Illustrative Session on Image Generative Models with Dall.E Mini

Karthik Desingu | Anirudh A | Karthik Raja A

Dept. of Computer Science, SSN College of Engineering

Session 3 *of the* Image and Video Analysis Workshop

International Conference on Computational Intelligence in Data Science, 2023

Agenda

- Overview of Dall.E Mini and its Building Blocks
- Brief Antedate of Autoencoders and GANs applied in Dall.E Mini
- BART Encoder-Decoder for Image-Text Latent Space Translation
- CLIP to Rank Generated Images by Relevance to Captions
- Piecing the Blocks of Dall.E Mini together

With Python Code Demo

Dall.E Mini — Text to Image

[Live Online Version of Dall.E Mini]

Part 1: Building Blocks of Dall.E Mini

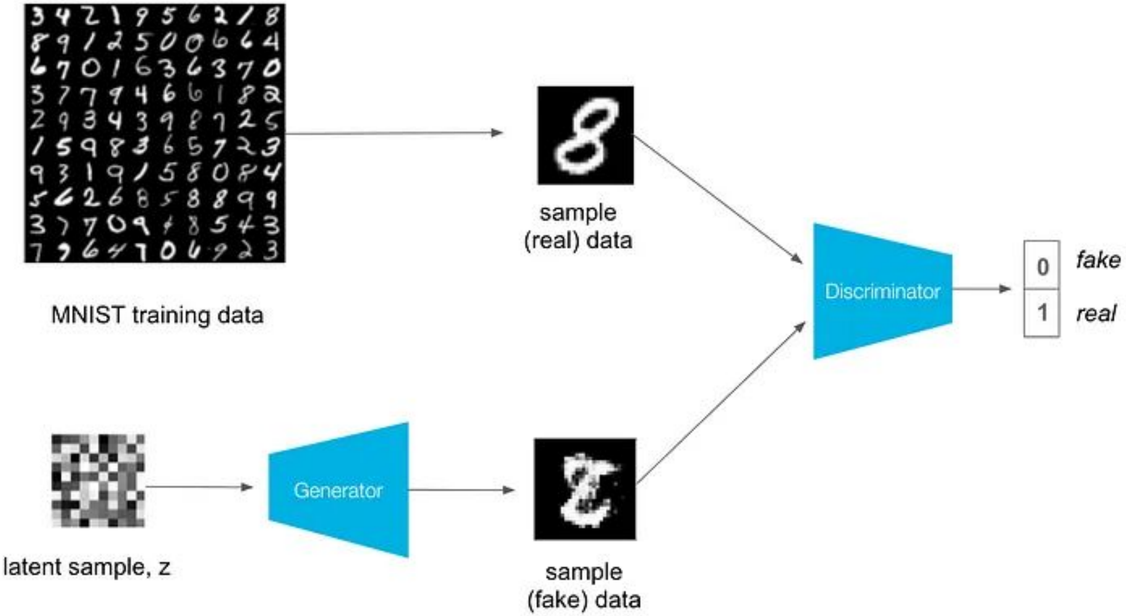
- **BART-based Encoder-Decoder**: Captions to Embeddings in the GAN's "vocabulary"
- **VQ-GAN**: Caption embeddings in latent space are translated into Images
- **CLIP**: Evaluates Caption-Image relevance

Part 2: Generative Adversarial Networks (GANs)

- Dall.E Mini uses a variant of GANs called **VQ-GANs**.
- The evolution of VQ-GANs,
  - Vanilla GAN
  - **Autoencoders** (AEs)
  - **Variational** Autoencoders (VAEs)
  - **Vector Quantized** Autoencoders (VQ-AEs)
  - Vector Quantized **GANs** (VQ-GANs)

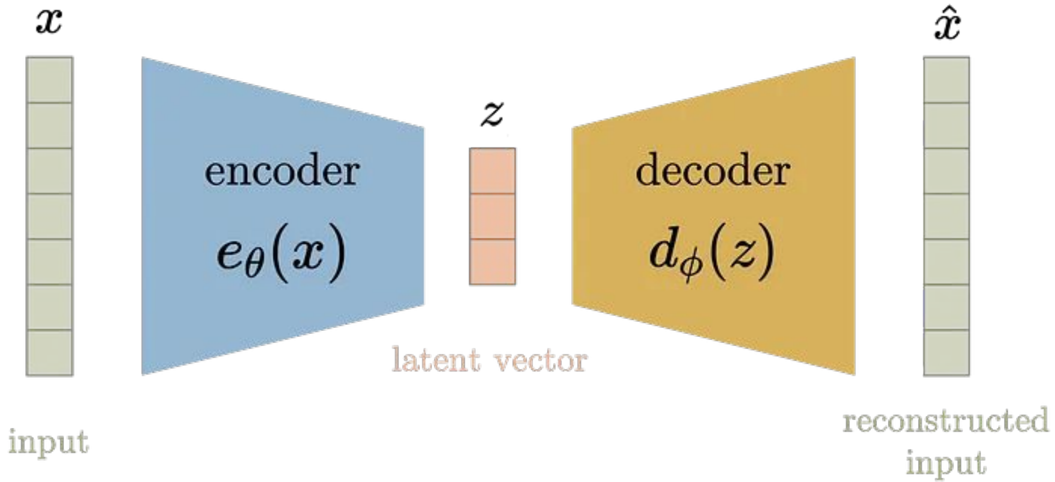
Python Code Demo

Vanilla GAN



Python Code Demo

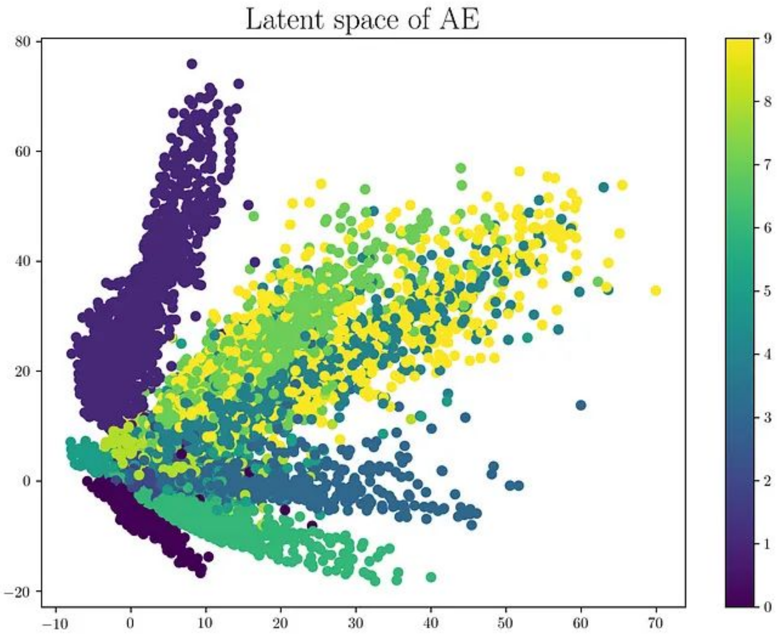
Autoencoder (AE)



$$loss = \|x - \hat{x}\|_2 = \|x - d_\phi(z)\|_2 = \|x - d_\phi(e_\theta(x))\|_2$$

- The latent space is discontinuous and has significant “gaps”.
- Consequently, meaningless images may be generated.

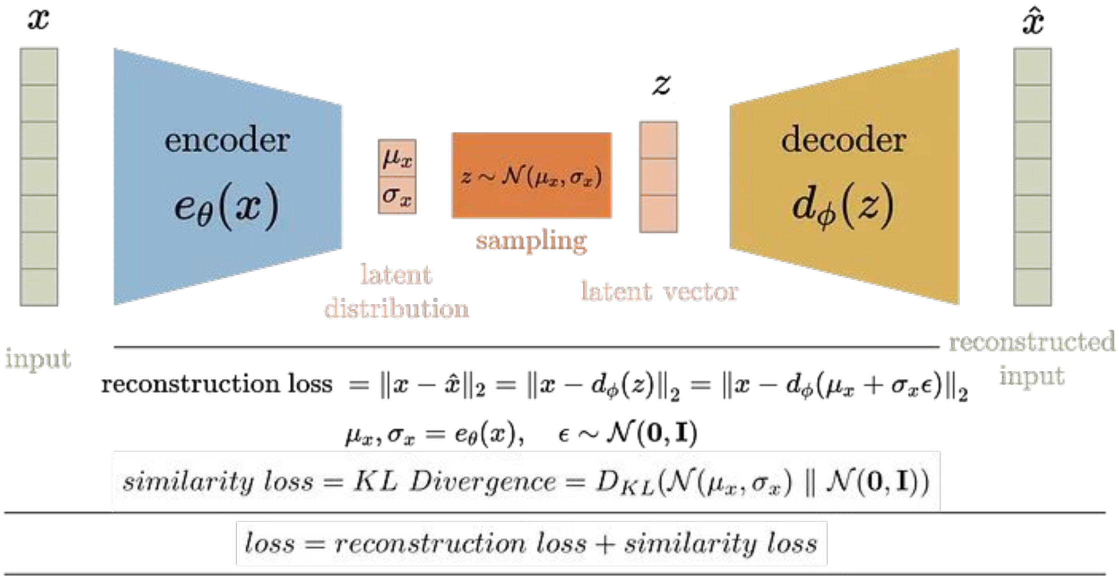
Autoencoder (AE)



- The latent space is discontinuous and has significant “gaps”.
- Consequently, meaningless images may be generated.

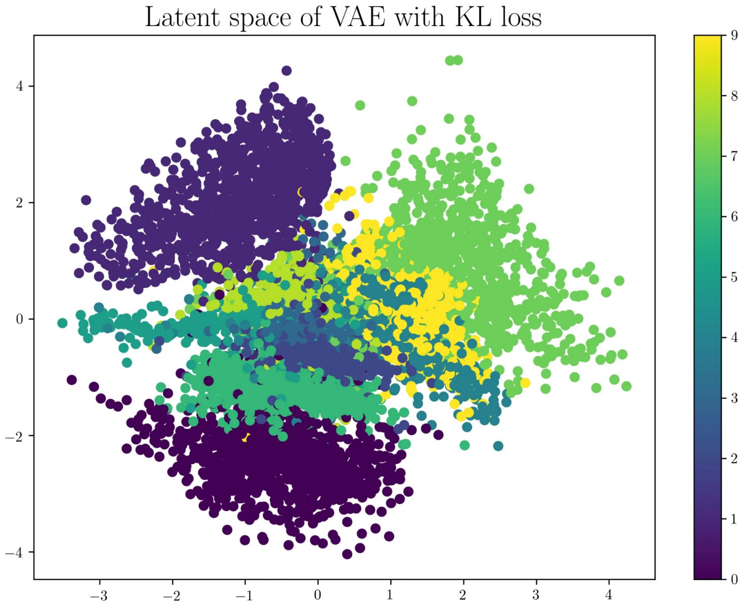
Python Code Demo

Variational Autoencoder (VAE)



- The latent space is more cohesive — resembles the unit norm.
- Overlapping regions produce “morphed” images.

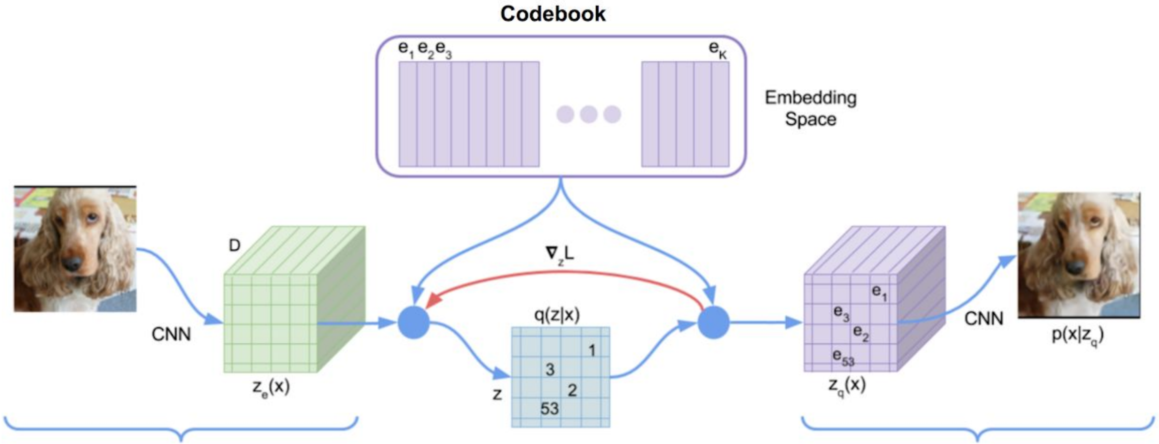
Variational Autoencoder (VAE)



- The latent space is more cohesive — resembles the unit norm.
- Overlapping regions produce “morphed” images.

Python Code Demo

Vector-Quantized Variational Autoencoder (VQ-VAE)



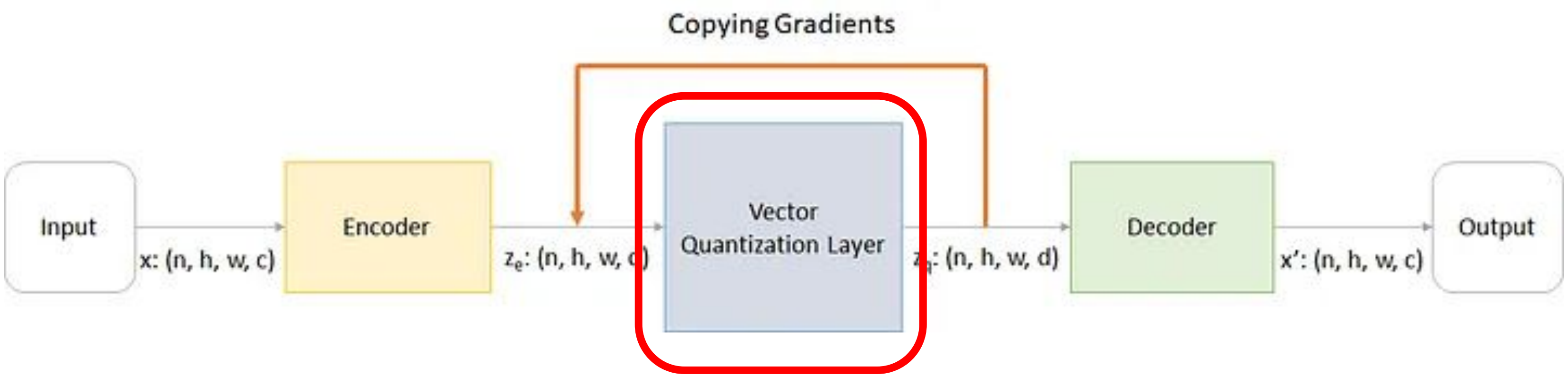
Posterior categorical distribution:

$$q(\mathbf{z} = \mathbf{e}_k|\mathbf{x}) = \begin{cases} 1 & \text{if } k = \arg\min_i \|\mathbf{z}_e(\mathbf{x}) - \mathbf{e}_i\|_2 \\ 0 & \text{otherwise.} \end{cases}$$

- The latent space is discrete.
- No “morphed” outputs.
- Latent space has same dimensions as codebook.
- Still does not model long-range interactions.

Python Code Demo

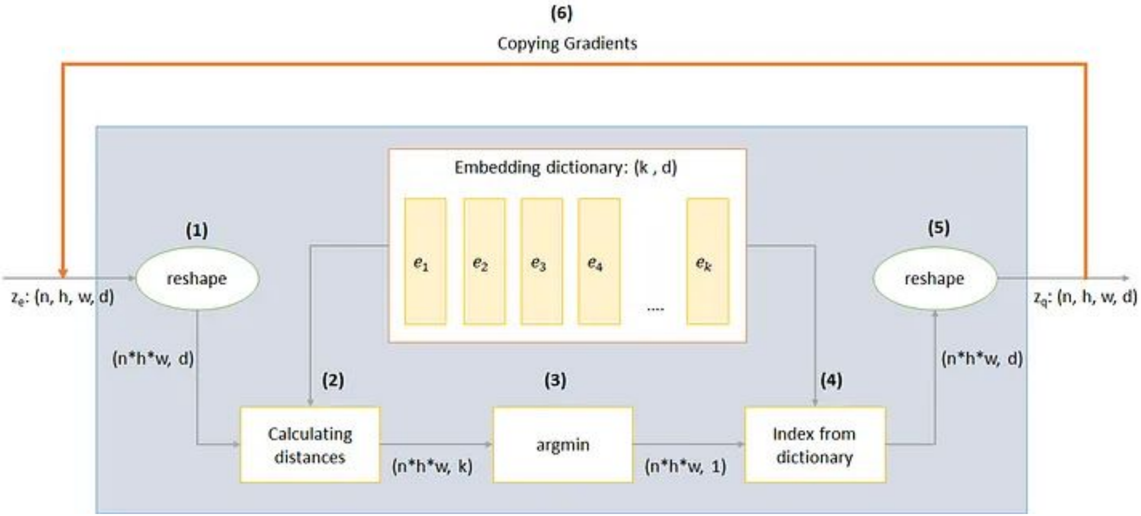
Vector-Quantized Variational Autoencoder (VQ-VAE)



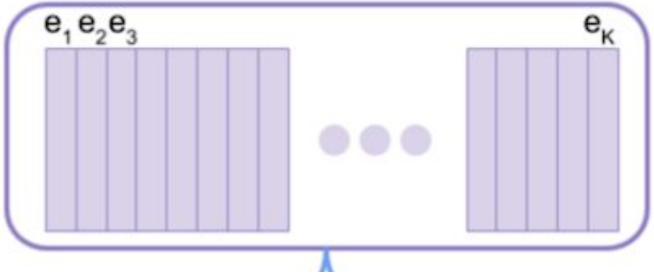
Adding the VQ layer to the AE

Python Code Demo

Vector Quantized Variational Autoencoder (VQ-VAE)

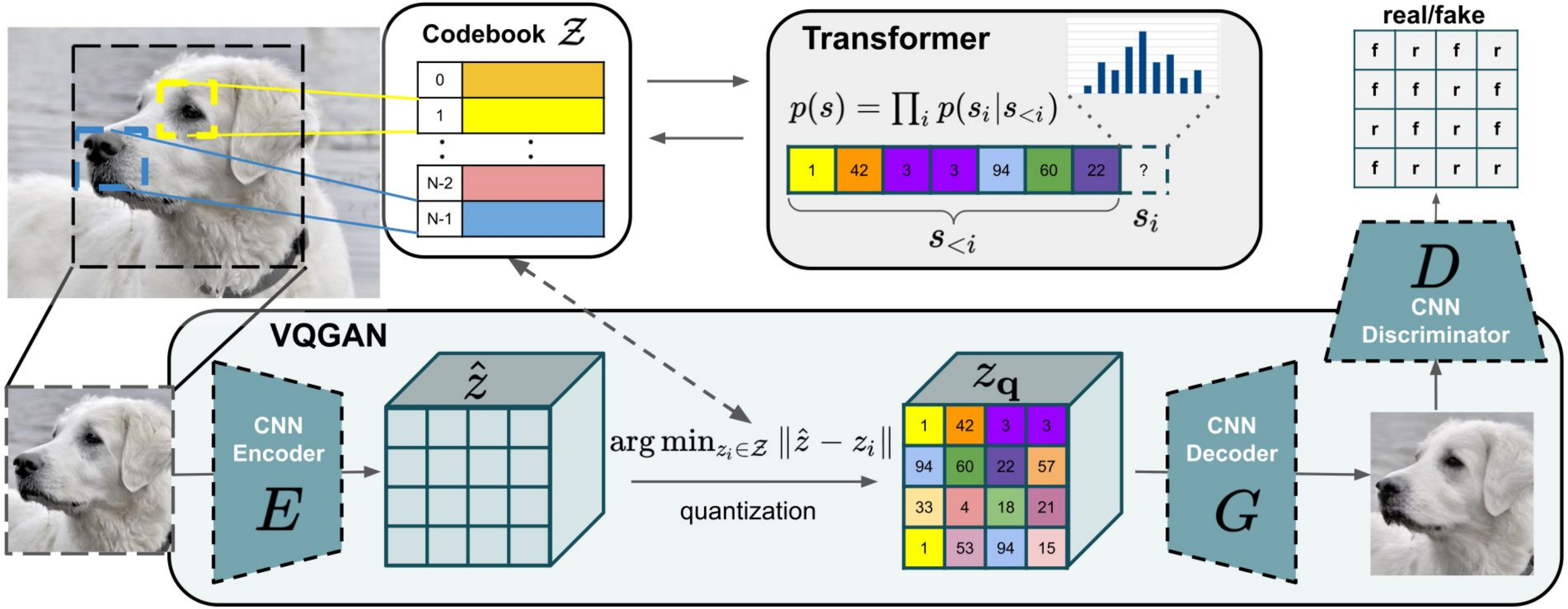


The Vector Quantization Layer

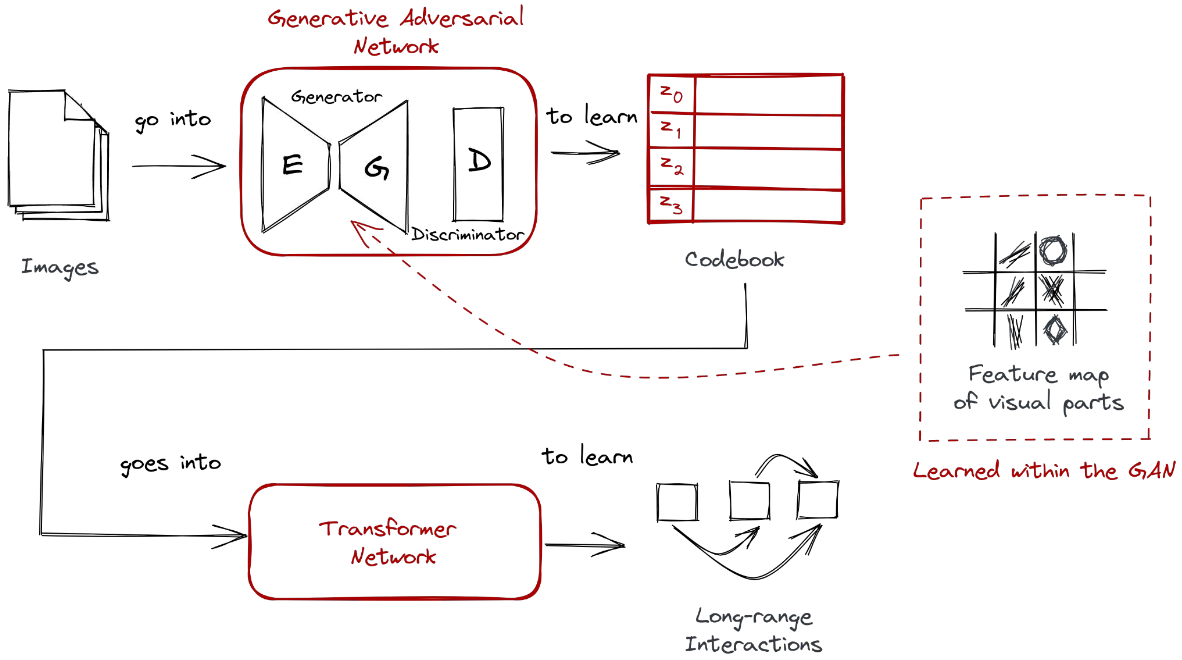


The Codebook

Part 5: Vector-Quantized GAN (VQ-GAN)

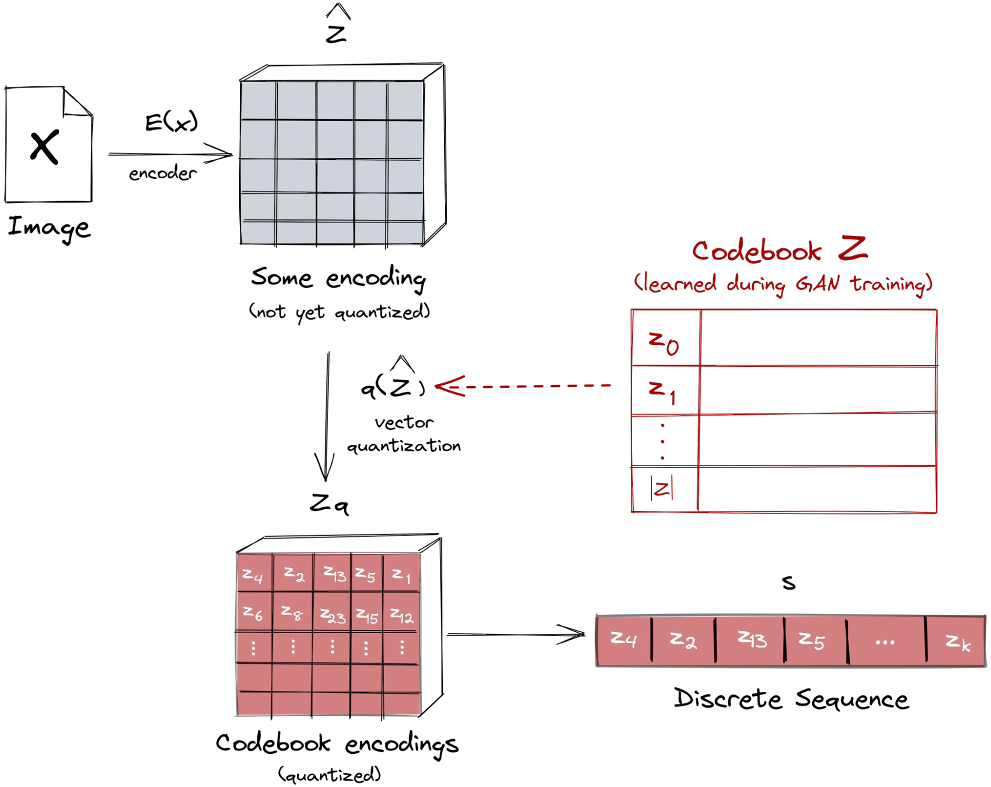


Vector-Quantized GAN (VQ-GAN)



- CNN-based VQ-VAE captures local relations well
- Transformer enhances long-range interactions

Training Objectives for the VQ-GAN



GAN Block

- Classical MiniMax Loss

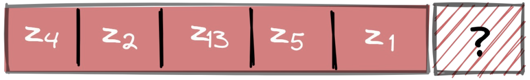
$L_{GAN}(G,D)=[\log D(x)+\log(1-D(\hat{x}))]$

VQ-VAE Block

- Codebook Loss
- Commitment Loss
- Reconstruction Loss

Transformer Block

- Negative log(p) of next element, given the sequence

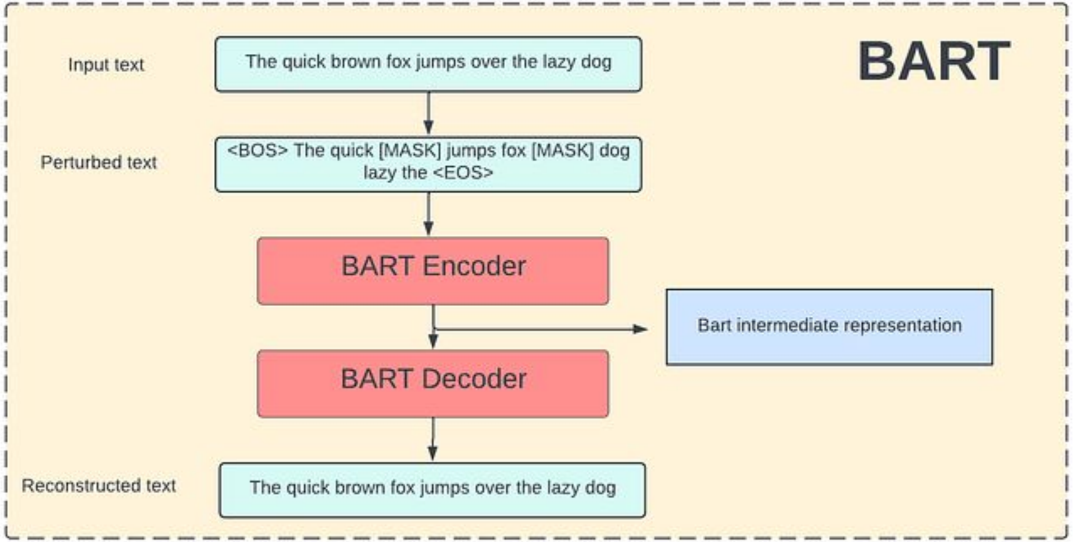


Part 3: BART Encoder-Decoder

- A BART model is pre-trained to “clean” text captions.
- For Dall.E Mini, the BART model **translates captions into the codebook vocabulary**.
- The codebook of VQ-GAN, in effect, maps text embeddings to image embeddings.

Python Code Demo

What BART does.



- For Dall.E Mini, it translates Text Captions to Embeddings in the Codebook Vocabulary.

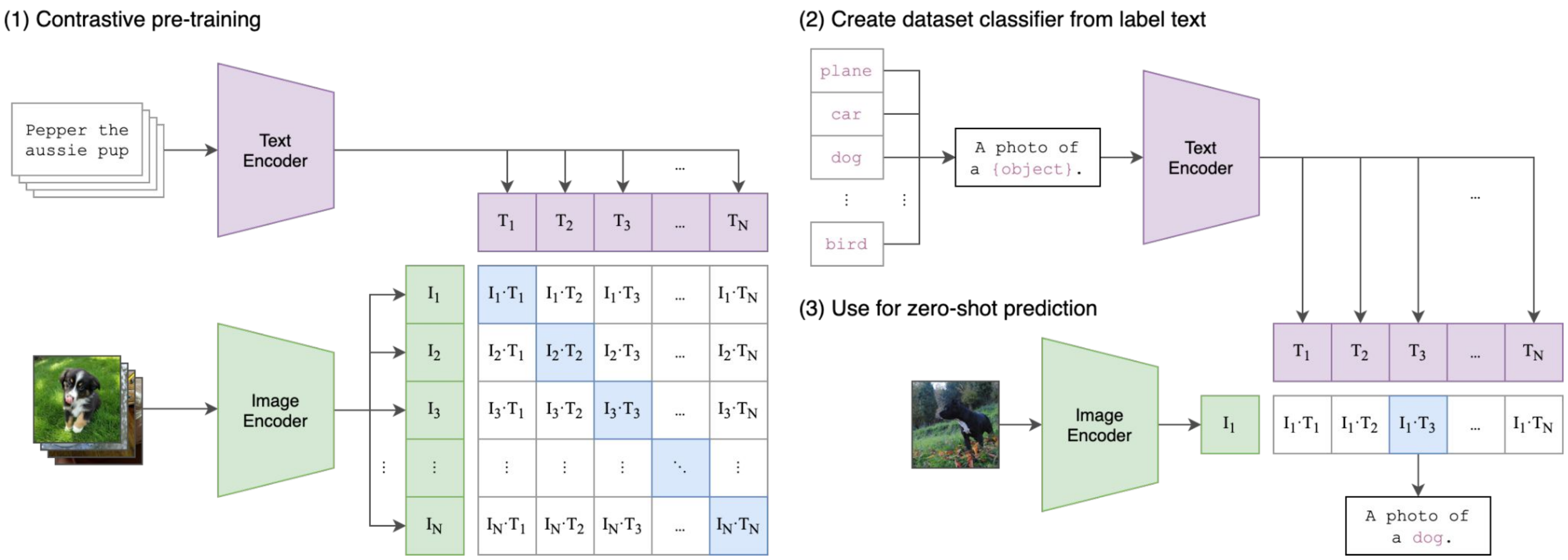
Part 4: CLIP to Rank Images by Relevance

Python Code Demo

- CLIP is a neural network trained on a variety of (image, text) pairs
- It can be instructed in natural language to predict the most relevant text snippet, given an image (and vice versa), without directly optimizing for the task
- CLIP is thus similar to the zero-shot capabilities of GPT-2 and 3
- CLIP matches the performance of the original ResNet50 on ImageNet "zero-shot" without using any of the original 1.28M labeled examples

CLIP Architecture

Python Code Demo



Contrastive pre-training is a type of self-supervised learning technique to learn representations of data that are useful for downstream tasks, such as image classification or natural language processing.

CLIP Glossary

We will be using OPENAI's CLIP library (https://github.com/openai/CLIP)

The CLIP module clip provides the following methods:

clip.available_models()

Returns the names of the available CLIP models.

clip.load(name, device=..., jit=False)

Returns the model and the TorchVision transform needed by the model, specified by the model name returned by clip.available_models().

clip.tokenize(text: Union[str, List[str]], context_length=77)

Returns a LongTensor containing tokenized sequences of given text input(s). This can be used as the input to the model.

CLIP Glossary

The model returned by **clip.load()** supports the following methods:

model.encode_image(image: Tensor)

Given a batch of images, returns the image features encoded by the vision portion of the CLIP model.

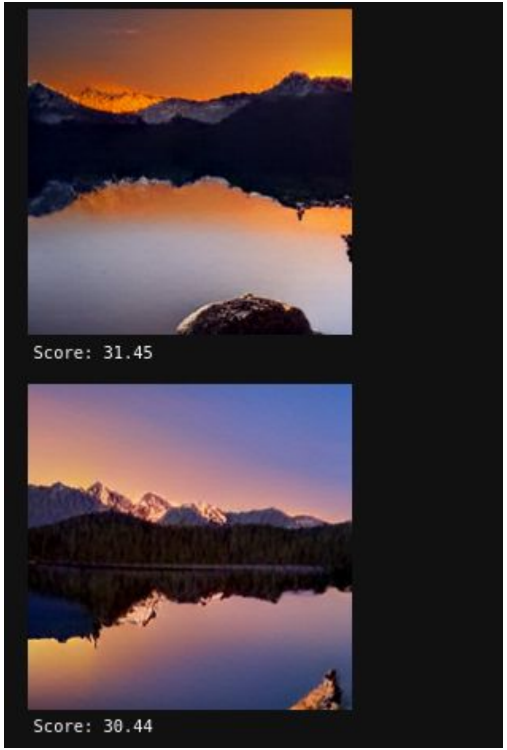
model.encode_text(text: Tensor)

Given a batch of text tokens, returns the text features encoded by the language portion of the CLIP model.

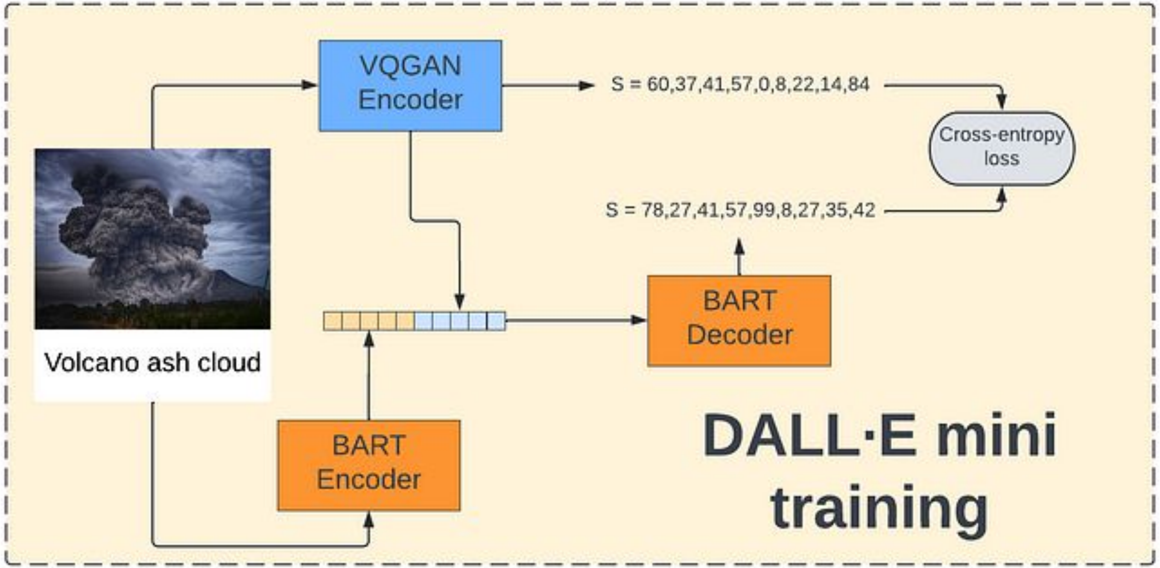
model(image: Tensor, text: Tensor)

Given a batch of images and a batch of text tokens, returns two Tensors, containing the logit scores corresponding to each image and text input. The values are cosine similarities between the corresponding image and text features, times 100.

Relevance Scores

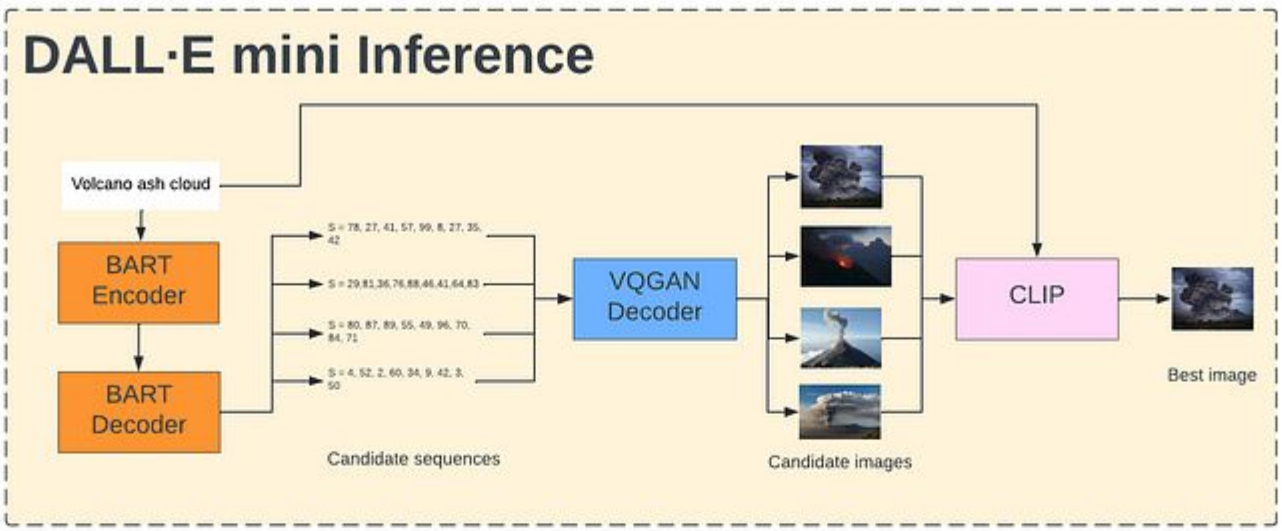


Part 5: Piecing the blocks together.



Python Code Demo

The Dall.E Mini Text-to-Image Pipeline.



Thank you for listening!

Questions?

Examples of Generated Images

Examples of Generated Images

Examples of Generated Images