Delhi Technological University

Deep Learning Project

Taming Transformers for high-resolution image synthesis

Taming Transformers for high resolution image synthesis (A VQGAN-CLIP-ESRGAN implementation)

by

Name	Roll no.	Email ID
Siddhant Bikram Shah	2K19/CO/374	siddhantbikramshah_2k19co374@dtu.ac.in
Ayush Karn	2K19/CO/454	ayushkarn_2k19co454@dtu.ac.in

A comprehensive project report has been submitted in partial fulfillment of the requirements for the degree of

Bachelor of Technology

in

Computer Engineering

Under the supervision of and submitted to

Mr. Kavinder Singh

Delhi Technological University, formerly

Delhi College of Engineering

Department of Computer Science Engineering

Delhi Technological University, Shahbad Daulatpur, Main Bawana Road, Delhi-110042, India

Abstract

In this project, we aim to build an image generation system with a single sentence. This project will be built using VQGAN + CLIP to handle the generation of images, along with ESRGAN for the sharpening of images.

This project is based on the work of **Katherine Crowson and Ryan Murdoch** (April 2021) who were able to combine the open-source model CLIP (from OpenAI) and other GAN architectures. Finally, we have modified this project by introducing ESRGAN in the mix which will help us in producing super-resolution images as the final output.

In simple words, VQGAN+CLIP(Vector Quantized Generative Adversarial Network and Contrastive Language–Image Pre-training), is a combination of deep learning models that can generate images based on a text prompt. Our project is trained on the dataset "**imagenet_16384**" which is an Image database organized in accordance with the WordNet hierarchy.

Introduction

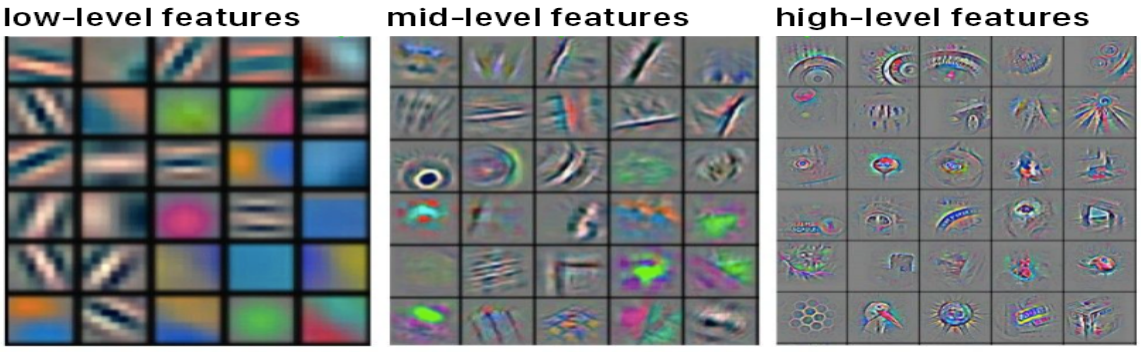
How we perceive images: the Theory of Perception

In accordance with this theory, human visual reasoning is symbolic, and we assign meaning to objects using modalities and discrete representations. Using this symbolic technique, we may understand relationships between different words or symbols.

This is referred to as the ability to model "long-range dependencies" in machine learning. We recognised that looking up is the cat's behaviour and white is the cat's description in the image above "a white cat gazing up."

Even though there is lots of study into employing discrete representations to explore complicated reasoning, most computer vision (CV) techniques ignore modalities. Instead, they think with regard to pixels.

How Most CV Technique process images

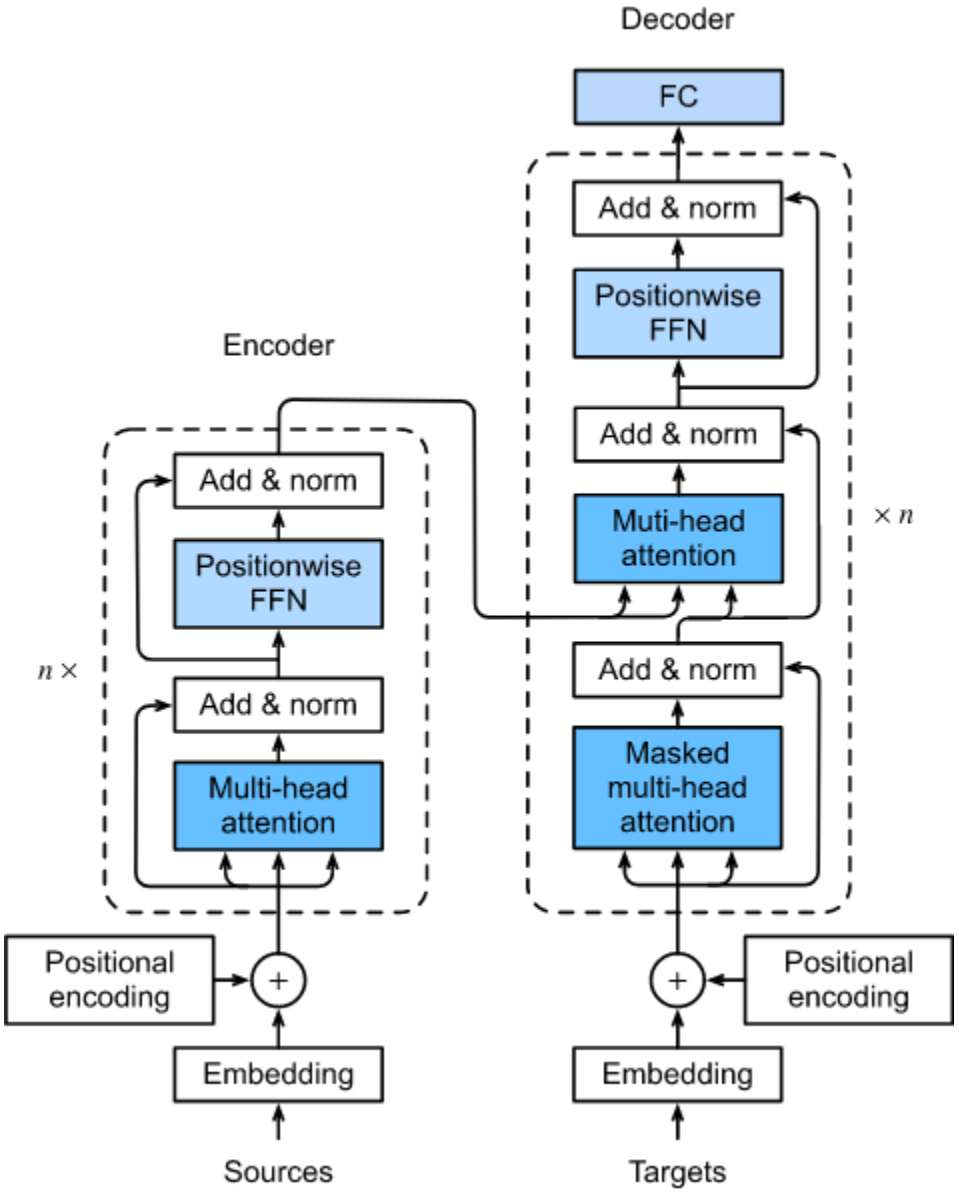


Rather than thinking in terms of huge chunks of data, most Computer Vision approaches focus on the tiniest unit of an image called a pixel. On a continuous scale, each channel represents a colour value(Red, Green, Blue).

Convolutional neural networks (CNN) demonstrated that by confining interactions inside their local neighbourhood, we can describe local interactions between pixels (the kernel). This helps us to finally "assemble" pixels & understand visual parts.

CNNs figured out how to put pixels together at several levels of abstractions: pixels become edges, edges become shapes, and shapes become components.

Transformers: An Overview



Transformers are a type of neural network design that has gained a lot of traction recently. Transformers were recently used by OpenAI in their language models, and DeepMind used them in AlphaStar, a machine that defeated a top professional Starcraft player.

Transformers were developed in response to the difficulty of neural machine translation, also known as sequence transduction. Any task that translates a

sequence of inputs into a sequence of outputs qualifies. Speech recognition and text-to-speech conversion are examples of this.

Some of their interesting properties are:

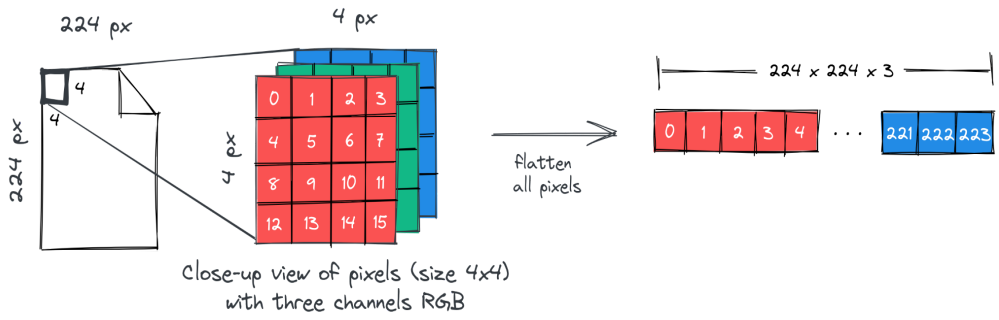
1. Employs Encoder-Decoder(or Sequence-to-Sequence) structure.
2. Divides images into patches similar to tokens in NLP.
3. Strong long-range modelling capabilities

Some of their salient features include:

1. Understand faraway sequential data
2. Fast processing
3. Adaptable to any kind of sequential data

This is why they are widely applied in Natural Language Processing and Computer Vision tasks.

Use of Transformers to Model Interaction

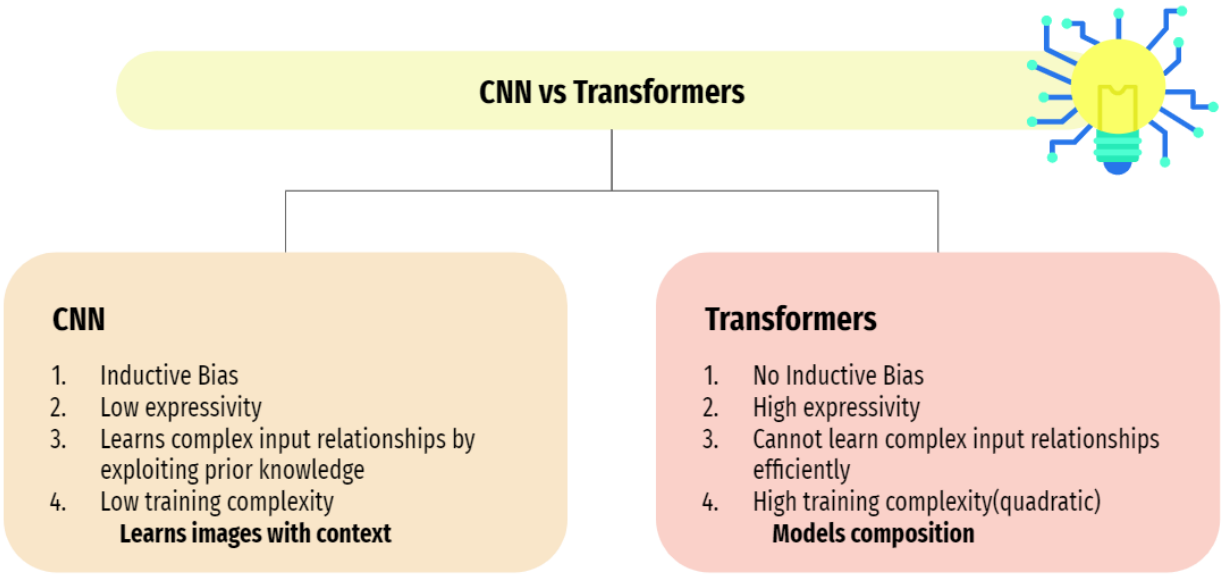


Transformer networks develop long-range dependencies among pictures using a discrete technique. VQGAN combines a continuous approach that uses a CNN to learn visual parts with the transformer's approach.

The transformer network, however, has a limitation: its calculation scales quadratically with the dimensions of the image. A 224x224-pixel image will have a length of 224*224*3, which is far more than a standard GPU can handle.

A similar two-stage structure is used by VQGAN, which learns an intermediate representation of the image. Then, it is passed to a transformer. VQGAN represents visual portions with a codebook rather than downsampling the sample. The authors did not directly model the image at the pixel level, but rather from the learned codebook's codewords.

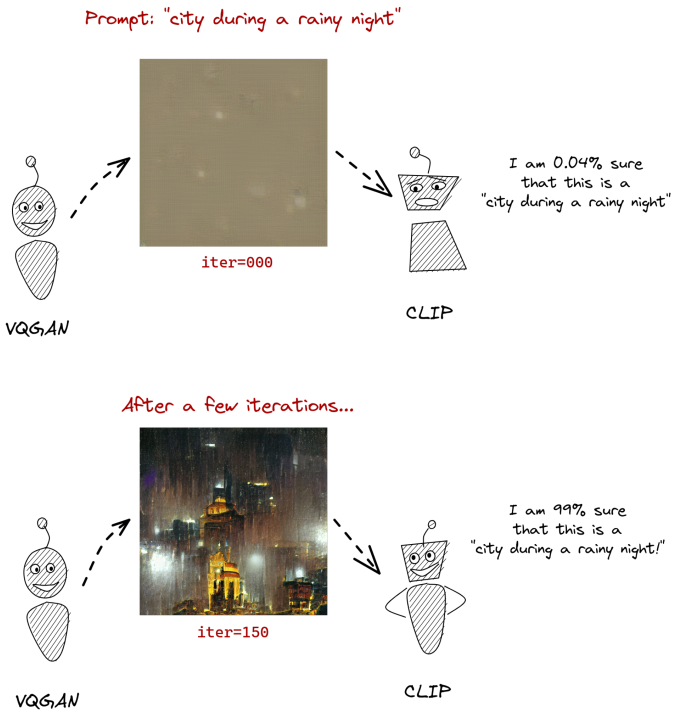
Motivation



When it comes to Computer Vision jobs, transformers are frequently used and investigated over typical CNN approaches for the reasons stated above.

System Overview

VQGAN



VQGAN stands for Vector Quantized Generative Adversarial Network, and CLIP stands for Contrastive Image-Language Pretraining. The interaction between these two networks is referred to as VQGAN-CLIP. They're two different models that function together.

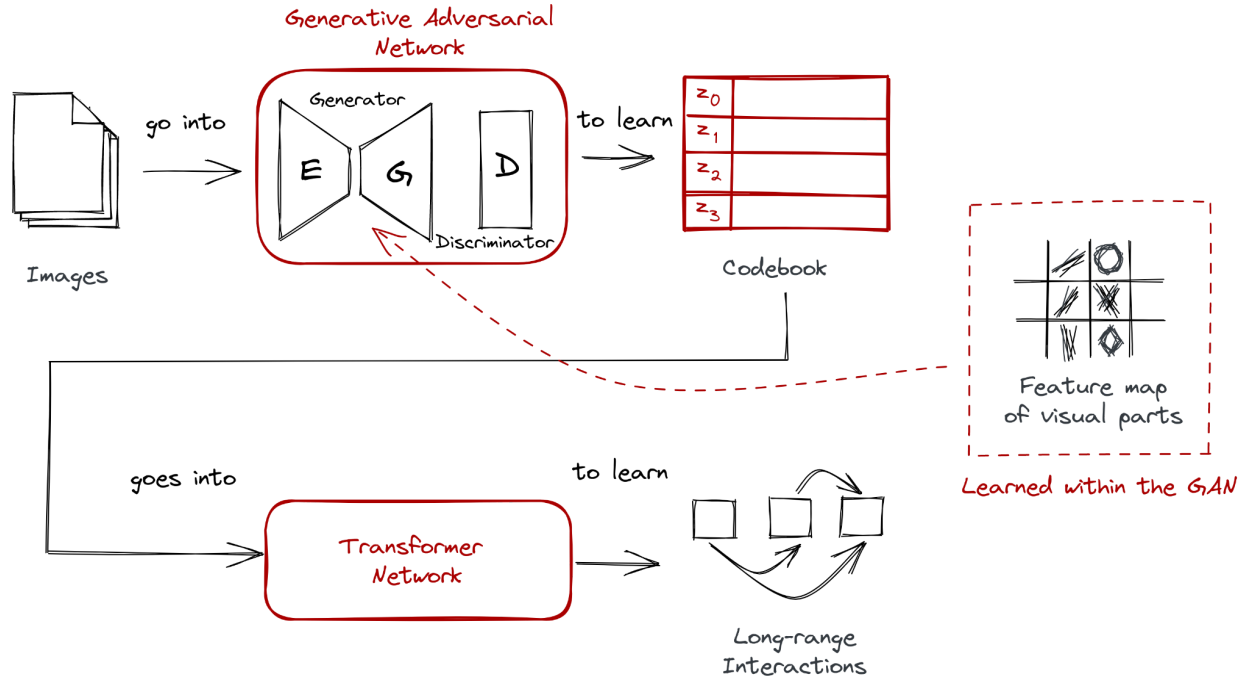
VQGAN creates the pictures, whereas CLIP assesses how well an image fits our text query. Our generator is guided by this interaction to create more accurate images.

VQGAN was able to bring them together. It can figure out:

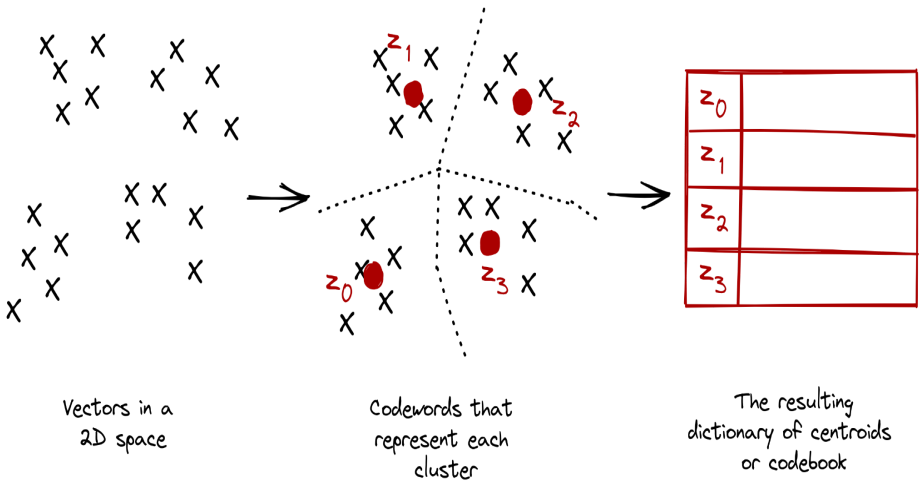
(1) the visual components of an image.

(2) the relationships between them.

What is a Codebook?



A procedure known as vector quantization is used to create the codebook (VQ). Specifically, the "VQ" portion of "VQGAN." When sent to a transformer network, it denotes all visual elements detected in the convolutional section in a quantized form, so that it require less computation capability.



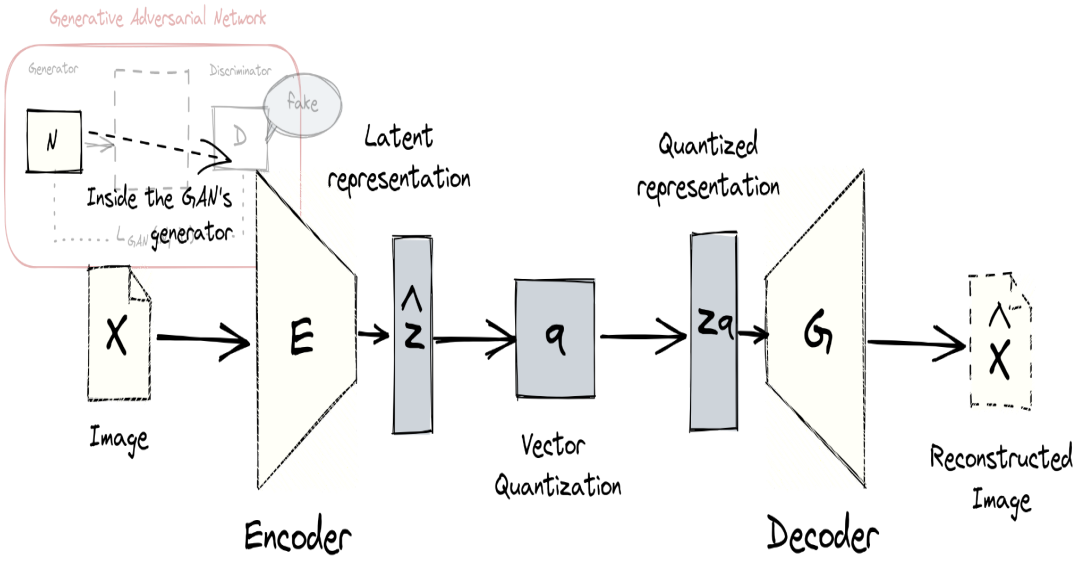
Vector quantization can be defined as the process of splitting vectors into groups with about the same number of points nearest to them. It's a vector encoding signal processing approach. The centroid (codeword) for each group is then determined using k-means or other clustering algorithms. There are two parts in Vector Quantization i.e encoding and decoding part. First, we have to create a codebook based on the input thatw e are taking and each codebook has a index assigned to it which is in binary form. The decoder also has the same code book and indexes in them. When an input is given to the encoder. It encodes it with some distortion formula and findes the code vector that is closested to it and finally the index corresponding to it is passed to the decoder which then finds that index using table lookup which is present in the decoder. Finally, a dictionary of centroids (codebook) and their members are learned and a reconstructed image is generated with it.

Training the GAN

$$LGAN(N,D)=[logD(x)+log(1-D(\hat{x}))]$$

The constituent expressions in this equation are described below:

1. **Generator N** - used for developing new samples
2. **Discriminator D** categorizes the samples as fake or real
3. **logD(x)** determines the likelihood of D correctly predicting whether a real image(from dataset) x is real.
4. **log(1-D(^x))** determines the likelihood of D correctly predicting whether a fake(generated) image x is real.



Between the decoder and the encoder, vector quantization is done. After encoding is done on the input x into z=E(X), we execute an element-wise q operation to get a discrete version of the given input:

$$z_q = \mathbf{q}(\hat{z}) := \arg\min_{z_k \in Z} ||\hat{z}_{ij} - z_k||$$

To acquire the reconstructed picture, instead of reconstructing the image using the output z from the encoder, we reconstruct it from its quantized form (zq):

$$\hat{x} = G(z_\mathbf{q}) = G(\mathbf{q}(E(x)))$$

The codebook and the autoencoder model were trained in tandem with each other using the loss function:

$$\mathcal{L}_{VQ}(E, G, Z) = ||x - \hat{x}||^2 + ||sg[E(x)] - z_\mathbf{q}||_2^2 + ||sg[z_\mathbf{q}] - E(x)||_2^2$$

Loss Function

$$L_{VQ}(E,G,Z) = \|x-\hat{x}\|^2 + \|sg[E(x)]-z_q\|^2_2 + \|sg[z_q]-E(x)\|^2_2$$

The constituent expressions in this equation are described below:

$$\|x-\hat{x}\|^2$$

1. **Reconstruction loss** evaluates how well our network approximated (through x) our input x when only given its quantized version zq. This is calculated in terms of perceptual loss rather than per-pixel loss.

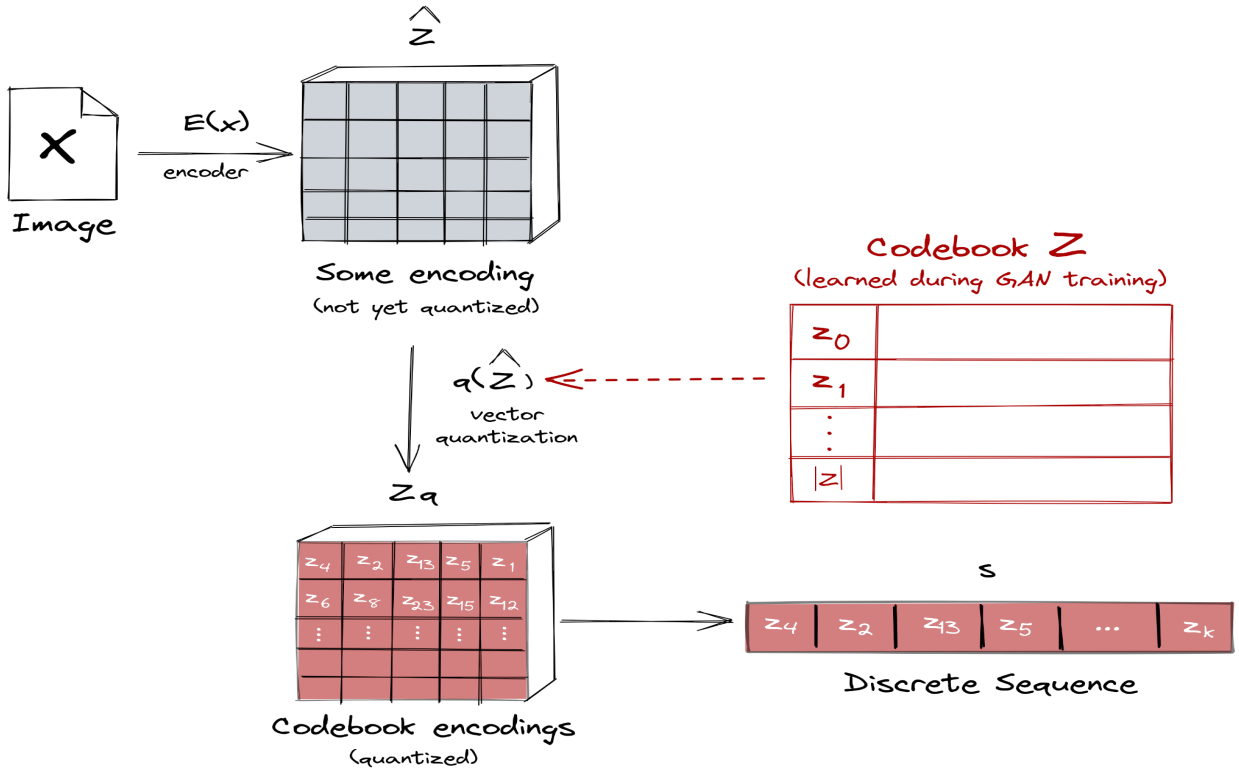
$$\|sg[E(x)]-z_q\|^2_2$$

2. Our embeddings are optimised through the above expression. "**Stop-gradient**" is the operation sg. It's an identity function with zero partial derivatives in forward pass, limiting it to be constant.

$$\|sg[z_q]-E(x)\|^2_2$$

3. It's referred to as the **Commitment Loss**. It ensures that the encoder E is dedicated to a certain picture representation.

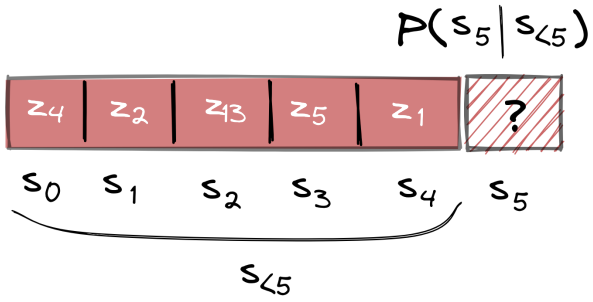
Training the Transformer



Our visual pieces are represented discretely in Codebook Z, which may be easily input into a transformer.

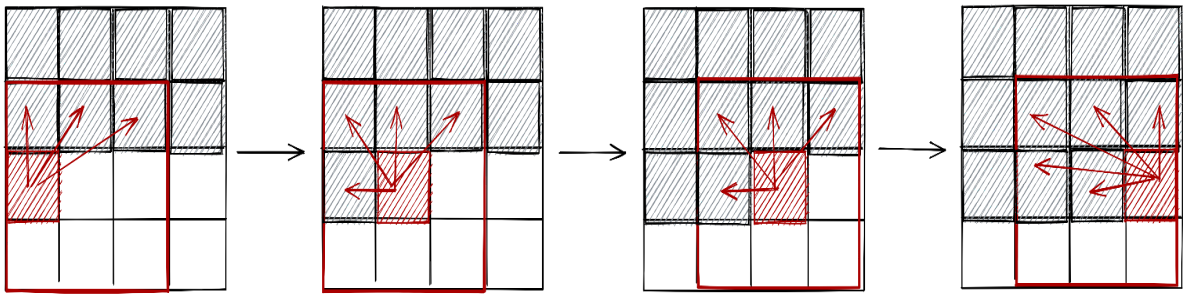
The encoding zq of picture x is denoted codebook Z's sequence s, that is, images are represented in terms of the codebook-indices of their embedding:

$s \in \{0,\ldots..|Z|-1\}$



In the case of s<i, the transformercan predict the future indices' distribution.

p(si | s<i)



We can find the likelihood as:

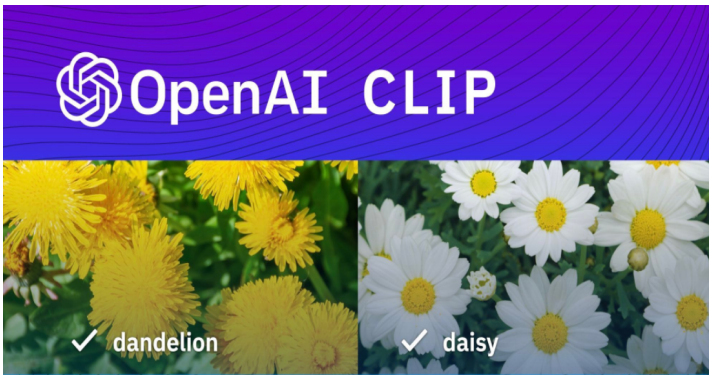
$$p(s)=\Pi ip(si \mid s<i)$$

and minimize the transformer loss.

$$LTransformer=Ex\sim p(x)[-logp(s)]$$

When generating high-res images, we restrict context via a sliding window which helps in improving resource-efficiency.

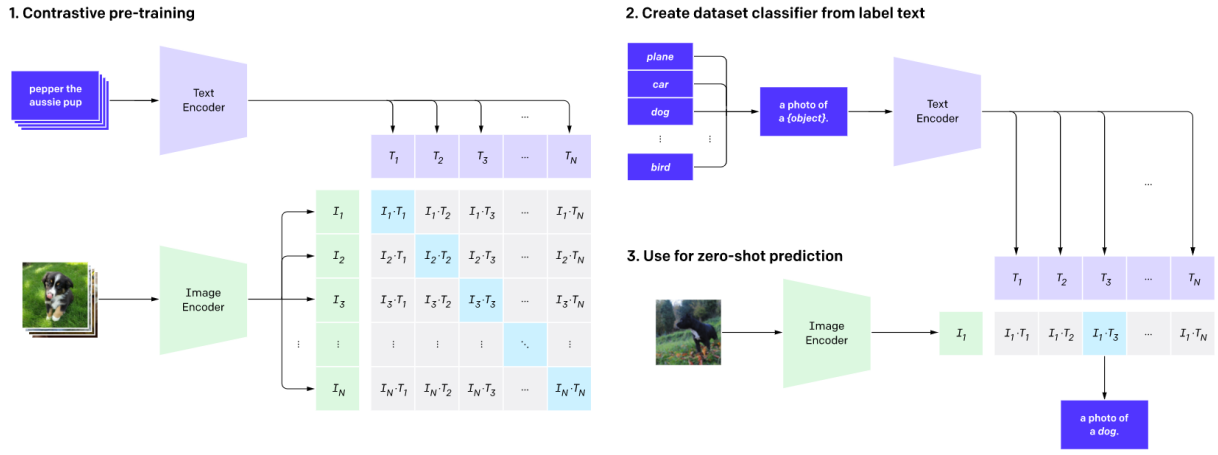
CLIP



Contrastive Language-Image Pretraining (CLIP) is an acronym for Contrastive Language-Image Pretraining. It builds on previous research in the areas of zero-shot transfer, natural language supervision, and multimodal learning.

CLIP models must learn to recognise and correlate a large range of visual concepts in photos with their names. As a result, CLIP models can be used to solve practically any visual categorization problem.

How CLIP works

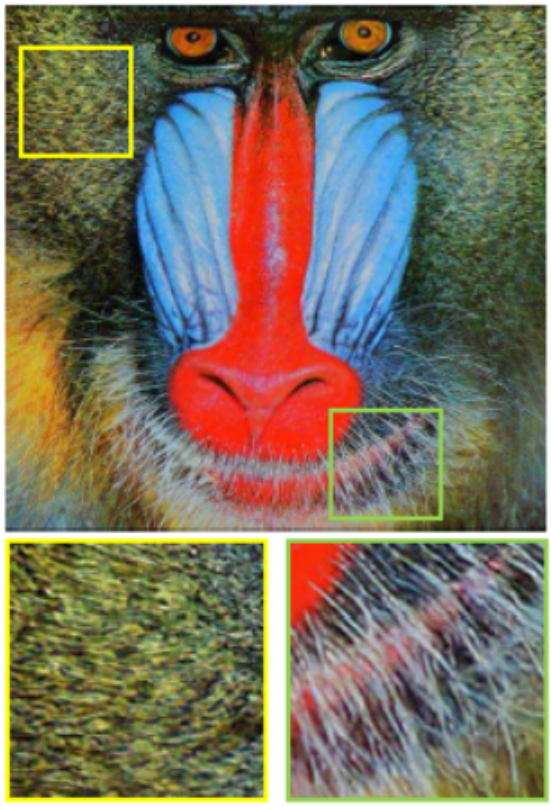


CLIP uses a pre-trained image and text encoder to predict which photos and sentences were associated in our dataset. This training then turns CLIP into a zero-shot classifier. Then, turn all of a dataset's classes into captions like "a snapshot of a dog" and forecast which caption class CLIP thinks best matches a given image.

CLIP was created to address many key flaws in the traditional deep learning approach to computer vision:

1. Expensive Datasets
2. The narrowness of models
3. Poor real-world performance

ESRGAN

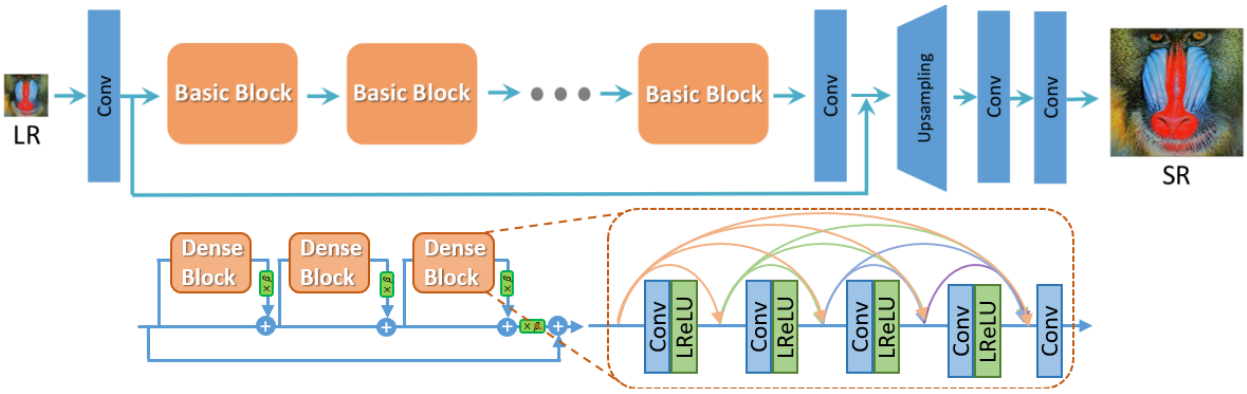


ESRGAN

ESRGAN is an acronym that stands for "Enhanced Super-Resolution Generative Adversarial Network." Image super-resolution techniques generate a high-resolution image from a low-resolution image, such as upscaling a 720p image into a 1080p image.

The ESRGAN employs a Relativistic discriminator to more accurately approximate the likelihood of an image being real or false, resulting in superior outcomes.

How ESRGAN works



In addition to the usual discriminator, ESRGAN uses a relativistic GAN to predict the likelihood that a genuine image is more realistic than a fake. The discriminator loss and the adversarial loss are then defined as follows:

$$L_D^{Ra} = -\mathbb{E}_{x_r}[\log(D_{Ra}(x_r, x_f))] - \mathbb{E}_{x_f}[\log(1 - D_{Ra}(x_f, x_r))].$$

$$L_G^{Ra} = -\mathbb{E}_{x_r}[\log(1 - D_{Ra}(x_r, x_f))] - \mathbb{E}_{x_f}[\log(D_{Ra}(x_f, x_r))],$$

Result

Some of the art generated by the VQGAN+CLIP system is highlighted below. On the left, we have the image created by VQGAN-CLIP, and on the right, we have the final image we obtain after super-resolution is done by ESRGAN.

Prompt: Hogwarts Castle Pencil Sketch

496x496 px

1984x1984 px

Prompt: Himalayas Cloud Sky Heavenly

496x496 px

1984x1984 px

Prompt: Snowy Mountains Sunset Pencil Sketch

496x496 px

1984x1984 px

Conclusion and Future Scope

VQGAN-CLIP, a method of generating and manipulating high-resolution images based on only human written text prompts, has been presented in this project.

Compared to its competitors like BigGan, VQGAN-CLIP performs well even when the image content and text prompt have a low semantic similarity.

Research on the use of augmented images during optimization proves that it is faster and shows more quality than the traditional codebook method.

References

1. Esser, Patrick, Robin Rombach, and Bjorn Ommer. "Taming transformers for high-resolution image synthesis." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021.

2. Li, Yangguang, et al. "Supervision exists everywhere: A data efficient contrastive language-image pre-training paradigm." arXiv preprint arXiv:2110.05208 (2021).

3. Wang, Xintao, et al. "Esrgan: Enhanced super-resolution generative adversarial networks." Proceedings of the European conference on computer vision (ECCV) workshops. 2018.

4. “Illustrated VQGAN” https://ljvmiranda921.github.io/notebook/2021/08/08/clip-vqgan/

5. “Guide To Image Super-Resolution By ESRGAN” https://analyticsindiamag.com/guide-to-image-super-resolution-by-esrgan/

6. “CLIP: Connecting text and images” https://openai.com/blog/clip/