Automatic Synthesis of Realistic Human Images from Text using GANs

Project Guide
Prof. Dr. S.D Deshpande
Dept. of Information Technology
P.E.S. MCOE Shivajinagar, Pune.

Kushal Jivarajani
Information Technology
SPPU

Eeshan Chanpura
Information Technology
SPPU

Anurag Pande
Information Technology
SPPU

Mayur Pokharkar
Information Technology
SPPU

ABSTRACT

The project aims to create a deep learning model that can generate realistic images from a given textual description of a suspected perpetrator in a crime scene. Traditionally, sketch artists are relied upon for this task, but it can be time-consuming and requires highly skilled professionals. To expedite this process, the project uses Generative Adversarial Networks (GANs), a type of deep learning model that has been effective in generating high-quality images. Specifically, the project uses a GAN called VQGAN that is conditioned on the given text using a language conditioning model called CLIP. The model is trained on the CelebA image dataset and deployed using the Flask framework, with a user interface created using Vite.js. The interface also includes a benchmark model for users to compare the results. Overall, the project seeks to improve the speed and accuracy of identifying suspects in criminal investigations by leveraging advances in deep learning and image generation. In this work, we extend this problem to the less addressed domain of face generation from fine-grained textual descriptions of face.

Keywords - GAN, Face, T2F

1. INTRODUCTION

With the emergence of different communication methods, the creation of images from narratives has become a research area. The main goal of our project is to create realistic images based on description. Text-to-face rendering (T2F), a subfield of text-to-image rendering (T2I), is more difficult due to the complexity and variability of facial expressions. Powerful generative competitor networks (GANs) have been developed to generate real images from text. However, most current projects only create simple images like flowers from text.

In this work, we extend this problem from good face description to creating a less relevant face, for example

"A person with curly hair, oval face and black beard".

It has many uses, mostly public safety. Although

there are many models for T2F, the quality of the face design needs to be improved.

Starting from this project, we propose a solution to create a face image that fits the description. Our framework leverages high resolution face generators, GAN-based techniques such as VQGAN and DCGAN, and explores the possibility of using it in T2F. Here we use the annotation to trace the written text and facial features in the input field of the GAN. We train our framework on behavior-based annotation to create better images. The presence of finer-grained content and variable-length subtitles makes the problem easier for users, but more challenging compared to other text-based tasks. The images created for the different explanations show good effect.

1.2 PROBLEM STATEMENT

Develop a machine learning model that can generate realistic images of human faces from textual descriptions, using a Generative Adversarial Network (GAN) trained on a large dataset of human faces and corresponding text descriptions. The goal is to create a system that can generate high-quality and diverse images that accurately represent the input text descriptions.

1.3 OBJECTIVE

a. To implement GAN's for generating images from the given text descriptions for the purpose of identifying criminal faces.
b. To convert natural language text descriptions into images using Deep Learning.
c. Publish research papers and contribute to the academic community's understanding of image generation using GANs.

1.4 SCOPE

The automatic synthesis of realistic images from text using GANs involves a wide range of disciplines, including computer vision, natural language processing, and artificial intelligence, to automatically create realistic images from text using GANs. This cutting-edge technology has a plethora of possible uses, including virtual reality, video game development, e-commerce, advertising, and entertainment. GANs technology holds the potential to alter user experiences and build immersive and interactive environments across a variety of sectors thanks to its capacity to generate highly accurate and lifelike images from textual descriptions.

2. EXISTING SYSTEMS

Text-to-facial image synthesis is a relatively new research area that has gained significant attention in recent years. The goal of this research is to generate facial images from textual descriptions, which has a range of applications, including creating realistic avatars for video games and virtual reality, generating personalized emojis, and aiding in criminal investigations by producing images of suspects based on witness descriptions. One of the earlier approaches to text-to-facial image synthesis involved using hand-crafted features and a decision tree to generate images. These early models were limited by their ability to generate images that lacked fine-grained details and could not capture complex features such as facial expressions and lighting. More recent approaches have utilized deep learning models, particularly Generative Adversarial Networks (GANs), to generate more realistic and detailed facial images. For example, researchers have used a

Conditional GAN (CGAN) to generate facial images from text descriptions by conditioning the generator network on both the textual input and a latent code. Another approach involves using Variational Autoencoder (VAE) to generate images based on a combination of textual input and an image prior, which helps to improve the realism of the generated images. In addition to these approaches, recent work has also explored the use of attention mechanisms to improve the quality of the generated images. These models use an attention mechanism to focus on specific parts of the text input when generating different parts of the facial image. This helps to ensure that the generated images are more coherent and realistic. Overall, previous work in text-to-facial image synthesis has made significant strides in generating realistic and detailed facial images from textual input. However, there is still room for improvement, particularly in capturing fine-grained details and generating more diverse and expressive facial images.

3. TECHNOLOGY

1. Python

Python is a high-level, interpreted programming language that is widely used in many different domains, including machine learning.Python is a versatile language that is known for its ease of use and readability, as well as its powerful libraries and frameworks. In machine learning, Python is commonly used to build and train machine learning models, as well as to process and analyze large datasets.Python's popularity in machine learning is due in large part to its flexibility and ease of use. With its vast ecosystem of libraries and frameworks, it has become a go-to language for data scientists and machine learning engineers alike.

2. Google Colab

Google Colab is a product from Google Research. Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education.

3. Pytorch

PyTorch is an open-source machine learning library for Python that allows efficient computation of multi-dimensional arrays using a tensor-based data structure. It offers automatic differentiation and modular design, enabling developers to create complex neural network architectures. PyTorch also supports distributed training across multiple GPUs and CPUs, making it ideal for research and experimentation in deep learning.

4. GAN

Generative Adversarial Networks (GANs) are a type of deep learning model that consists of two neural networks, a generator and a, that work together to generate new data that is similar to a training set of real data. The generator learns to produce synthetic data that resembles the training data, while the discriminator learns to distinguish between the real and synthetic data. The two networks are trained together in a process where the generator aims to produce more realistic synthetic data that can fool the discriminator, and the discriminator aims to correctly identify the real data from the synthetic data. GANs have been successfully applied in various domains, including image, video, and text generation, and have shown promising results in generating high-quality, realistic data that can be used in a range of applications.

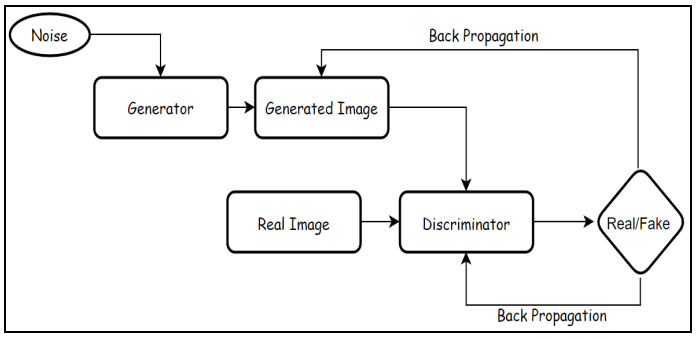


Figure: *Basic GAN Architecture.*

Here , GANs form an important part of our Project as our Model is based on GANs architecture, There are different types of GANs architecture that vary based on . One of them is *VQGAN* (Vector Quantized Generative Adversarial Network).

5. VQGAN

VQGAN *(Vector Quantized Generative Adversarial Network)* is a deep learning model that combines the power of generative adversarial network (GAN) and Vector Quantization layer to generate high-quality, diverse images from textual prompts. The VQGAN architecture was first introduced in a research paper titled "Making the World a Little More Representable, At Least for a Neural Network" by Patrick Esser, Robin Rombach, and Björn Ommer in 2021.

In this architecture, the generator network produces an initial low-resolution image, which is then refined by a series of residual blocks. The vector quantization layer then maps the continuous latent space to a discrete space, where each latent vector is replaced with the closest vector from a learned codebook of discrete vectors.

This allows the VQGAN to generate high-quality images with a greater degree of control over the features of the generated images. The VQGAN has been used in various applications, including image synthesis, style transfer, and image editing.

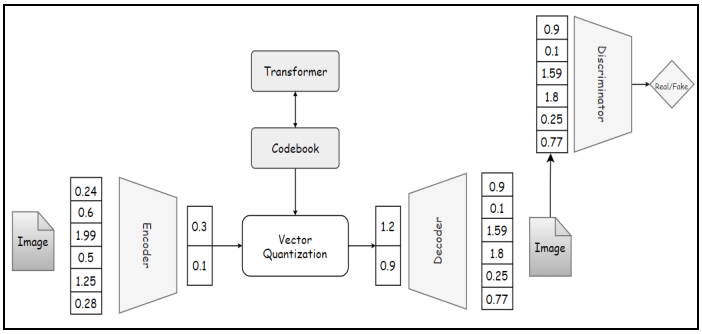


Figure: *The VQGAN architecture*

VQGAN uses a pre-trained encoder network to convert textual descriptions into a compressed vector representation, which is then used by the generator network to produce images. The Vector Quantization technique used in VQGAN helps to generate diverse images that match the textual description by clustering similar image patches in the generator output space. The reason for choosing VQGAN is it has shown impressive results in generating realistic images that match textual descriptions and is an active area of research.

The VQGAN is an extension of VQVAE model which is then simply integrated with usual GAN architecture.

Generally when we see an image we seem to see them through discrete representations. Eg. If we see a man. We see the discrete representations: *man, white, tall/short, hairs* etc. This tells us that our visual reasoning is symbolic. This approach assists us to understand relationships between different words and symbols. In machine learning, this is commonly known as being able to model long-range dependencies. VQGAN can learn not only the visual parts of an image, but also the relationship between these parts (long range dependencies).

The VQGAN architecture consists of :

1. A convolutional neural network which helps in learning the visual parts of the image. We can learn high level features
2. A transformer network to learn long range

interactions.

3. A codebook which helps in training the transformer.

Common to most techniques is a two-stage approach that first learns a representation from the image and encodes it to a different form before feeding into a transformer. However, transformers have a limitation: they scale quadratically. As a result we have to reduce the pixel dimensions of the image. However, instead of downsampling the image, VQGAN uses a *codebook* to represent visual parts. It does not model the image from a pixel-level directly, but instead from the codewords of the learned codebook. VQGAN was able to solve Transformers' scaling problem by using an intermediate representation known as a codebook which then fed to a transformer network. The codebook is then learned using *vector quantization (VQ)*.

Vector quantization is a signal processing technique for encoding vectors. It represents all visual parts in a quantized form, making it less computationally expensive once passed to a transformer. What happens during vector quantization is that we divide vectors into groups with each group having a centroid (codeword). So on a high level we can consider these codewords as discrete symbols. So by training them with a transformer we can under or discover their relationships. The codebook consists of discrete vectors consisting of the same dimensions as of latent vectors generated from encoders. So after encoding our image to a latent space we replace the vector with its nearest vector from the codebook. This process is a complex process because our encoder outputs multiple vectors for each image and each of these will be replaced with corresponding nearest neighbor, this just increases the number of possible latent vectors .

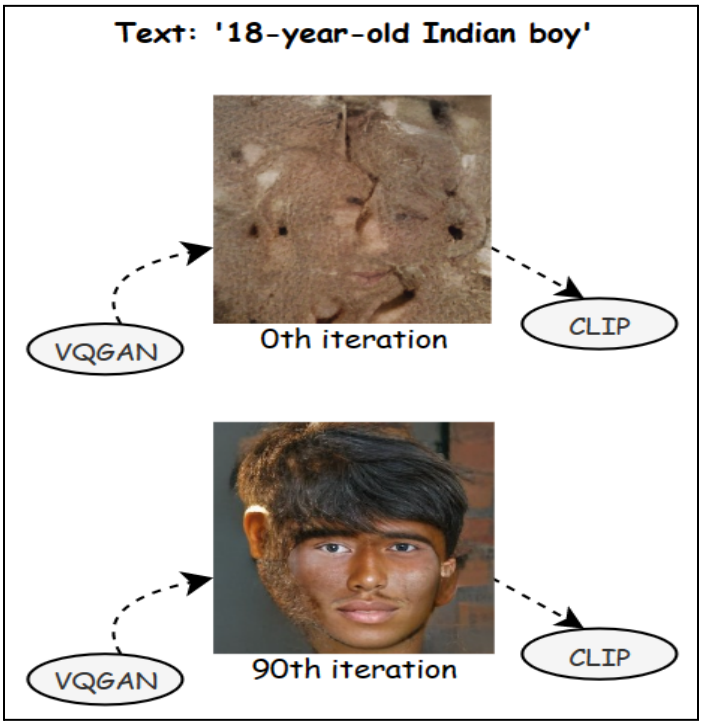


Figure: *The VQGAN model generates images while CLIP evaluates the process.*

So, what happens in a VQGAN is we give an image as an input to our *encoder* which then converts this image to a latent space. Then these latent vectors are replaced by the vectors from the codebook and the *decoder* decodes this to a reconstructed image. Additionally, we also have a discriminator in our training loop which takes in the reconstructed image and the original image and tries to distinguish between real and fake ones. This provides additional information to our encoder and decoder as to how and what to improve to fool the discriminator which leads to mimicking the original data distribution better. With this setup right here we are successfully able to reconstruct our image.

But the question arises *how do we generate new images?*

To answer the above question we introduce a second stage of VQGAN which are the transformers. Transformers are on the rise

now-a-days and are taking over as the de-facto state-of-the-art architecture in all language-related tasks and other domains such as audio and vision. Transformers are free to understand and learn the complex relationships among the inputs. One key goal is to obtain an effective and expressive model that combines convolutional and transformer architectures. As we know codebook would learn the pattern of the latent vectors and would assign specific codebook vectors to each property. For example we would have a vector which represents green eyes and a vector which represents black hairs. So if we want to construct a face of a boy with brown hairs and blue eyes and long chin we can just take these vectors from code and fuse them together to the decoder to generate the image.

So the transformer here learns which codebook vector would make sense together and which does not. And after learning this it generates new possible combinations of codebook vectors which it thinks would go along together. The reason we are using a transformer is because a codebook vector representing an image is just a sequence and transformers are really good sequences.

We are able to generate images which are way bigger then the training images because the encoder and decoder are fully convolutional architectures and as a result are invariant to image sizes. For this the transformer will sample more latent vectors than usual to the decoder.

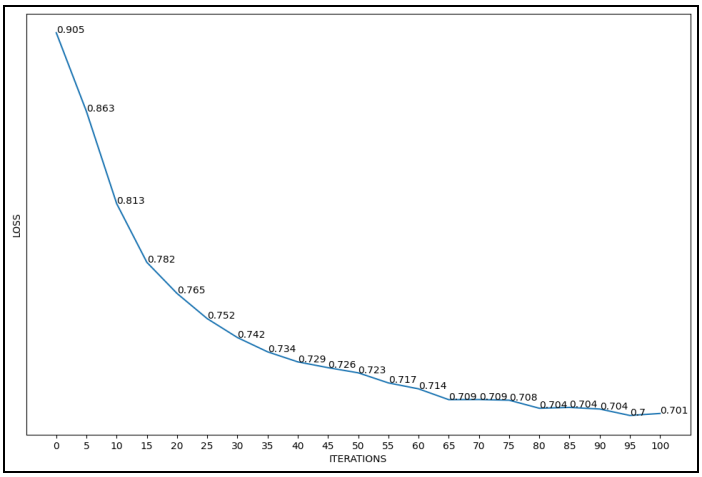


Figure: *Line chart of loss per iterations.*

6. CLIP

A neural network model called **CLIP** ***(Contrastive Language-Image Pre-Training)*** was created by OpenAI and is trained to comprehend the link between natural language descriptions and images. It can identify and categorize different visual concepts since it has been pre-trained on a sizable dataset of photos and the descriptions that go with them.

By defining textual prompts, CLIP provides a powerful way to control the production of graphics when combined with VQGAN. You can feed a prompt into CLIP to get an embedding that explains the prompt's purpose. The development of a picture can then be guided by the VQGAN model using this embedding. Essentially, CLIP acts as a bridge between the textual and visual worlds, allowing users to modify the features of the produced image using straightforward natural language instructions.

For example, a user could enter a prompt like "a young man with brown beard and white skin with straight black hair" into CLIP, and the resulting embedding could be used to guide the VQGAN in generating an image that matches that description. The VQGAN can also be

fine-tuned on specific datasets to generate images that match specific styles or characteristics.

The combination of CLIP and VQGAN has shown impressive results in generating high-quality images that match specific prompts, making it a useful tool for various creative and practical applications, such as generating artwork, product designs, or even realistic simulations.For a comprehensive explanation you can refer [6].

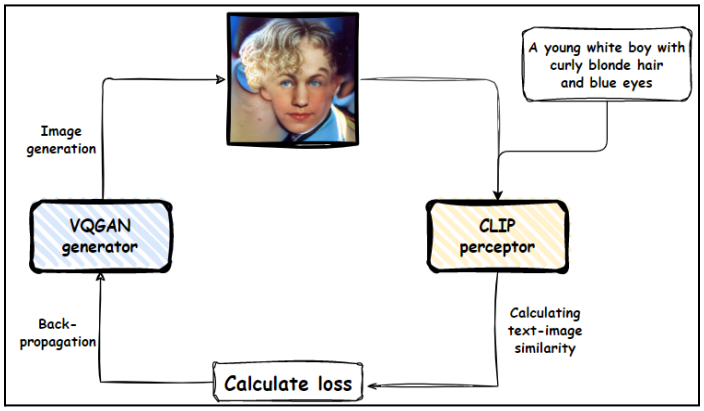


Figure: *CLIP and VQGAN.*

In VQGAN, CLIP is used to guide the image generation process by providing a score for how well the generated image matches the input text prompt. This score is computed by comparing the embedding of the generated image with the embedding of the input text using a similarity measure, such as cosine similarity. By incorporating CLIP into VQGAN, the model can generate more accurate and diverse images that better match the input text descriptions.

This model is based on zero-shot transfer, natural language supervision, and multimodal learning.

7. WORKING

Whenever we say VQGAN-CLIP, we refer to the interaction between these two networks. They are separate models that work in collaboration. In short, the way they work is that VQGAN generates the images, while CLIP evaluates how well an image matches our input text. This helps our generator to produce more errorless images. In the case of VQGAN+CLIP we deal with 2 models: VQGAN is trained on a mostly canonical dataset like CelebA or COCO. CLIP on the other hand was trained on a vast (and unknown) dataset of random internet material.

Procedure:

1. Initially the user gives a text description as an input.

2. The generator of VQGAN generates an image and this image is then used by the CLIP model.

3. The CLIP model is an evaluation model which takes in the input prompts and generated image and then evaluates the image to find the similarities between them.

4. Now as we are familiar with the loss in the image, the GAN generates a new improvised image.

5. The above steps are repeated from multiple iterations unless we get a required image which fits the text description.

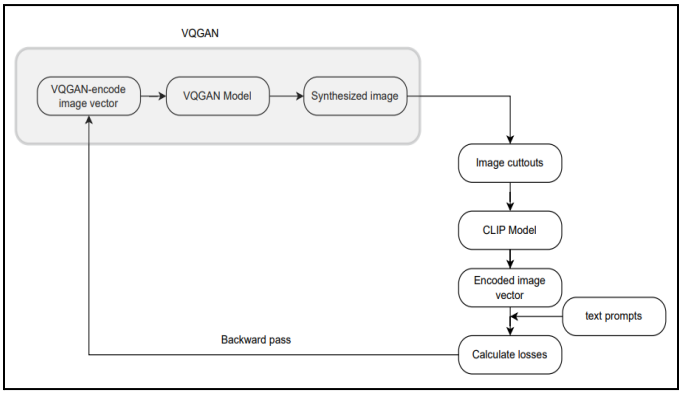
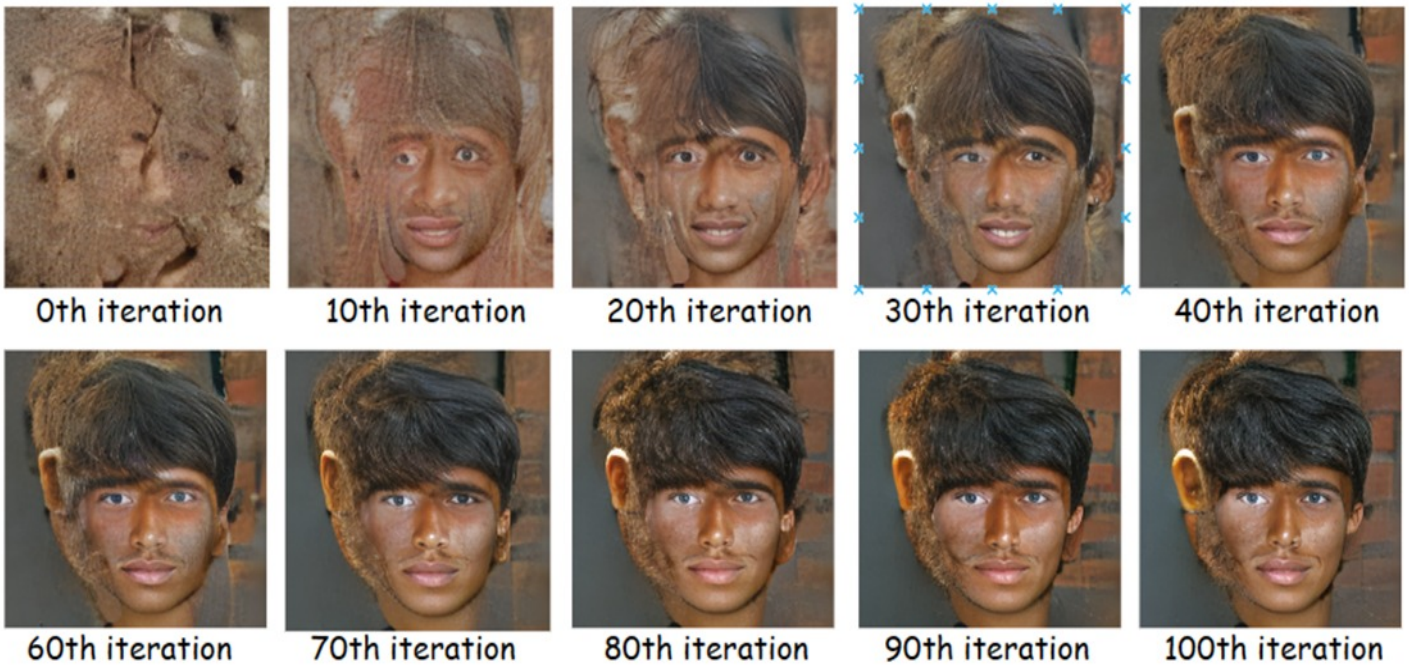


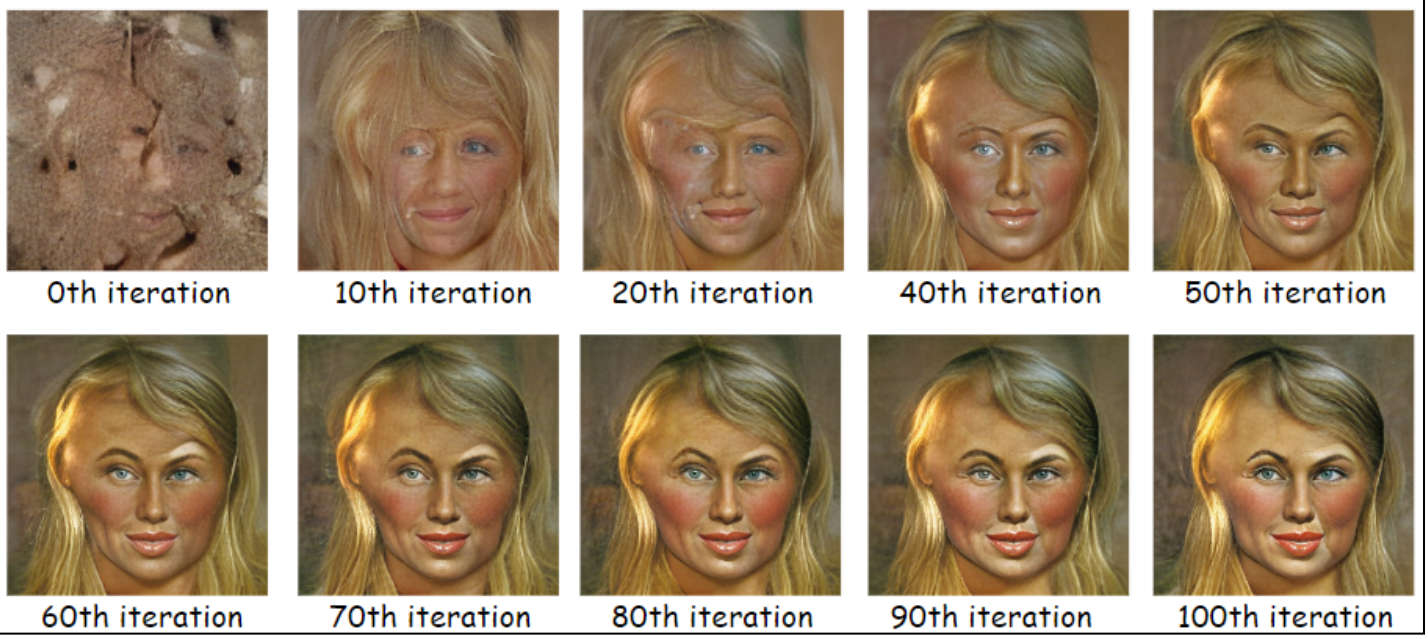
Figure: *Workflow of the model.*

8. RESULTS

Text: 'An average Indian boy, 18 years old, has a round face with dark hair and brown eyes. He has a straight nose and a light to medium skin tone with minimal facial hair'



Text: 'The girl has oval or round face with defined cheekbones and slightly pointed chin. She has Light-colored, deep-set eyes with long lashes and arched eyebrows and straight and narrow nose, full or thin lips with natural color. She has straight blonde hairs.'



9. CHALLENGES

a. Training the model on low end devices may result in slow computation and heating up the device.
b. Generation and accuracy of the generated face is limited to the significant features of the facial features.
c. Generating realistic images from text requires even more data than traditional image synthesis tasks. Additionally GANs need large datasets to train on. Therefore, techniques for scaling up the training process and handling large amounts of data must be developed.

10. FUTURE WORK

Text-to-face generation using GANs has a lot of potential for future development and applications. Here are a few possible areas of future scope:

1) Future research on the automatic synthesis of realistic images from text using GANs may concentrate on improving the quality and variety of the generated images, improving the model's handling of complicated and multi-sentence descriptions.
2) Using High-end Devices and Infrastructure for faster computation power. Thus increasing scalability of model.
3) Adding Speech Navigation for describing the prompt and directly adding the prompt to the model instead of typing it.
4) Forensic investigations: Law enforcement agencies could use text-to-face generation to generate images of suspects based on eyewitness accounts. This could potentially help solve cases where traditional composite sketches fall short.
5) Medical applications: Text-to-face generation could be used in various medical applications, such as reconstructive surgery planning or facial prosthetics. Doctors could input specific facial feature descriptions to create personalized models for their patients.
6) Advertising and marketing: Marketers could use text-to-face generation to create targeted advertisements that feature models with specific facial features that appeal to their target audience.

11. CONCLUSION

In conclusion, our research has shown that VQGANs can produce decent images from text descriptions. Our tests demonstrated that the model could generate a wide range of graphics that tried to match the input text cues. VQGANs are computationally demanding and need a lot of training data, thus there is still opportunity for development in terms of scalability and effectiveness. Overall, our findings imply that VQGANs have enormous potential for use in virtual reality, gaming, and entertainment, where realistic facial images can improve the user experience.

12. REFERENCE

[1] Patrick Esser, Robin Rombach .., “Taming Transformers for High-Resolution Image Synthesis” arXiv:2012.09841v3 [cs.CV] 23 June 2021.

[2] Ayanthi, Akila & Munasinghe, Sarasi. (2022). Text-to-Face Generation with

StyleGAN2. 49-64. 10.5121/csit.2022.120805.
[3] T. Xu et al ., "AttnGAN: Fine-Grained Text to Image Generation with Attentional Generative Adversarial Networks," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition , 2018, pp. 1316-1324, doi: 10.1109/CVPR.2018.00143.
[4] H. Zhang et al., "StackGAN++: Realistic Image Synthesis with Stacked Generative Adversarial Networks," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 41, no. 8, pp. 1947-1962, 1 Aug. 2019, doi: 10.1109/TPAMI.2018.2856256.
[5] H. Zhang et al ., "StackGAN: Text to Photo-Realistic Image Synthesis with Stacked Generative Adversarial Networks," 2017 IEEE International Conference on Computer Vision (ICCV) , 2017, pp. 5908-5916, doi: 10.1109/ICCV.2017.
[6] Alec Radford, Aditya Ramesh ., “Learning Transferable Visual Models From Natural Language Supervision” arXiv:2103.00020v1 [cs.CV] 26 Feb 2021.
[7] Lj Miranda The Illustrated VQGAN 08 Aug 2021.
[8] Ryan Moulton Generating panorama images 23 Aug 2021.