

Generation of Graphic Design Models using Stable Diffusion

Deepak N R¹, Shaik Jannat Al Firdaus², Shayan Shaikh³, Syed Tajamul Shah⁴

¹Professor, ^{2,3,4}Student,

Department of CSE, HKBKCE, Bangalore, India

ABSTRACT

The generation of AI Images in recent times has become more mainstream and this technique is quickly making its way into various industries that need asset creation and image generation. Although it is still somewhat limited, AI image generators can create breath-taking photorealistic images from complex prompts. Diffusion models can take in some user input, and some noise, and create content like what it has been trained on. For such cases, the diffusion models are trained using images, and input to the model. Upon passing a user prompt or input, we have the model produce a related image based on the input. Basically, a generative model using the diffusion method can generate images by adding Gaussian noise to the training data, randomizing it, and reversing the noise processing to restore the data. Diffusion models are parameterized Markov sequences, trained with different inferences to produce a picture that fits the data after a period. Using this technology and resources, we infer to create our model that will make use of all these prompts.

Keywords: - Diffusion Modelling, Graphic Design, Stable Diffusion, Image generation, Machine learning.

I) INTRODUCTION

Image processing is an ever-growing domain that has seen its ties with Mathematical Functions and Models. Even in the field of graphical models, multiple software like Blender, Maya, etc., has shown their importance as the Movie industry along with healthcare and the Global Positioning Systems have made use of image processing. However, with the development and growth of Machine Learning and especially Convolutional Neural networks, we have seen image processing reach newer heights that once seemed impossible. With the recent rise of Stable Diffusion which makes use of Diffusers, these new developments have given us the ability to do the unimaginable i.e., create images from just a text prompt and

many more interesting developments This paper dwells deep to find out more about the developments done in the field of image generation and processing and how incorporating textual inversion in the same could help one makes customized images. The ability to make customized images could just help in many processes. From video game models to making comics with oneself as the character to finding appropriate cover images, stable diffusion lets you do about everything especially given AI creates the art for you. Stable Diffusion's edge over other such methodologies is the accuracy and completeness of an image due to the intense training hours put into it.



Fig 1.1. Usage of Diffusion Modelling over the years

II) PROPOSED WORK

2.1 Stable Diffusion Methodology

We mainly focus on the most important forms, these are Salt and Pepper Noise, Gaussian Noise, Poison Noise, Speckle Noise, Impulse Noise and Poison Noise. To convert a text prompt to an image, we follow the following architecture:

1. Authenticate and connect to the OpenAI API: You would need to create an account on the OpenAI platform and obtain an API key, which you can use to authenticate your requests. You can then use a software development kit (SDK) or the API directly to connect to the OpenAI platform. This is needed because we need a model which has been trained on millions and billions of images which cannot be locally made. The choice of Open AI was also made as it facilitates firmware friendly image generation with almost every system being able to generate images and not just high-end systems.

The model is trained on a large dataset of text-image pairs to learn the relationship between text and images. An image generator takes as input a set of tokens produced by a language model and

2. Create a text prompt: You would then create a text prompt that describes the image you want to generate. This can be done using natural language, and you can specify various parameters such as the image size and the style of the image, theme, variation, and many other factors. The more the detailed a prompt is, the better and more accurate image will be generated.

3. Submit a request to the API: You would submit a request to the OpenAI API, specifying the text prompt and any other relevant parameters. The API would then generate an image based on the text prompt using one of its pre-trained generative models, such as the one which we use here, DALL-E.

4. Generate the Image: The model (Dall-E) consists of a transformer-based language model and a transformer-based image generator. A language model takes as input a natural language description of an image and generates a set of tokens representing the image.

produces an image corresponding to a text description. The generator is based on a variant of the GAN (Generative Adversarial Network) architecture, which consists of a generator network and a discriminator network. A generator network

takes a token sequence as input and creates an image. A discriminator network takes an image and a corresponding sequence of tokens as input and attempts to determine if the image matches the description.

5. Retrieve the generated image: Once the Open AI API receives your request, it generates an image using a pre-trained generative model. The generative model is a deep neural network that has been trained on a large dataset of images and text

descriptions. The network takes the text prompt as input and produces an image as output. The exact architecture of the models typically involves several layers of convolutional and fully connected neural networks.

6. Post-processing and use of the generated image: You may choose to post-process the generated image, such as cropping or resizing it, or using it as input to another machine learning model or application.

```
C:\Users\Shayan>imggencli
Usage: imggencli [OPTIONS] COMMAND [ARGS]...

◆ Use this Open AI api to generate, as well as customize : edit & filter
images from the cmd line.

Options:
  --version  Show the version and exit.
  --help    Show this message and exit.

Commands:
  edit      🎨 Change the brightness, contrast and sharpness of an image.
  filter    🎨 Apply filters and effects to an image.
  generate  🌸 Generate an image from the OpenAI api
  img2img   🖼️ Generate an image from an existing image using Open AI api
  update    🗝️ Update the OpenAI API key.
```

Fig 2.3. The command and help directory of the system

Fig 2.3 describes all the available options in the model system with all the descriptions provided. We can use the following commands to generate, as well as customize, edit & filter images from the command line. The options available are:

--version: Shows the version of the existing API.

--help: Shows the whole list of available commands and the necessary options.

We also have a list of commands available. These are:

1) *edit*: This command changes the brightness, contrast, and sharpness of an image.

2) *filter*: This applies the necessary filter and effects to an image.

3) *generate*: This command is used to generate the image from the OpenAI API,

4) *img2img*: This command is used to generate an image from the existing image using OpenAI API.

5) *update*: This command is used to update the OpenAI API key.

```
Microsoft Windows [Version 10.0.22621.1555]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Shayan>imggencli
Usage: imggencli [OPTIONS] COMMAND [ARGS]...

  Use this Open AI api to generate, as well as customize : edit & filter
  images from the cmd line.

Options:
  --version Show the version and exit.
  --help Show this message and exit.

Commands:
  edit      Change the brightness, contrast and sharpness of an image.
  filter    Apply filters and effects to an image.
  generate  Generate an image from the OpenAI api
  img2img  Generate an image from an existing image using Open AI api
  update   Update the OpenAI API key.

C:\Users\Shayan>imggencli generate
Prompt [Suprise me]: "A yellow persian cat with black spots"
C:\Users\Shayan>
```

```
Microsoft Windows [Version 10.0.22621.1555]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Shayan>imggencli
Usage: imggencli [OPTIONS] COMMAND [ARGS]...

  Use this Open AI api to generate, as well as customize : edit & filter
  images from the cmd line.

Options:
  --version Show the version and exit.
  --help Show this message and exit.

Commands:
  edit      Change the brightness, contrast and sharpness of an image.
  filter    Apply filters and effects to an image.
  generate  Generate an image from the OpenAI api
  img2img  Generate an image from an existing image using Open AI api
  update   Update the OpenAI API key.

C:\Users\Shayan>imggencli generate
Prompt [Suprise me]: "A yellow persian cat with black spots"
C:\Users\Shayan>imggencli img2img "C:\Users\Shayan\images\output.png"
C:\Users\Shayan>
```



Fig 2.4 Text to Image & Image to Image Generation

The above Fig 2.4 describe the text to image and image to image generation capabilities of the

system This novel method is contribute and developed to work on any prompts or image within reason.

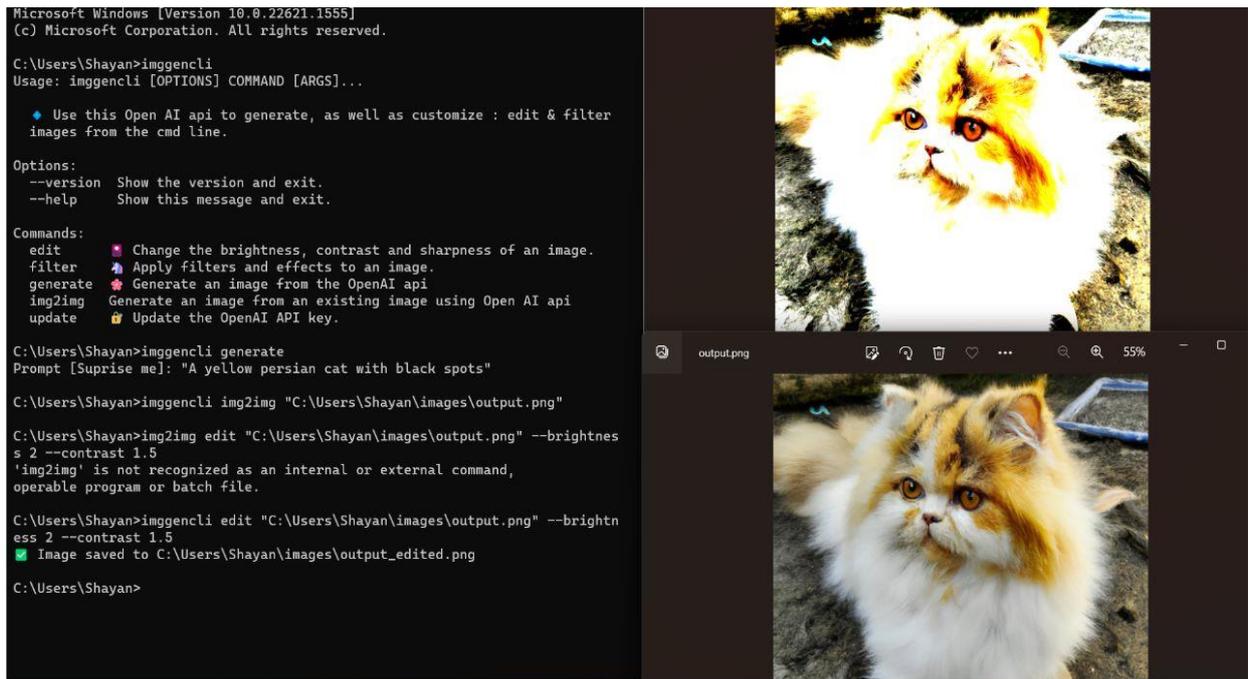


Fig 2.5. Editing the generated image

Fig 2.5 demonstrates the editing capacity of the system based on images such that the brightness, contrast and sharpness of the generated image can

be changed at will. The following flowchart in Fig. 2.6 describes all the necessary steps involved in text to image or image to image generation.

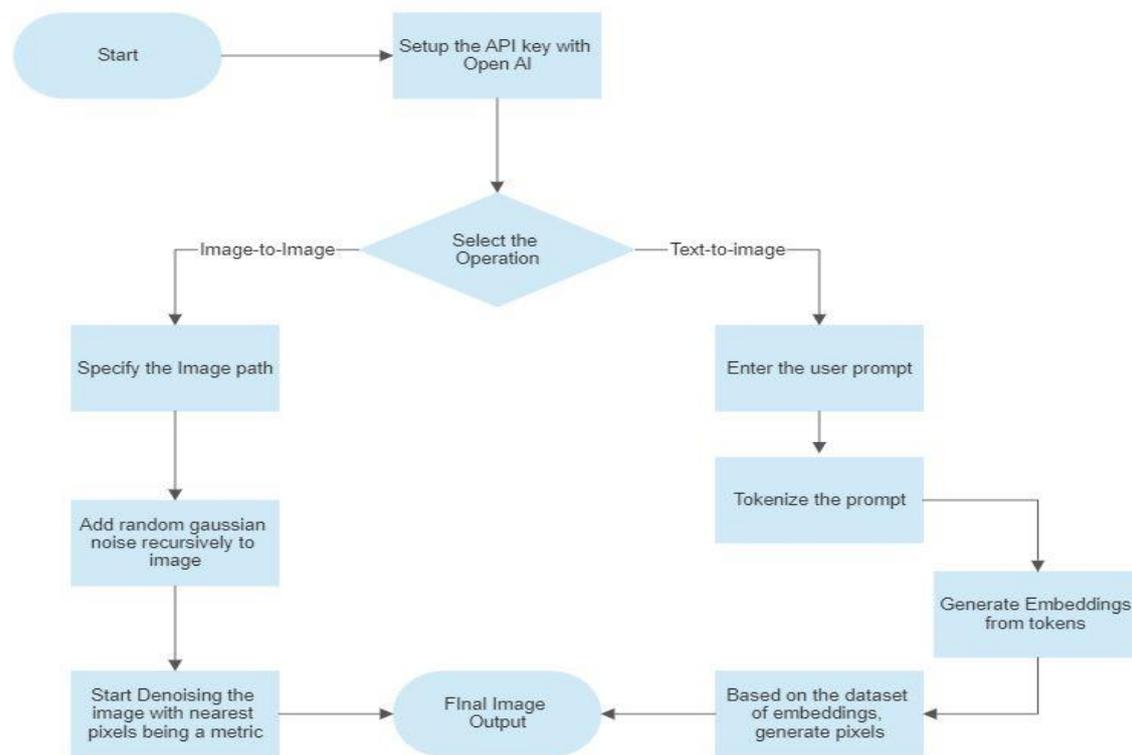


Fig. 2.6 Flowchart describing text to image or image to image generation steps.

The algorithm for image denoising using Smooth Lets is like that using image approximation. It can be described as follows.

1. Do a full quadtree split. Find the best-fit Smoothlet for each node in the quadrilateral tree.
2. Apply the incremental tree pruning algorithm to the given λ to find the best fit.
3. Repeat step 2 for different values of λ and choose the one that results in the best denoising result.

III) RESULTS AND DISCUSSIONS

3.1 Image Generation and Output

The designed model completes the objectives and ensure the completion of the given conclusions to give these precise outputs. The objectives of the model are given as:

- A fully functional working model which allows you to generate images based on user text prompts.
 - Users can specify their own input images and generate customized images out of them.
 - Users can fine-tune their generated image.
- Here, however, an input sample image is needed as a reference image. The generated images are semantically and visually like the one given as a reference. This process is conceptually like the style-based GAN models however, it does much better job in preserving the semantic structure of the image. The given text, however generalized it might be, results in an image generated that is based on the given user text. The given text here

that depicts an image to be generated based on the 3D view of a flying car illustrates various images based on the text. The representations are the various ways of representations of the given text based on the understanding of the model.

IV) CONCLUSION

This paper performs an extensive analysis of the various methods and models that have been used to utilize effectively the diffusion modeling procedure. Through the various papers, it is found that there are different implementations based on the datasets utilized as well as an extensive workaround for common ML algorithms. The main obstacle with the currently existing system is that there exists very limited work in the image generation and graphic designing field. Also, the performance of these systems reduces when further information, as well as data, is fed to these systems.

To overcome this, there is a need for a deep learning model that will be able to carry out the required task effectively and efficiently. It should be able to improve the efficiency of multimedia forensics in generating high-quality images. It should also be able to ensure information integrity with high accuracy and be capable of visualizing and producing all kinds of images. The limitations of the model being that the API itself supports limited images because it only accepts the type of images that it creates. The images that you give in

image to image need to be:

1. A square image. (256×256, 512×512, 1024×1024)
2. A PNG image.
3. Depth of image should be 24 bit and not 8 or 16 bits.

Although converting images to PNG 24 bit is easy and free online.

V) REFERENCES

- [1] Jooyoung Choi; Jungbeom Lee; Chaehun Shin; Sungwon Kim; Hyunwoo Kim; Sungroh Yoon. Perception Prioritized Training of Diffusion Models (27 September 2022)
- [2] Davide Andreoletti; Sebastian Troia; Francesco Musumeci; Silvia Giordano; Guido Maier; Massimo Tornatore. Network Traffic Prediction based on Diffusion Convolutional Recurrent Neural Networks. (23 September 2019)
- [3] Yanan Wang; Xiuzhen Chen; Jianhua Li. A new genetic-based rumor diffusion model for social networks. (10 September 2015)
- [4] Roger Ratcliff; Gail McKoon. The Diffusion Decision Model: Theory and Data for Two-Choice Decision Tasks. (01 April 2019)
- [5] Chengjun Yuan; Dong Ji. Stochastic asymptotically stability of an Information Diffusion Model with Random Perturbation in Social Network (05 August 2019)
- [6] A. Selikhov. mL-CNN: a CNN model for reaction-diffusion processes in component systems. (07 November 2002)