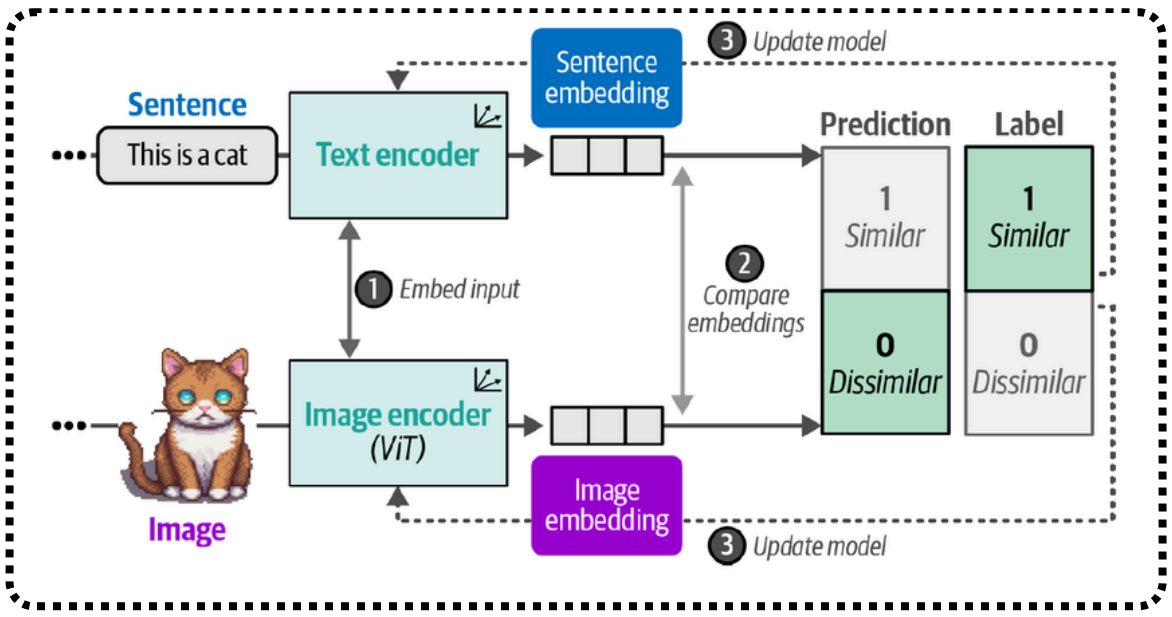
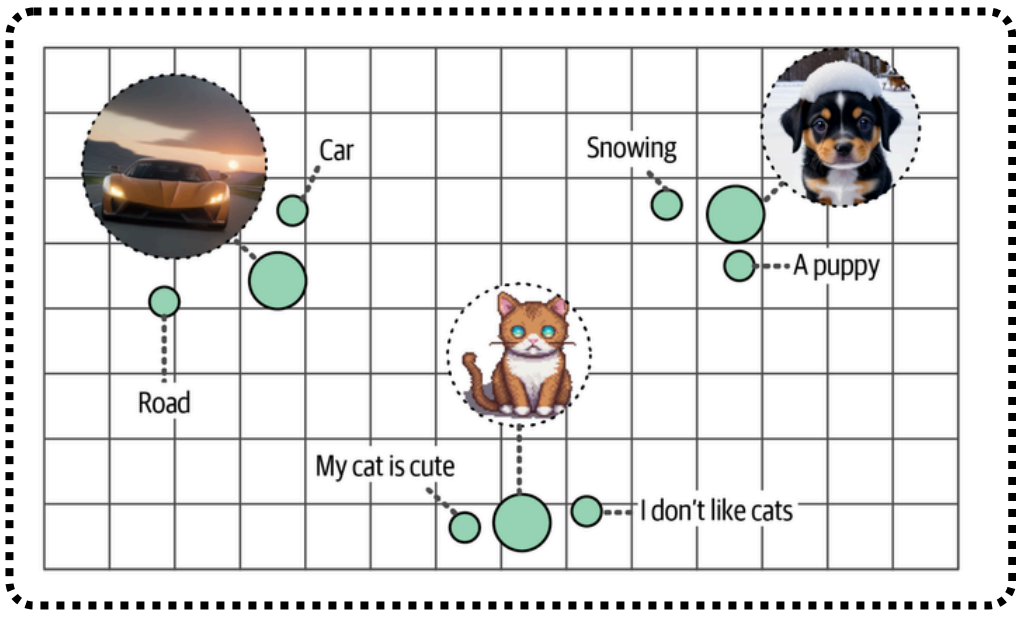
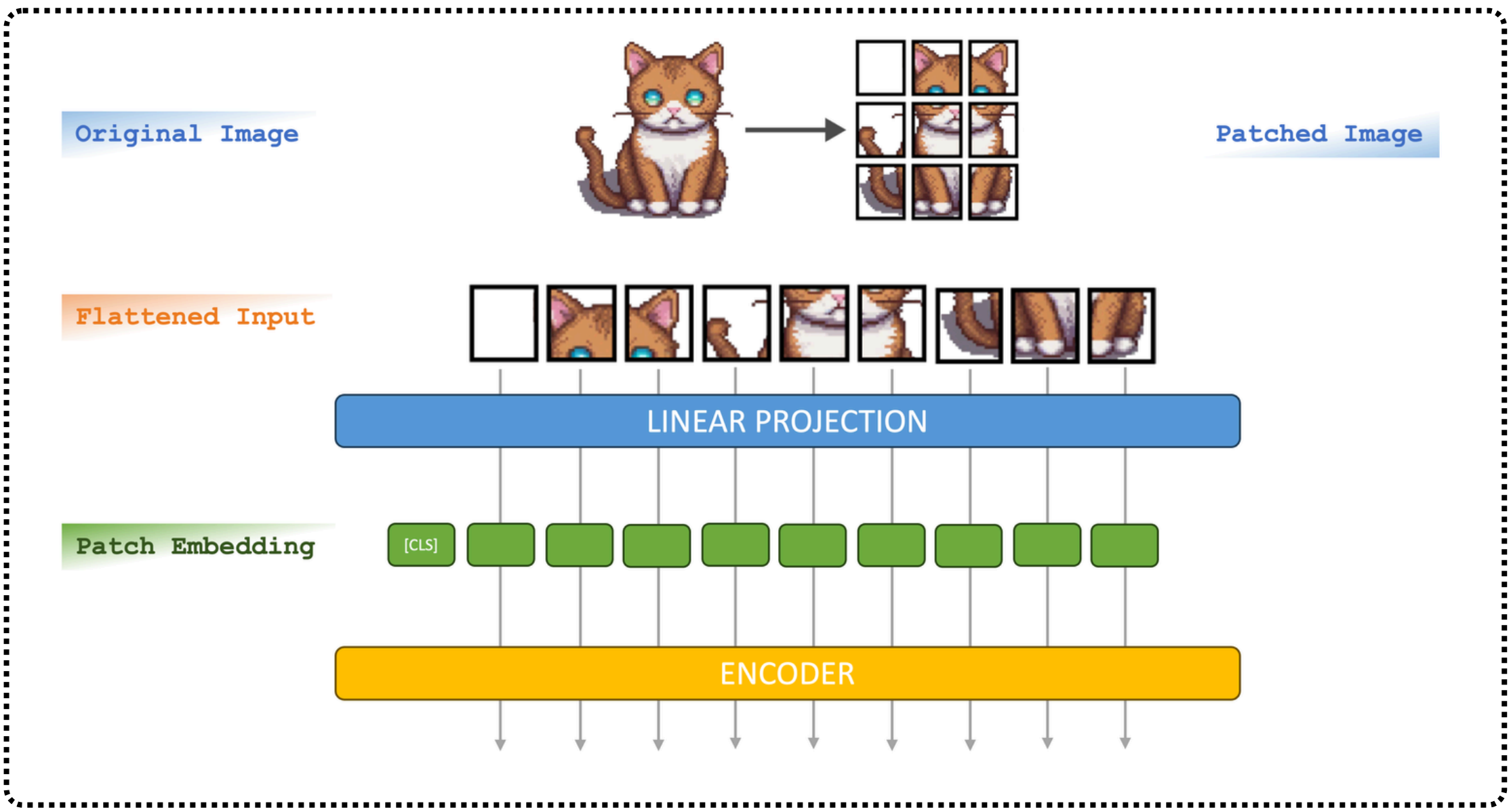
Analytics Vidhya

Multimodal Large Language Models







Introduction

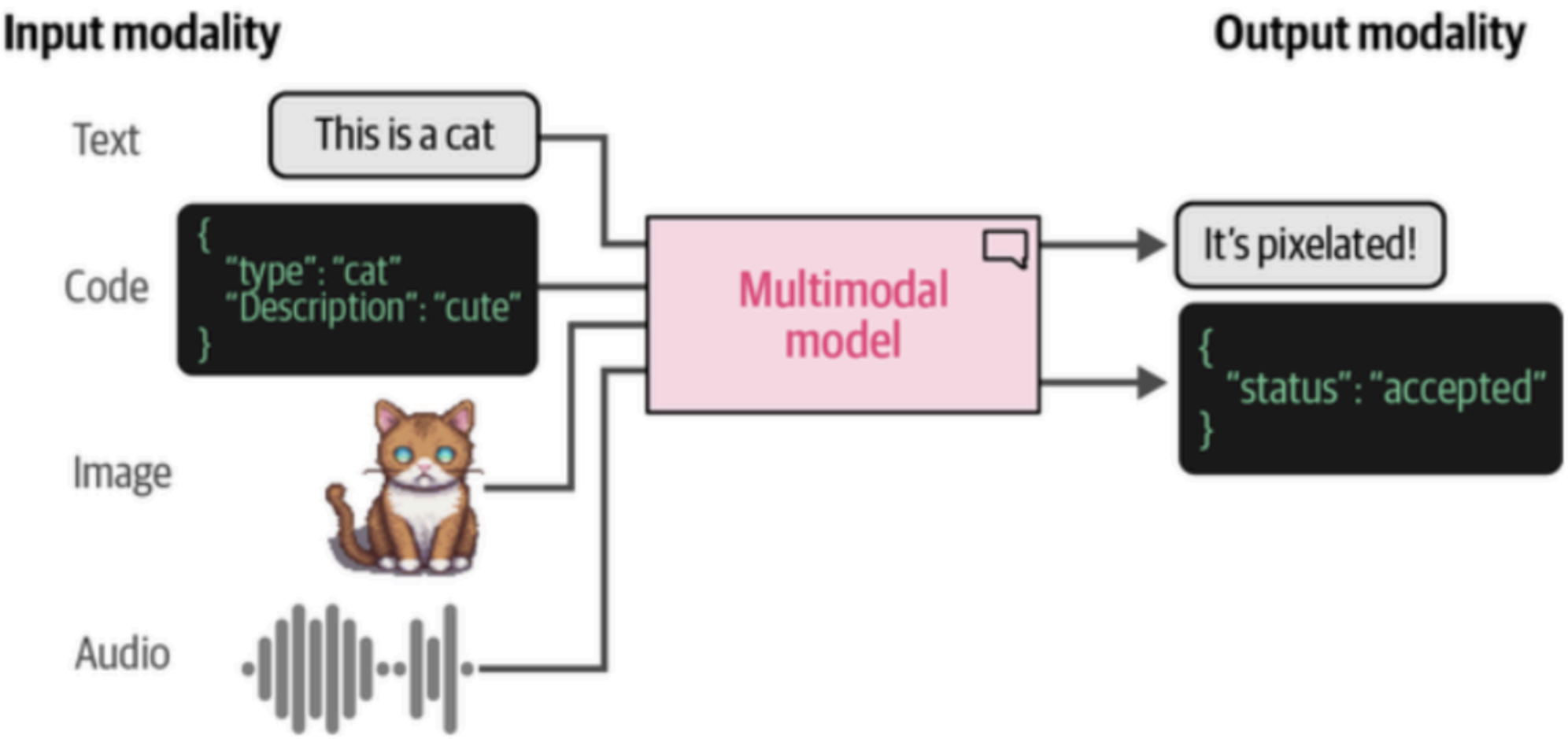
Multimodal models are advanced AI systems capable of processing and integrating multiple types, or modalities, of data, such as text, images, audio, and even video. Traditional large language models (LLMs) are designed primarily to work with text data. However, enabling these models to handle various forms of data significantly broadens their utility and real-world applicability.

For instance, a multimodal model can:

- Analyze Images: Interpret visual content and answer questions about images.
- Process Audio: Understand spoken language or identify sounds, enabling interaction through speech.
- Combine Modalities: Take inputs from both text and images, providing responses that integrate information from both sources.

By handling multiple modalities, multimodal models can perform complex tasks, such as:

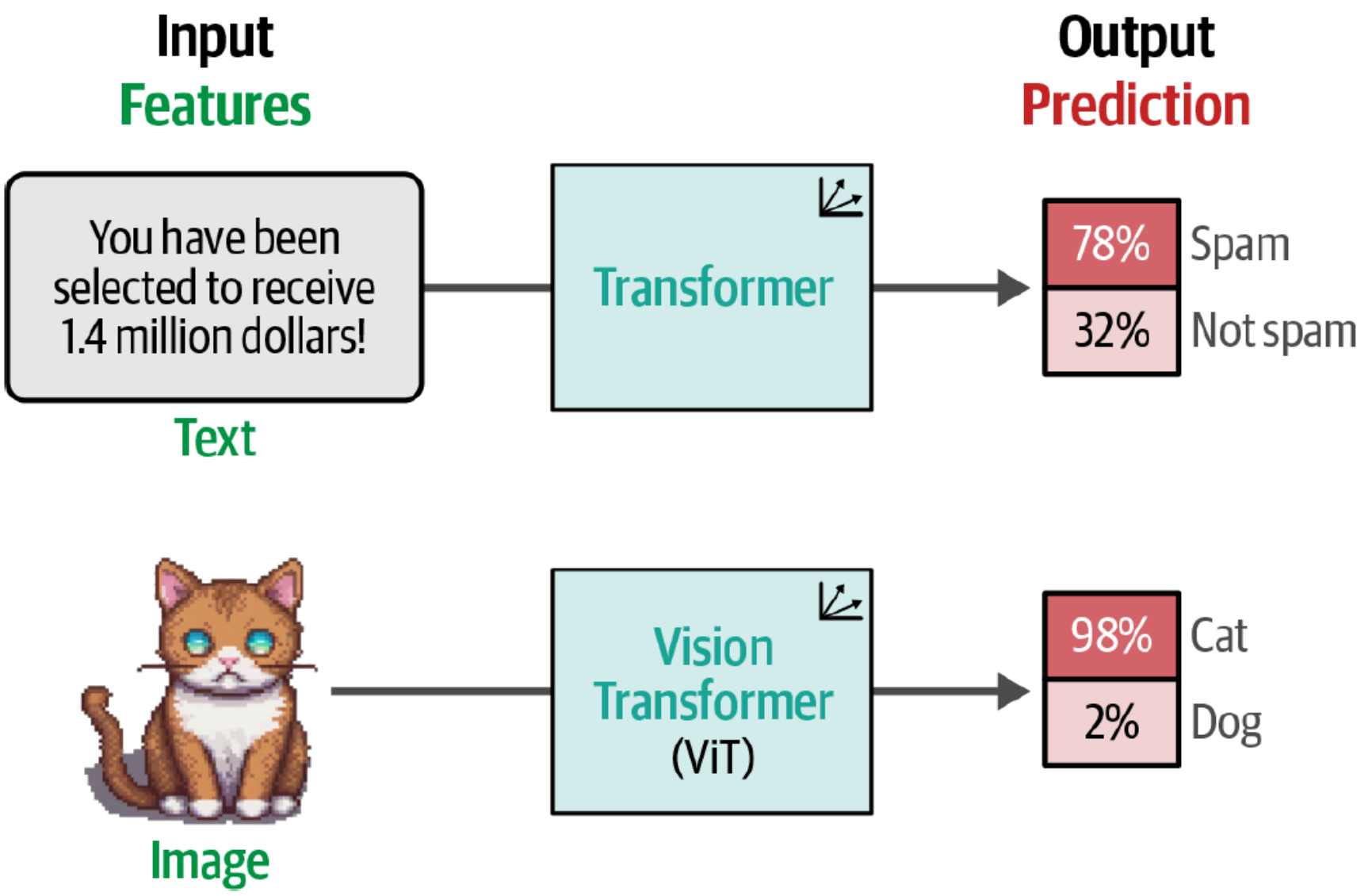
- Image Captioning: Generating a description of an image.
- Visual Question Answering: Answering questions about specific elements in an image.
- Speech Recognition and Synthesis: Converting spoken language to text and vice versa.



How can we have images have a numerical representation such that we can make it work with the original Transformer technique? And how LLMs can be extended to include vision tasks using this Transformer.

Transformers for Vision

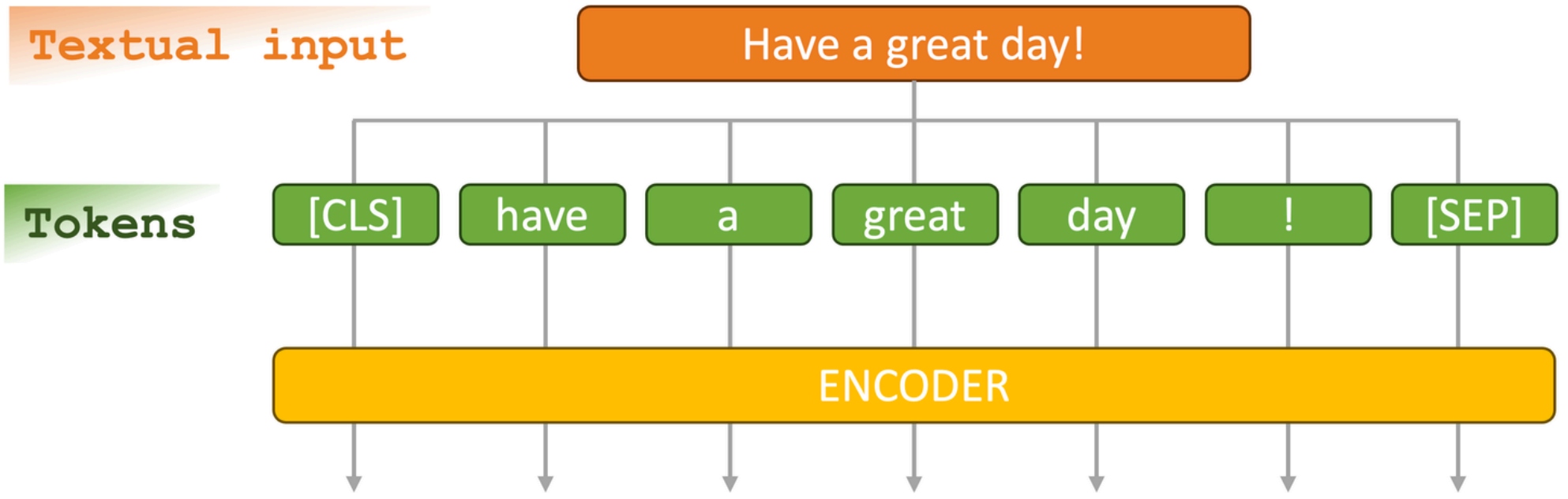
Both the original Transformer and the Vision Transformer process unstructured data by converting it into numerical representations, which are then used for tasks like classification.



- Textual Input: The sentence "Have a great day!" is used as an example of textual input.

- Tokenization: The text is tokenized into individual units (tokens) using a tokenizer, breaking it down into components like [CLS], "have," "a," "great," "day," "!" and [SEP]. This tokenization is essential for preparing the text for encoding.

- Encoding: Once tokenized, the tokens are passed to one or more encoders within the Transformer model, which processes these tokens to generate numerical representations used in downstream tasks like classification.



Challenge with Images: Unlike text, images do not consist of words, so the standard text-based tokenization approach cannot be directly applied to visual data.

Vision Transformer (ViT) Solution: To address this, the authors of the Vision Transformer (ViT) developed a method to tokenize images by dividing them into smaller patches or “words.” This allows images to be processed through the original Transformer encoder structure, making it possible to apply the self-attention mechanism to image data.

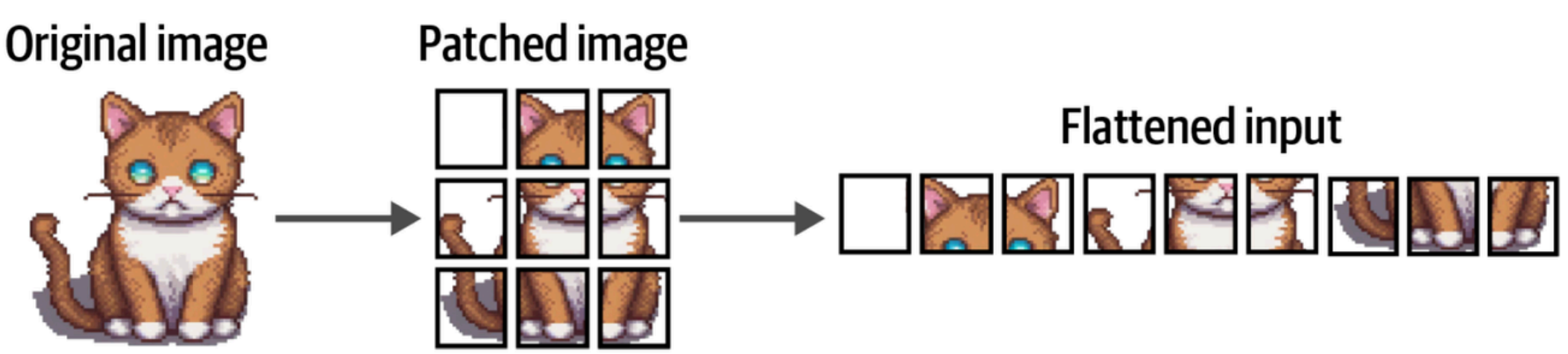
Vision Transformer (ViT)

- Transformers for Vision, specifically the Vision Transformer (ViT), adapt the Transformer architecture.
- Originally designed for language processing—to work with images.

- Unlike traditional convolutional neural networks (CNNs), which have been the standard for image recognition, Vision Transformers divide images into smaller patches and convert these patches into numerical representations, similar to how text is tokenized for NLP models.

- In this approach, the encoder component of the Transformer takes these image patches as input, transforming them into a format that the model can process. This allows the Vision Transformer to perform tasks like image classification effectively, using the same self-attention mechanisms that made Transformers successful in language tasks.

- ViT has shown strong performance in image recognition tasks, sometimes surpassing CNNs by leveraging the flexibility and scalability of Transformer architectures.

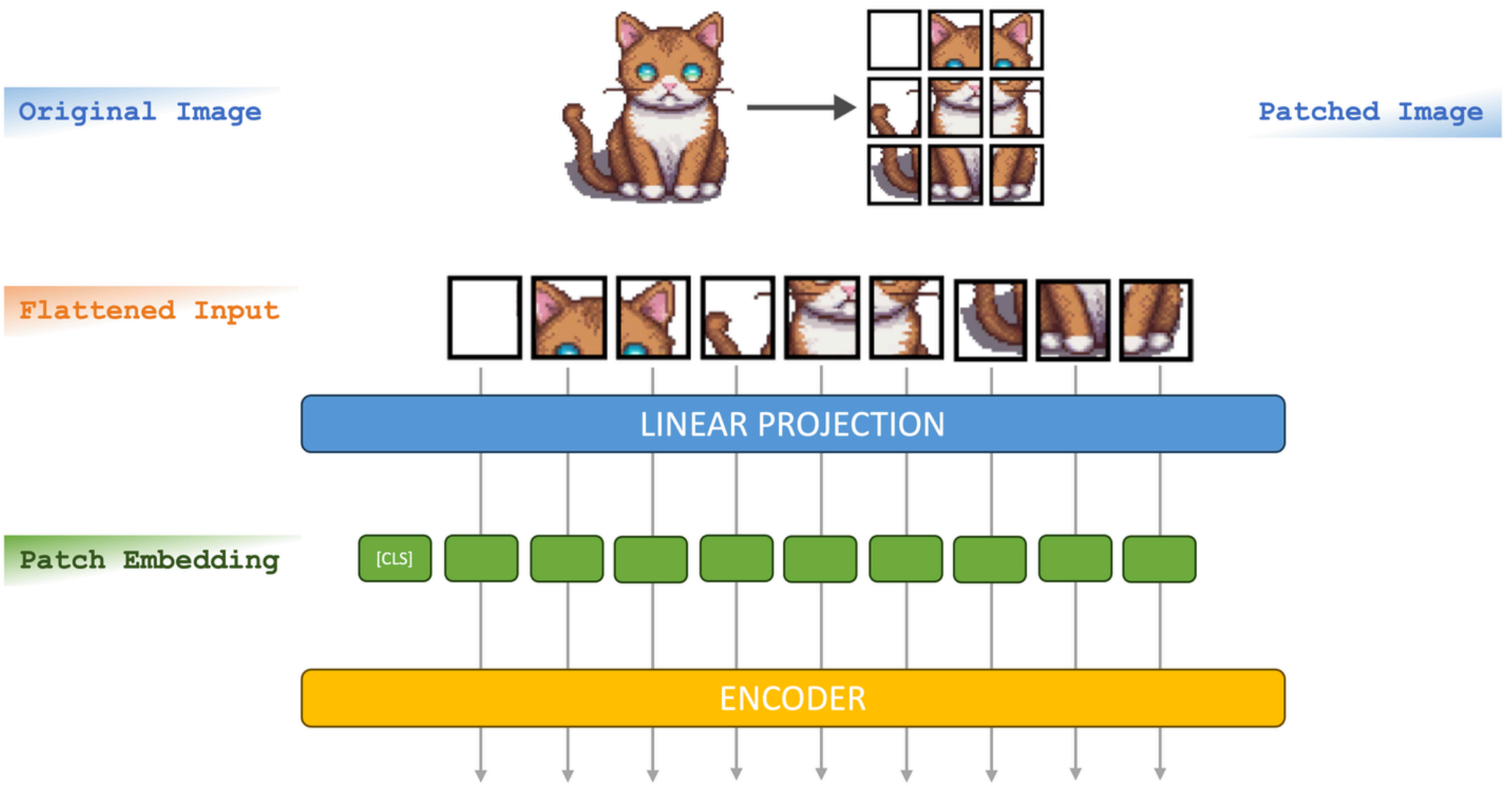


- The images in the examples were patched into 3 × 3 patches but the original implementation used 16 × 16 patches The flattened input of image patches can be thought of as the tokens in a piece of text.

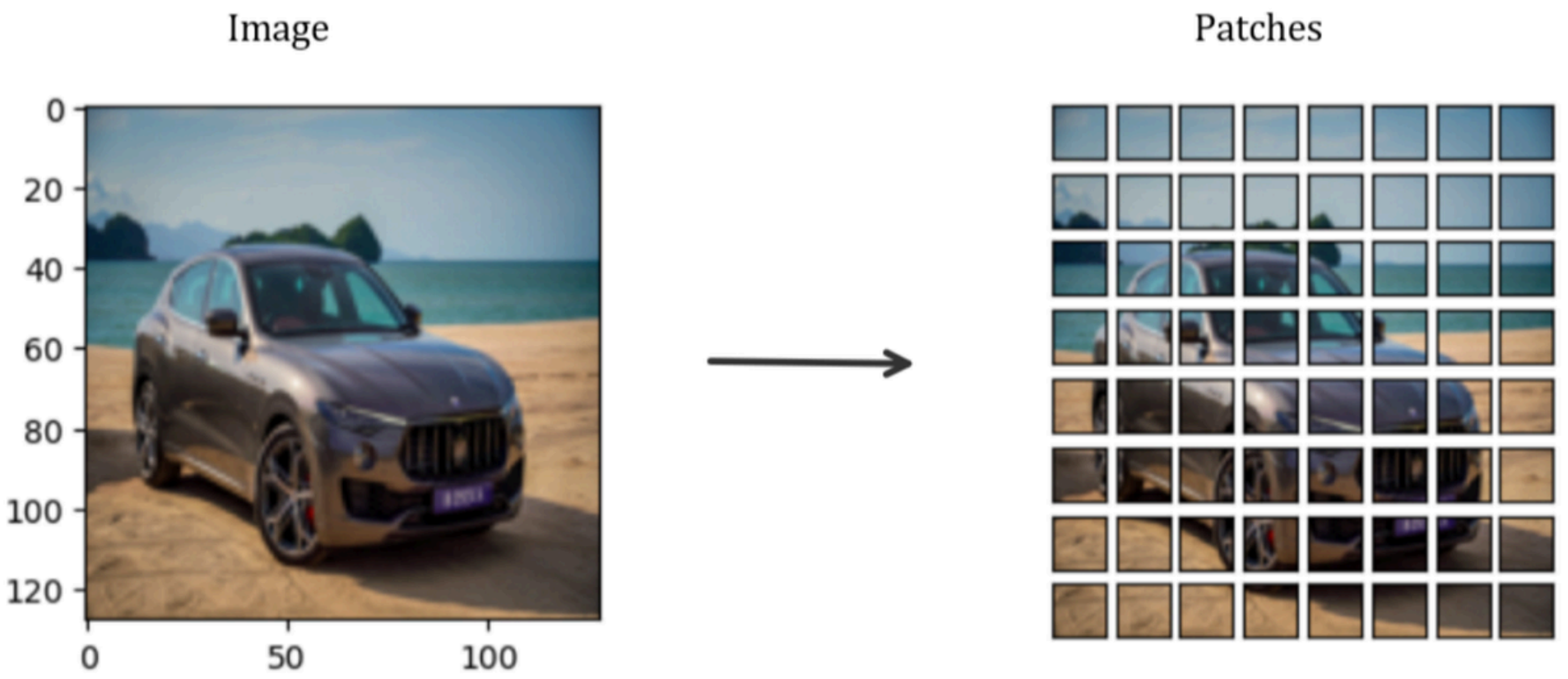
- However, unlike tokens, we cannot just assign each patch with an ID since these patches will rarely be found in other images, unlike the vocabulary of a text.

- Instead, the patches are linearly embedded to create numerical representations, namely embeddings. These can then be used as the input of a Transformer model.

- So with this approch, the moment the embeddings are passed to the encoder, they are treated as if they were textual tokens. From that point forward, there is no difference in how a text or image trains.



The main algorithm behind ViT. After patching the images and linearly projecting them, the patch embeddings are passed to the encoder and treated as if they were textual tokens.



128 x 128 resolution image to 64 number of 16 x 16 resolution patches

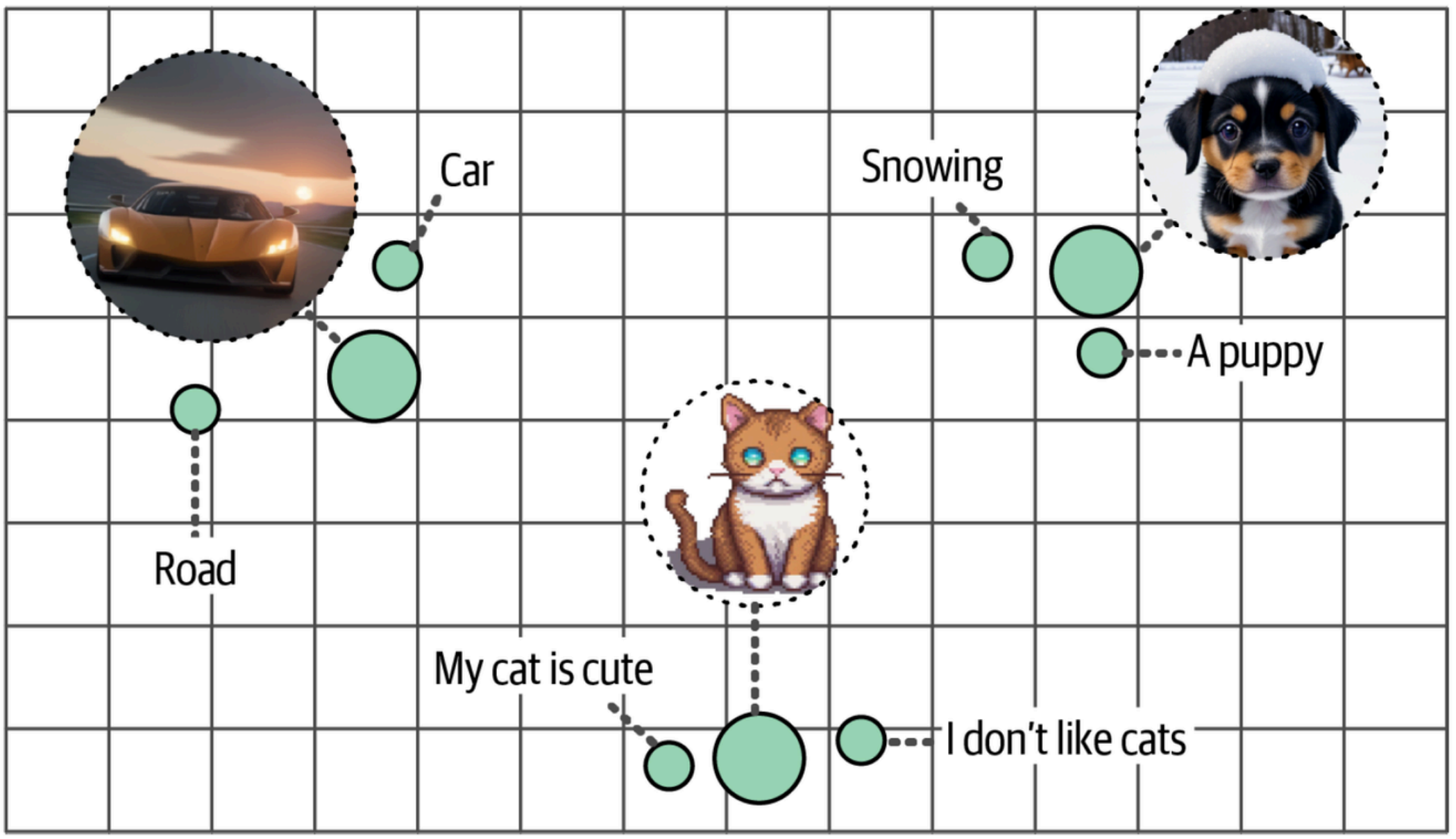
We order the sequence of patches from top left to bottom right. After ordering, we flatten these patches.

We get a linear sequence with flattening

Multimodal Embedding Models

- Multimodal embedding models are designed to capture and represent information from multiple data types, or "modalities," such as text and images.

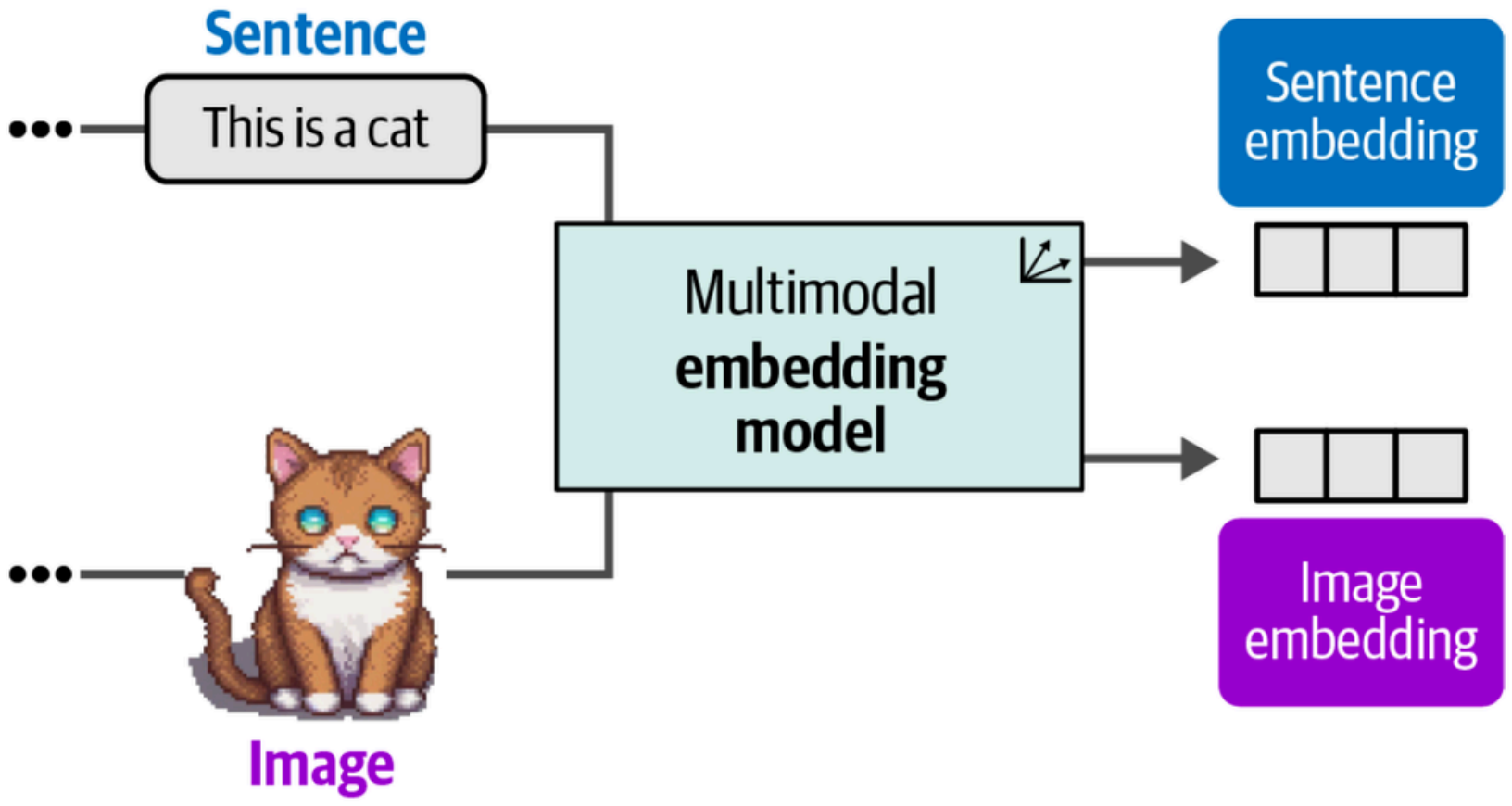
- While traditional embedding models focus on generating embeddings (numerical representations) for text alone, multimodal embeddings allow for the integration of various forms of data into a shared vector space.



Key Concepts

- **Unified Representation**: Multimodal embeddings enable us to map different types of data, like sentences and images, into the same vector space. This means that similar content, regardless of modality, can be placed closer together in this shared space. For example, the sentence "This is a cat" and an image of a cat can have embeddings that lie near each other in the vector space.

- **Cross-Modal Search and Retrieval**: By embedding both text and images in the same vector space, we can perform cross-modal searches. This allows us to search for images using text descriptions or find textual documents based on visual input. For instance, using a multimodal embedding model, we could search for "pictures of a puppy" and retrieve relevant images of puppies, or vice versa, find relevant text descriptions for a given image.

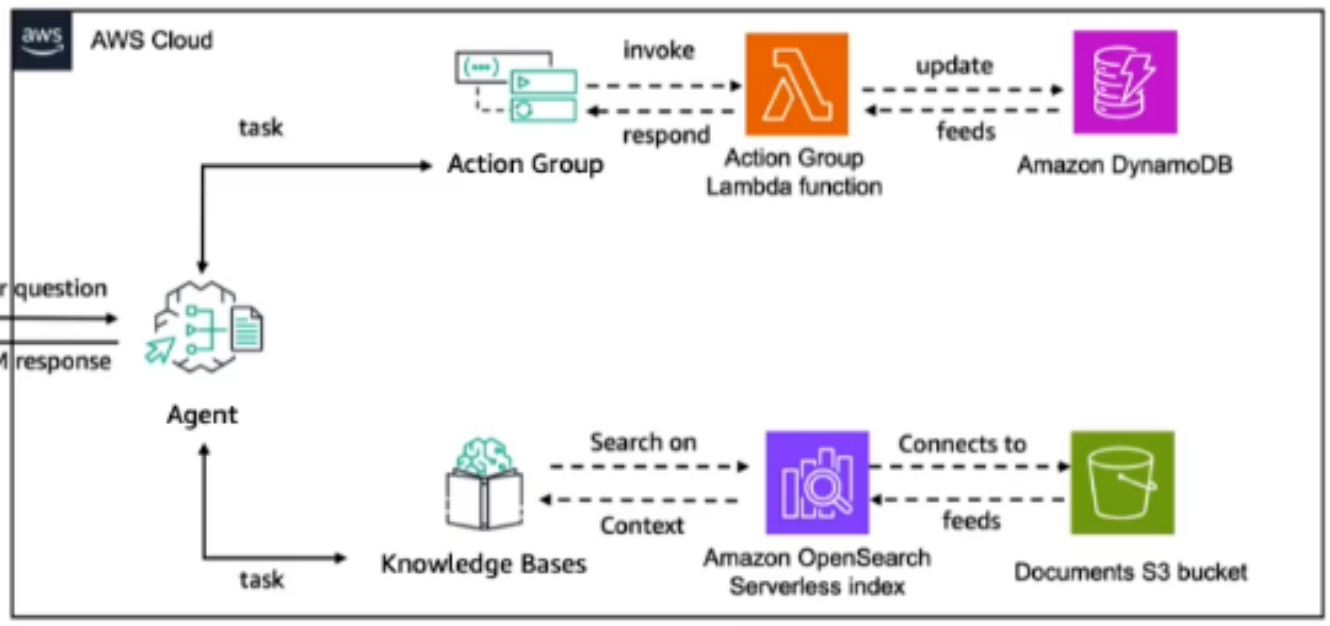


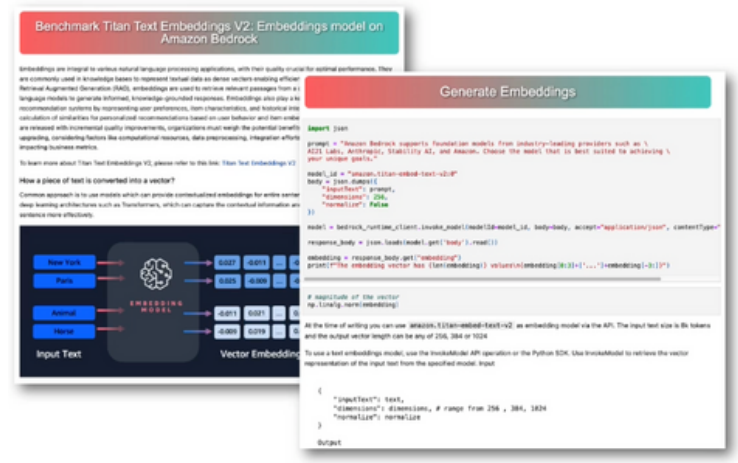
For more information, kindly visit the Free Course

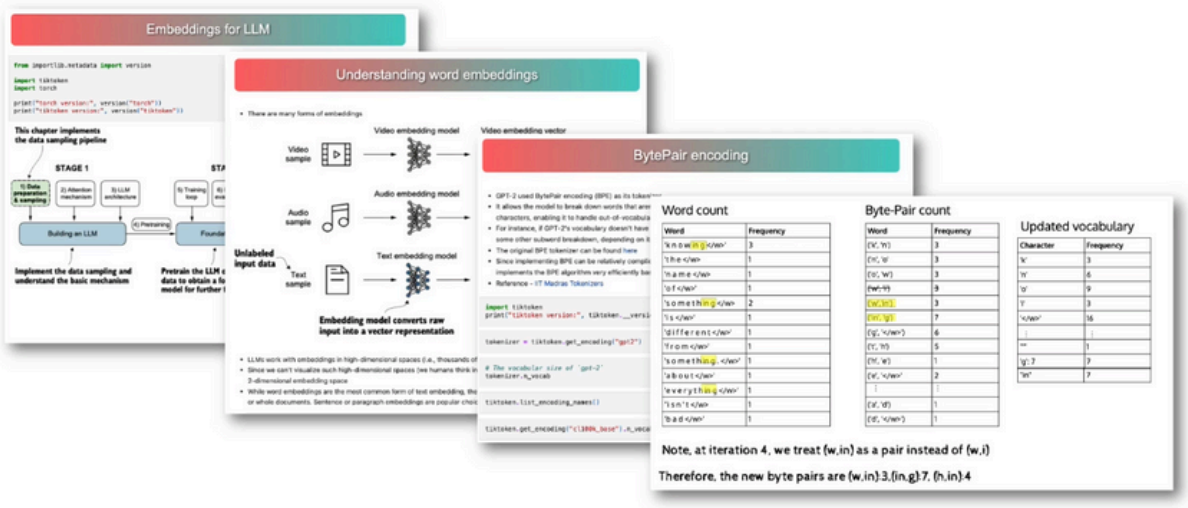
Analytics Vidhya

Free Course on

Mastering Multimodal RAG & Embeddings with Amazon Nova & Bedrock







Suman Debnath

Principal Developer Advocate (AI/ML)

aws

Enroll Now