

# **Text to Image Generator**

---

**MNIT Jaipur**

By: Abhishek Rai

# INTRODUCTION

## Stable-diffusion-2 Model

- Model trained on **LAION-5B** database
- Achieved by encoding text inputs into latent vectors using pretrained language models like **CLIP**.
- Applying **diffusion** process over a lower dimensional latent space, instead of using the actual pixel space.

# DIFFUSION

- FORWARD DIFFUSION

- Drop of ink fell into a glass of water
- Ink drop **diffuses** in water
- You can no longer tell whether it initially fell



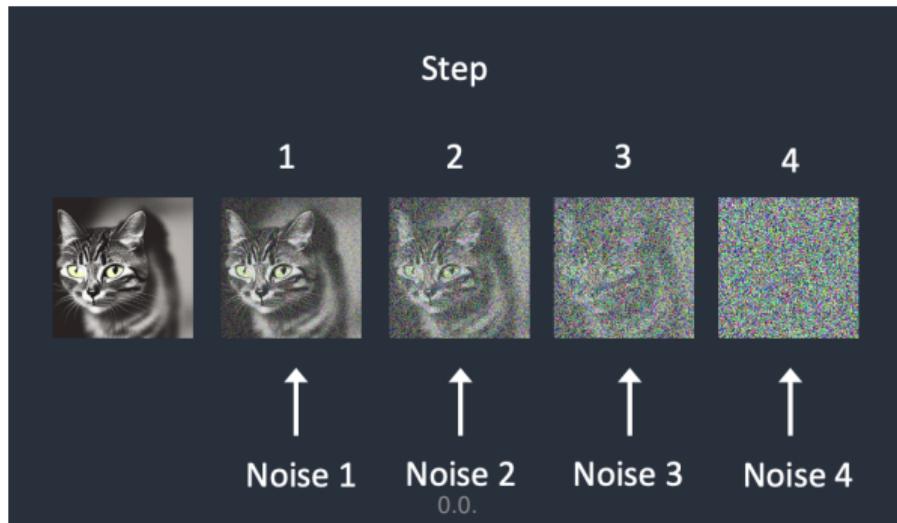
- BACKWARD DIFFUSION

- Going backward in time
- We will see where the ink drop was initially added

# DIFFUSION

- FORWARD DIFFUSION

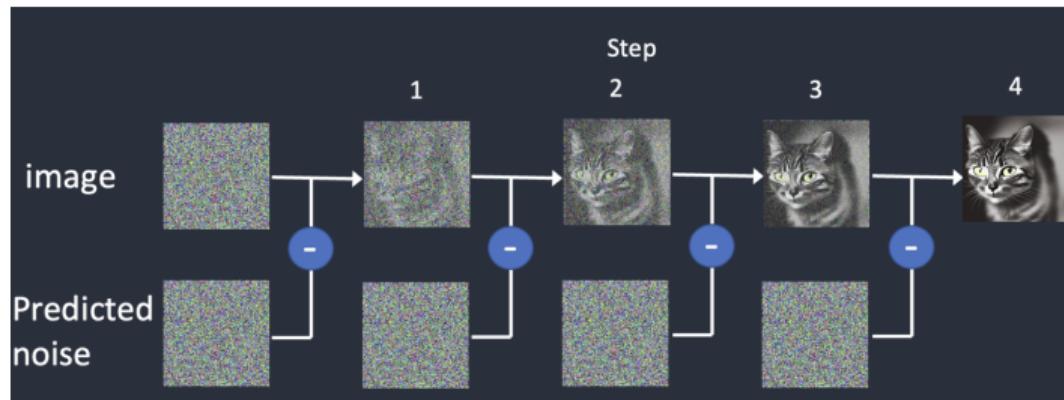
- Pick a training image
- Generate a random noise image
- Corrupt the training image by adding this noise
- Teach the **noise predictor** to tell us how much noise was added
- After training, we have a noise predictor capable of estimating the noise added to an image



# DIFFUSION

- BACKWARD DIFFUSION

- Generate a completely random image
- Ask the noise predictor to tell us the noise
- Subtract this estimated noise from the original image

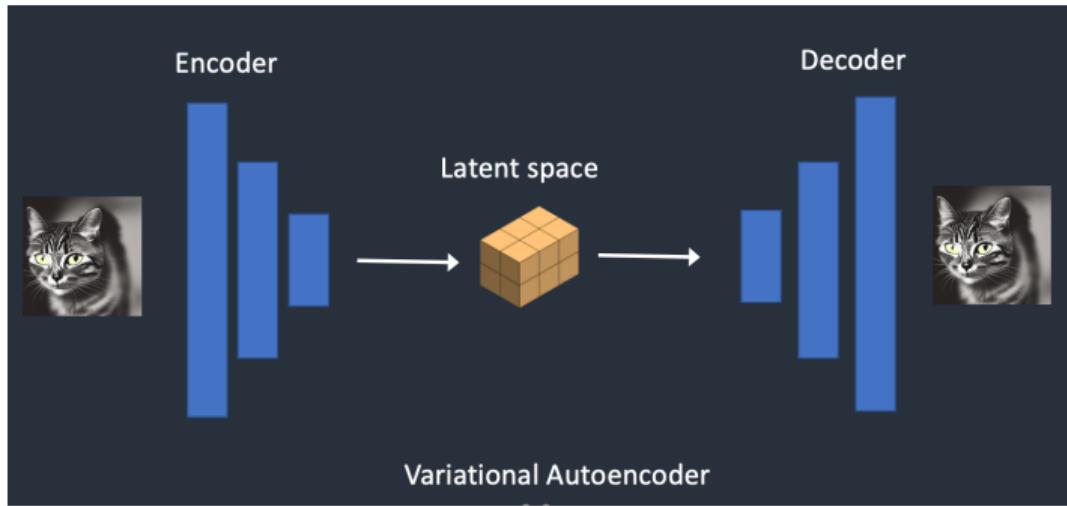


## PROBLEM

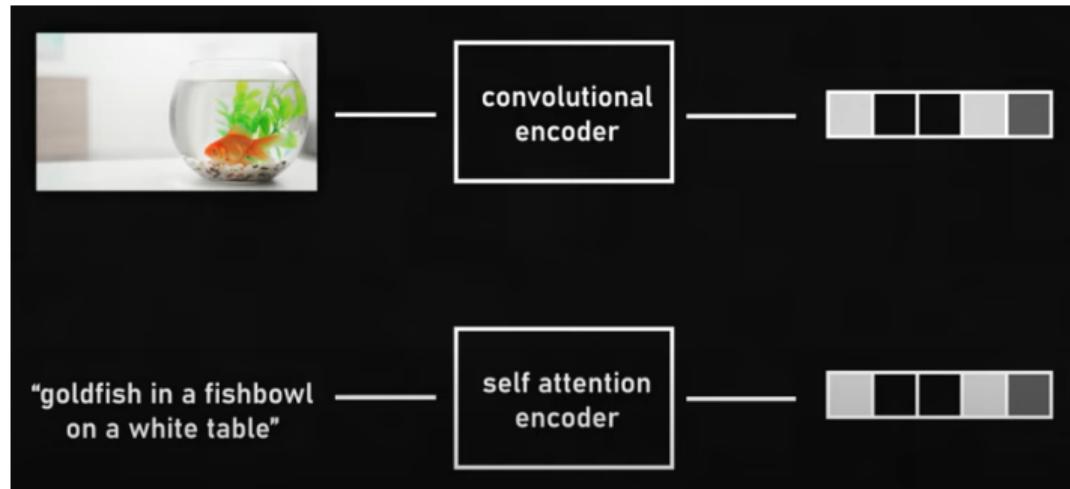
- Image space is enormous
- computationally very high

# LATENT DIFFUSION MODEL

- Use Variational Autoencoder
- Compresses the image into the latent space
- Latent space is 48 times smaller
- Instead of generating a noisy image, it generates a random latent space (latent noise)
- VAE decoder is responsible for painting fine details

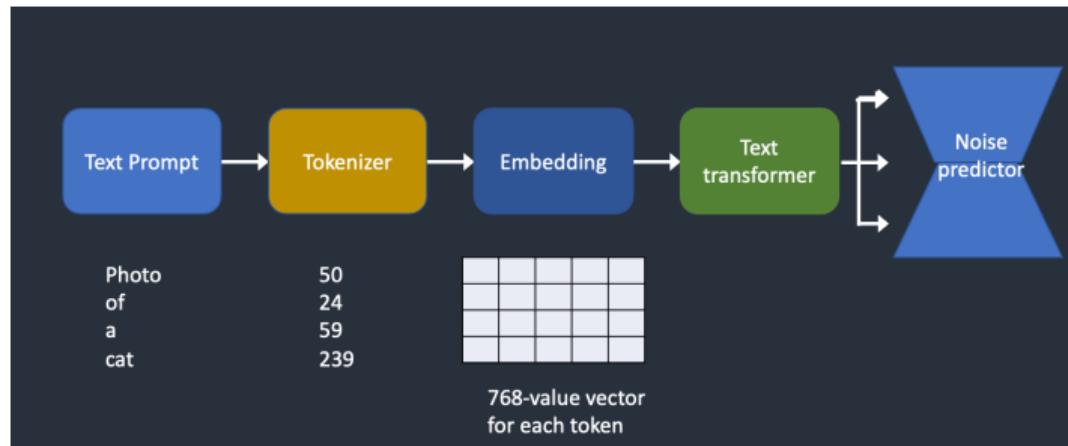


# LATENT DIFFUSION MODEL



# TEXT CONDITIONING

- Steer the noise predictor
- Text prompt is processed and fed into the noise predictor
- TOKENIZER
  - Text prompt is first tokenized by a **CLIP tokenizer**



# TEXT CONDITIONING

- EMBEDDING
  - Stable diffusion uses Open AIs **ViT-L/14 Clip model**
  - Embedding is a 768-value vector
  - Words are closely related to each other
- TEXT TRANSFORMER
  - Provides a mechanism to include different conditioning modalities
  - **cross-attention** between the prompt and the image
  - **A lady with blue eyes** as an example. Stable Diffusion pairs the words blue and eyes together. It then uses this information to steer the reverse diffusion of an image region to render a pair of blue eyes

# TEXT CONDITIONING

- CLIP(Contrastive Language-Image Pre-training)
  - learns to associate images with their corresponding textual descriptions
- GPT-2
  - Generate new captions or descriptions for images based on the learned CLIP embeddings

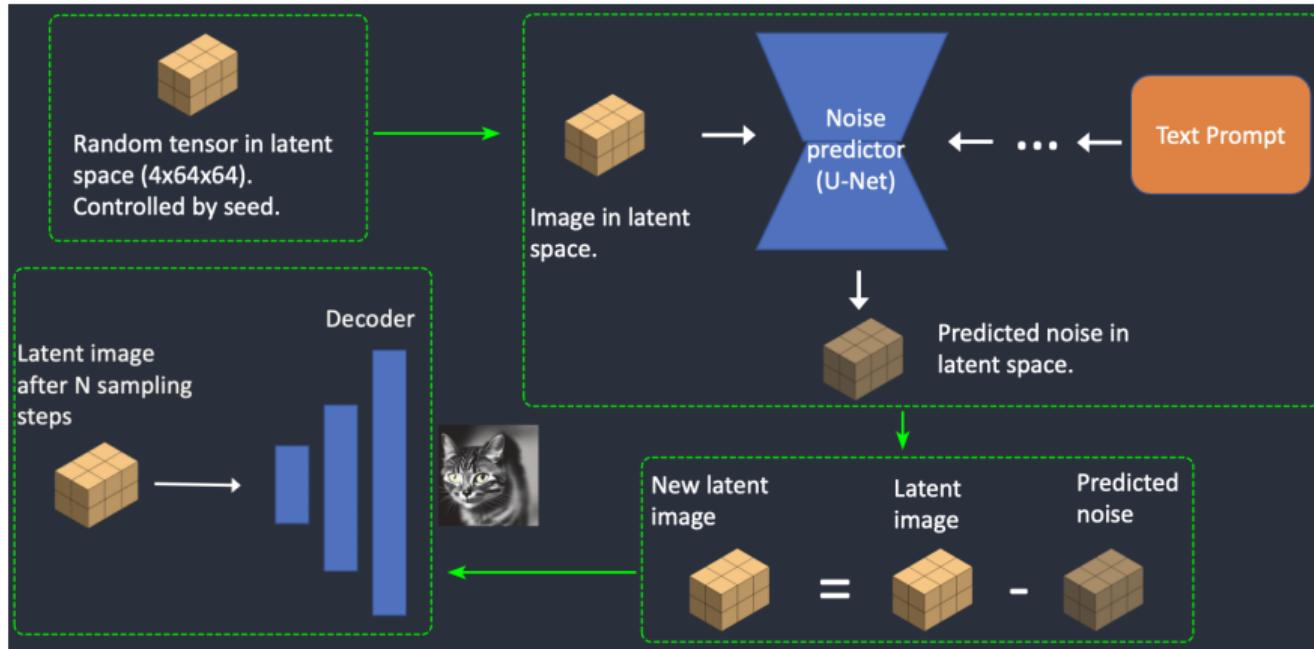
# TEXT TO IMAGE

## STEP-BY-STEP

1. Generates a random tensor in the latent space
2. The noise predictor U-Net takes the latent noisy image and text prompt as input and predicts the noise, also in latent space
3. Subtract the latent noise from the latent image. This becomes your new latent image
4. Steps 2 and 3 are repeated for a certain number of sampling steps, for example, 20 times
5. Finally, the decoder of VAE converts the latent image back to pixel space. This is the image you get after running Stable Diffusion

# TEXT TO IMAGE

## STEP-BY-STEP



# CFG

## Classifier-Free Guidance

- Parameter for controlling how closely should the diffusion process follow the label
- Put the classifier part as conditioning of the noise predictor U-Net
- CFG scale : 0 ( prompt is ignored )
- A higher CFG scale steers the diffusion towards the prompt.

# GOAL BASED TUNING

- Faster Generation
  - Image gen. steps : less (25)
  - Image gen. size : 256\*256
  - Prompt dataset size
- High Image Quality
  - Image gen. steps : more (75)
  - Image gen. size: 768\*768 ( need more VRAM )
  - Image gen. guidance scale: 12 (too high can make things look over-processed or rigid )
  - Prompt max length : 17
- Creative result
  - Lower guidance scale : 7
  - Add temperture to gpt 2 model : high value



# Thank You