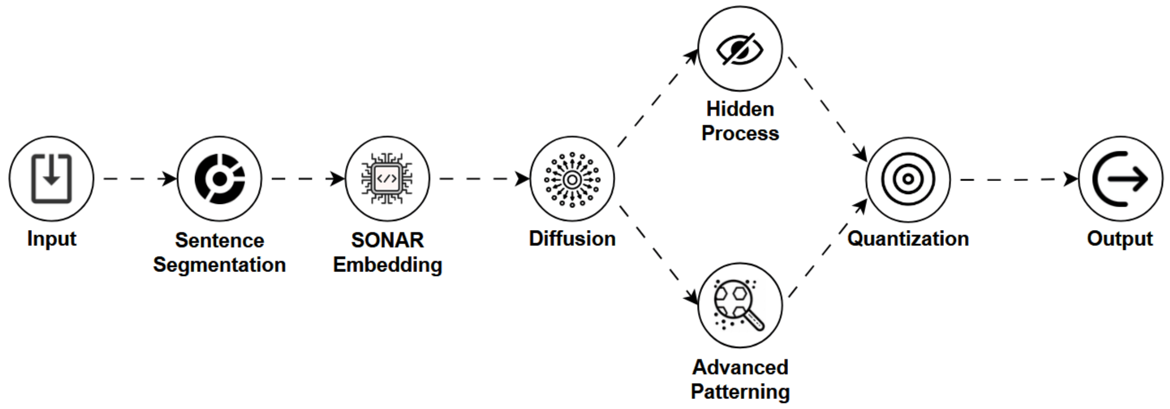


Diagram drawn using **Draw.io** (by the author)

Picture this: you're using an AI image generator on your phone, and it gives you a crisp result in under a second, no cloud connection, no heavy lifting.

That's the power of **LCMs (Latent Consistency Models)**.

Unlike LLMs that generate text, LCMs are designed primarily for **images**, and they're optimized for **speed, efficiency, and small devices**. They're the fast, lightweight cousins of the more heavyweight image generation models like Stable Diffusion.

Think of LCMs as the **real-time engines** of the AI world, designed to work smoothly even on mobile devices or low-powered edge hardware.

How Do They Work?

LCMs build on the concept of **diffusion models**, a class of models that gradually "denoise" random patterns into meaningful images. But instead of needing dozens of slow steps to do this, **LCMs shortcut the process** by learning **consistent patterns in a compressed (latent) space**.

Imagine sketching a face. A normal model draws 50 lines slowly. LCM? Just a few confident strokes and it's done.

Real-World Use Cases

- **On-device image generation** (think AI filters or avatars)
- **AR/VR applications** where speed is critical
- **Faster prototyping tools** for designers
- **Real-time vision enhancement** on smart cameras

In essence, **LCMs** are the go-to model when **you want fast, beautiful results without needing a supercomputer**.

Why They Matter in 2025

We're moving into an era of **edge computing**, where devices generate content locally for speed and privacy. LCMs are a big part of this shift.

In the future, your smart glasses or smartwatch might generate and enhance images using an LCM, all on the fly.

3. LAM — Language Action Model

What Exactly Is a LAM?

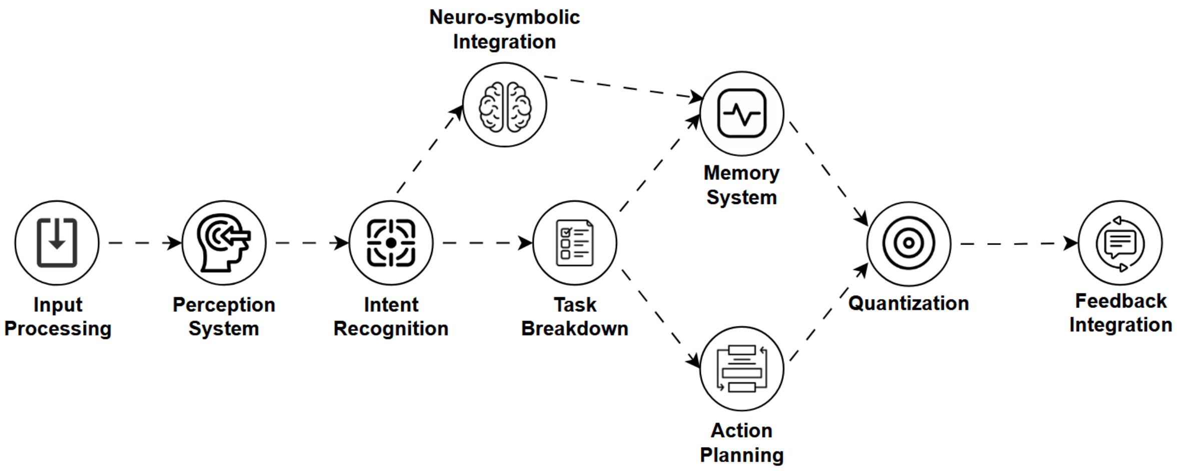


Diagram drawn using **Draw.io** (by the author)

If an **LLM** is your chatty friend and an **LCM** is your quick-drawing artist, then a **LAM** is your **smart assistant that plans, remembers, and executes tasks**.

LAM (Language Action Model) bridges the gap between **understanding language and taking meaningful actions**. It doesn't just generate text, it **understands intent**, **remembers context**, and **interacts with tools or environments**.

Think of LAMs as the **backbone of AI agents**, the kind of models that can help automate tasks, operate software tools, or plan multi-step actions like booking a trip or debugging code.

How Does It Work?

LAMs typically combine,

- **LLMs** for natural language understanding,
- **Memory modules** for keeping track of past actions or inputs,
- **Planners** that can break down complex tasks,
- **Tool use** capabilities to actually execute steps (e.g., via APIs or interfaces).

Imagine asking your AI, ***"Book a flight to Tokyo, compare hotel prices, and set a reminder for my visa appointment."***

A pure **LLM** might just give you suggestions.

A **LAM**? It **acts**, checking calendars, querying APIs, and building a task flow behind the scenes.

Real-World Use Cases

- **AI agents** that automate workflows (e.g., Zapier AI, Devin)
- **Digital assistants** that interact with apps and services
- **Customer support bots** that solve problems, not just reply
- **Productivity tools** that complete tasks based on instructions
- **Robotics**, where language input controls physical actions

Why LAMs Matter in 2025

LLMs changed the game by understanding text. But LAMs are pushing things forward by **doing things**.

In a world of increasing automation, LAMs are unlocking AI that can work across apps, understand **long-term goals**, and adapt to changing environments.

Imagine an AI that not only drafts your email but also sends it, follows up, and schedules a meeting, all based on one prompt.

4. MoE — Mixture of Experts

What Is a MoE Model?

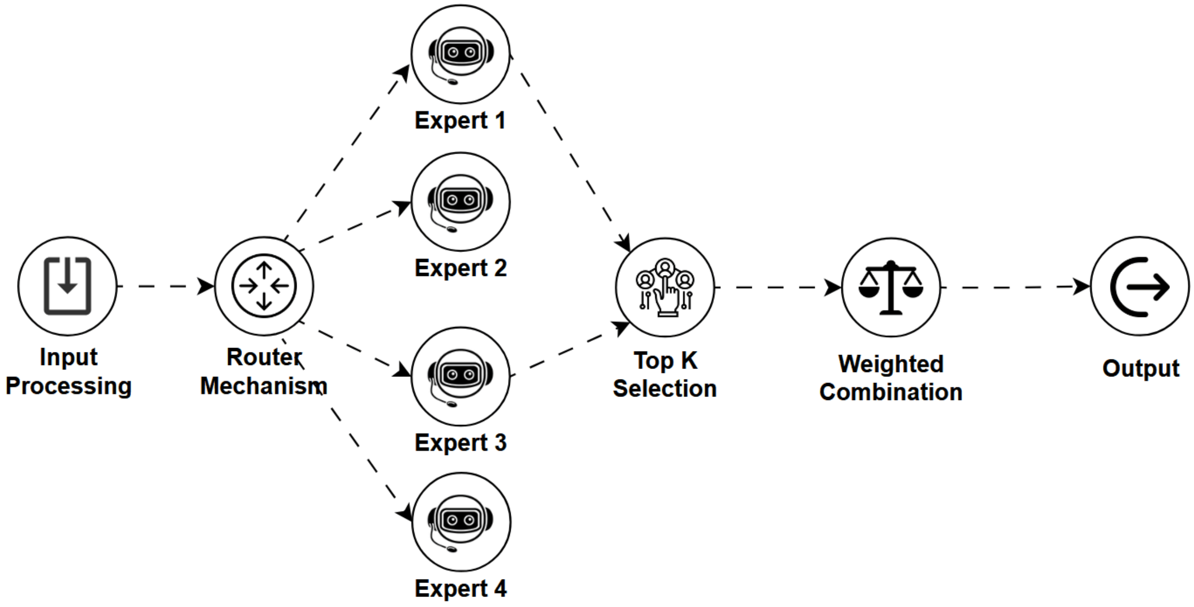


Diagram drawn using **Draw.io** (by the author)

Imagine you're asking a big question and instead of getting an answer from one generalist, you're directed to a **team of specialists**, each an expert in a narrow domain.

That's what **MoE (Mixture of Experts)** models do.

A Mixture of Experts model is made up of **many sub-models ("experts")**, but when a prompt comes in, **only a few experts are activated** based on what's relevant. This makes the model **scalable** and **efficient**, because not every expert is used every time.

Think of it like consulting the best surgeon for surgery, the best chef for cooking, and the best mechanic for your car, all within one AI.

How It Works

MoE uses a **"router"**, a smart internal system that decides which expert(s) to activate based on your input.

- The router evaluates the input.
- It chooses the top N experts (often 2 out of 100+).
- Only those selected experts process the input and return an output.
- This output is combined and returned to the user.

So you get **targeted intelligence** with **minimal compute overhead**.

Real-World Use Cases

- **High-performance AI at scale** (e.g., Google's Switch Transformer, GShard)
- **Efficient cloud inference** — fewer resources, faster outputs
- **Domain-specialized assistants** (e.g., a medical expert vs. a legal expert)
- **Multilingual systems** — experts for different languages
- **Fine-grained personalization —** experts tuned to user behavior or tasks

Why MoE Models Matter in 2025

With AI models growing into **hundreds of billions of parameters**, compute costs are becoming a bottleneck. MoE models provide a brilliant workaround, **you scale wide without scaling heavy**.

By activating only what's needed, MoEs offer a **massive increase in performance** without needing supercomputers for every query.

Imagine a model that's 10x larger but only costs as much to run as a model half its size. That's the power of MoE.

They also make way for **more modular and expandable systems**, where new experts can be added without retraining the entire model.

5. VLM — Vision Language Model

What Is a VLM?

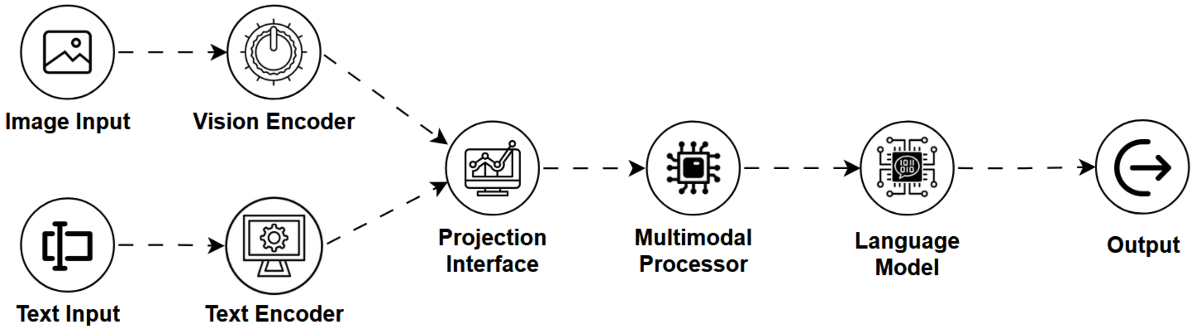


Diagram drawn using **Draw.io** (by the author)

Imagine an AI that **sees** an image and **reads** your caption or query and then responds with deep understanding of both.

That's the magic of a **Vision Language Model (VLM)**. These models are designed to process and understand **both visual** and **textual** inputs simultaneously.

They're like the Swiss Army knife of AI, combining the perception of vision models with the reasoning power of language models.

How It Works

At the core of a VLM is a **shared embedding space**, a special zone where images and text are mapped into similar **"meaningful"** numerical representations.

This allows the model to **match images to descriptions**, **answer questions about visual content**, or even **generate text from images** and vice versa.

Here's a simplified flow,

1. Image goes through a **vision encoder** (like a modified transformer or CNN).

2. Text goes through a **language encoder** (like BERT or GPT).
3. Both are aligned in a **shared latent space** for cross-modal understanding.
4. The model produces outputs such as answers, captions, classifications, etc.

Real-World Use Cases

- **Multimodal assistants** (e.g., ChatGPT-4o, Gemini)
- **Image captioning**
- **Visual question answering (VQA)**
- **Search engines that understand both text & image queries**
- **Accessibility tools** (e.g., for visually impaired users)
- **Robotics** — interpreting surroundings using both vision and instruction
- **AR/VR** — contextual interaction with the real world

Example: *You upload a photo of a cracked phone screen and ask,* ***"Can I still use this?"*** *A VLM can analyze the image, understand the question, and respond helpfully.*

Why VLMs Matter in 2025

In a world where digital content is **increasingly visual**, we need models that go beyond text-only capabilities. VLMs are foundational to,

- **Multimodal search**
- **Context-aware agents**
- **Assistive AI for real-world perception**

They are key to bridging the gap between **language-driven interfaces** and the **visual-first world** we live in, making AI more intuitive and human-friendly.

VLMs also serve as the building blocks for **embodied AI.** Systems that can "see," "understand," and "act" in physical or virtual environments.

6. SLM — Small Language Model

What Is a Small Language Model?

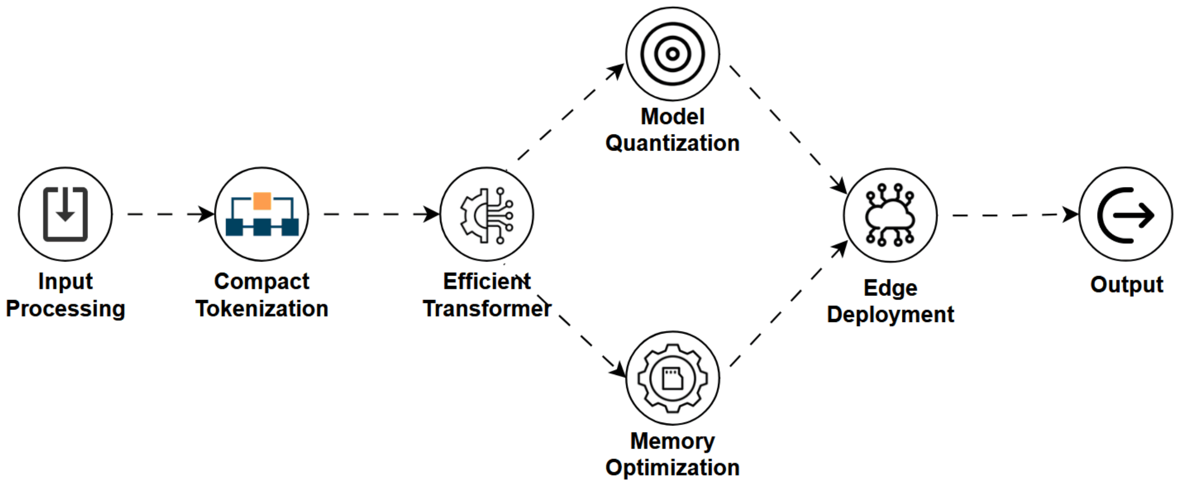


Diagram drawn using **Draw.io** (by the author)

While LLMs grab the spotlight with their massive scale, Small Language Models (SLMs) work quietly in the background. **On your phone, your laptop, or even your smart toaster**.

SLMs are **compact, efficient language models** designed to deliver fast, low-latency responses on limited hardware.

Think of them as the LLM's minimalistic cousin, less compute-hungry but still impressively capable.

How It Works

SLMs are typically built using the same transformer architecture as LLMs, but with **fewer parameters** and **optimized inference paths.**

- **Parameter count:** Usually in the millions (vs. billions or trillions in LLMs).
- **Optimizations:** Quantization, pruning, knowledge distillation, or architectural tweaks.
- **Deployment:** Edge devices (phones, IoT), browsers, local servers.

While they may lack the deep reasoning and context memory of LLMs, their **lightweight footprint** allows for real-time, offline performance.

Real-World Use Cases

- **On-device chatbots** (e.g., mobile virtual assistants)
- **Smart appliances and embedded systems**
- **Privacy-first applications** (data never leaves your device)

- **Developer tools and code autocomplete on local IDEs**
- **Real-time inference in robotics or AR headsets**

Example: Imagine asking your smart TV, **"What's a good movie like Interstellar?"** and getting an instant answer without pinging the cloud. That's an SLM at work.

Why SLMs Matter in 2025

As AI becomes more integrated into daily life, the demand for **low-latency**, **energy-efficient**, and **privacy-respecting** models is surging.

SLMs unlock,

- **Offline intelligence** — no internet? No problem.
- **Data sovereignty** — keep sensitive data on-device.
- **Scalable deployment** — from smartphones to smart meters.

And with projects like **Phi-3, TinyLLaMA, and Apple's rumored on-device models**, SLMs are entering a golden era.

"Not every task needs a supercomputer. Sometimes, a smart calculator does the job just fine."

7. MLM — Masked Language Model

What Is a Masked Language Model?

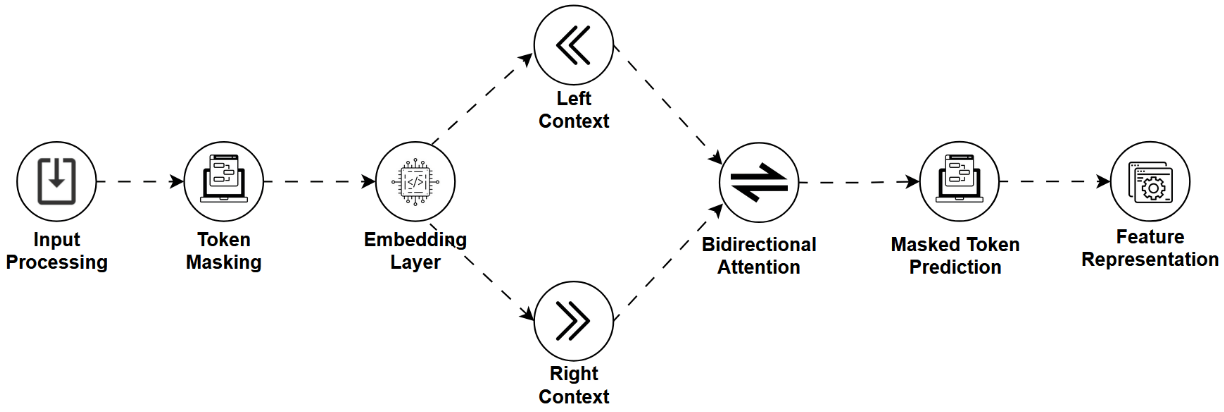


Diagram drawn using **Draw.io** (by the author)

Before ChatGPT was dazzling the world with fluent essays and code generation, there was **BERT**, and with it came the **Masked Language Model (MLM)**.

MLMs are trained by **masking random words in a sentence and having the model predict the missing ones**. It's a bit like a fill-in-the-blank puzzle except the model learns **deep, bidirectional understanding** of language by doing it.

Instead of predicting the next word like LLMs, MLMs look at the whole sentence and reason about what should go in the blank.

How It Works

Let's say we mask a sentence like

"The Eiffel Tower is located in **[MASK].**"

An MLM will use both the left and right context ("The Eiffel Tower is located in ...") to predict the missing word, in this case, "Paris."

This approach helps the model understand,

- **Syntax** (grammar and structure)
- **Semantics** (meaning and relationships)
- **Context** from both directions (bidirectional learning)

MLMs are usually **pretrained** on massive text corpora and then **fine-tuned** for specific tasks.

Real-World Use Cases

MLMs may not be flashy, but they are **powerful workhorses** in many AI systems,

- **Search engines** (semantic matching of queries and results)
- **Text classification** (spam detection, sentiment analysis)
- **Named Entity Recognition** (identifying names, dates, organizations)
- **Embeddings for vector databases**
- **Pretraining for other model types**

Example: When you search for **"cheap hotels near me"**, the model understands that "cheap" relates to price, "hotels" are accommodations, and "near me" depends on location. That's deep semantic parsing powered by MLMs.

Why MLMs Still Matter

Despite the surge in autoregressive models (LLMs), MLMs continue to shine in scenarios that require:

- **Bidirectional understanding**
- **Strong contextual representations**
- **Lower compute needs for training**

They are often the **foundation** for larger systems, or used in **hybrid approaches** where models like BERT handle representation while LLMs handle generation.

And they're evolving too with models like **RoBERTa**, **DeBERTa**, and **E5** offering optimized variations for different tasks.

"Masked language modeling is like learning to read between the lines and then predicting what the lines actually say."

8. SAM — Segment Anything Model

What Is SAM?

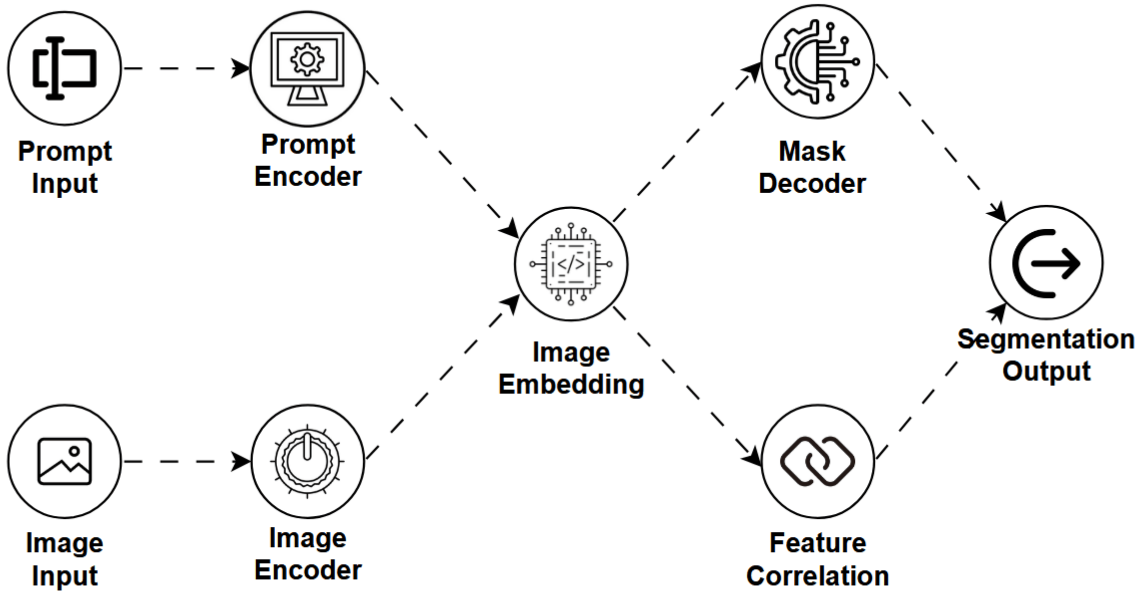


Diagram drawn using **Draw.io** (by the author)

The **Segment Anything Model (SAM)** by Meta AI is a game-changer in **computer vision**.

Unlike models that classify or detect whole objects, SAM **segments**, meaning it draws precise outlines ***around every object*** in an image, even those it hasn't seen before. It doesn’t just label “cat” or “dog”. It understands their **shape, boundaries, and position** with pixel-level precision.

Imagine dropping a photo into a model and instantly getting every object neatly cut out. That’s the magic of **SAM.**

How SAM Works

At its core, SAM is built for **promptable segmentation**. You give it a prompt (a point, a box, or a mask), and it returns the exact segment of the object you’re referring to.

It uses,

- A **Vision Transformer** backbone to process the image
- An **embedding-based approach** to compare visual features
- A fast segmentation decoder that outputs masks instantly

And here’s the kicker. It can segment **anything**, even if it hasn’t been explicitly trained on that object class.

It’s not just trained to **“know”** what a cat is. It’s trained to **“see”** any object in visual space.

Real-World Use Cases

SAM is making waves across industries,

- **Medical Imaging**: Identifying tumors or organs in scans with surgical precision
- **Augmented Reality (AR)**: Real-time object detection and masking
- **Robotics**: Helping machines understand and interact with their environment
- **Video Editing**: Instant background removal, object isolation
- **Scientific Research**: Segmenting cells in microscopy images or objects in satellite images

Example: A medical researcher can segment a brain tumor in an MRI scan just by clicking near it. No manual outlining. No training needed. That’s SAM at work.

Why SAM Is a Big Deal

Segmenting everything, not just known categories — unlocks a new paradigm in AI vision.

- **Zero-shot generalization** (works on unseen objects)
- **Fast and interactive** (real-time or near real-time)

- **Modular** (can be paired with other models like VLMs or LAMs)

It's the **LEGO brick** of vision AI. Pluggable, flexible, and incredibly powerful.

SAM is already being integrated into larger multimodal systems. When combined with VLMs (like GPT-4o or Gemini), you get models that can **see, understand, and act**, making it a vital part of the **next generation of AI agents**.

Pro Tip

While SAM focuses purely on visual segmentation, you can pair it with **language models** or **action models** to create powerful visual agents, like a robot that sees an object, understands what it is, and picks it up.

Wrapping It All Up

Let's take a step back.

From **LLMs** writing essays, to **SLMs** powering chatbots on your phone, to **SAM** dissecting images pixel by pixel, the AI landscape is ***far richer*** than just **"language models."**

Each model type — **LLM, LCM, MoE, LAM, VLM, SLM, MLM, SAM** — is a **tool in the AI toolbox**, specialized for its domain, designed with specific capabilities in mind.

So what's the takeaway?

- **Use the right model for the job**, not everything needs an LLM.
- **Understand the differences**, architecture informs application.
- **Think in systems, not silos**, the future is multimodal, multi-agent, and deeply specialized.

Which AI model are you most excited to explore? Already building, or just getting started? Drop a **comment** below, share your thoughts, ask a question, or tell us what you're curious about. **Let's learn from each other and grow together.**

Remember, the future of AI isn't just in the hands of experts. It's shaped by curious minds like yours. Stay bold, keep exploring, and who knows? Your next idea could be the one that changes everything.

If you found this article helpful and would like to **support** more content like this, you can ***buy me a coffee here.***