**Text To Face Generation Using DCGAN With Bert-Embedding Vectors**

| Abhishek Pawar,  abhishek.pawar@mitaoe.ac.in  School of Computer Engineering,  MIT Academy of Engineering,  Alandi (D), Pune | | Riya Hiwanj  riya.hiwanj@mitaoe.ac.in  School of Electrical Engineering,  MIT Academy of Engineering,  Alandi (D), Pune | | Ashwini Jadhav  ashwini.jadhav@mitaoe.ac.in  School of Computer Engineering,  MIT Academy of Engineering,  Alandi (D), Pune | |
| --- | --- | --- | --- | --- | --- |
| Asrar Sheikh,  asrar.sheikh@mitaoe.ac.in  School of Computer Engineering,  MIT Academy of Engineering,  Alandi (D). Pune | | | Sharmila Kharat,  sbkharat@mitaoe.ac.in  School of Computer Engineering,  MIT Academy of Engineering,  Alandi (D), Pune, India | | |

**Abstract:**

**Text-to-face generation aims to generate facial images from textual descriptions. It has a significant impact on new research fields and the wide spectrum of public safety applications. The study of text-to-face generation is quite scarce due to a lack of datasets. This paper aims to enhance text-to-face image generation using deep convolutional generative adversarial networks and Bert embedding models. Our methodology incorporates domain-specific BERT embedding models for better understanding of face text descriptions. To enhance its knowledge understanding for the face domain, the BERT model is fine-tuned over large text face description corpus data as a transfer learning method, subsequently enhancing word embedding for substantial textual data. This paper presents an improved bert model training approach with DCGAN for facial feature understanding and face image generation. We evaluated the results in terms of binary cross entropy loss over a period of epochs and the realistic image outputs generated by the approach model.**

**keywords: Text-To-Face Generation, DCGAN, BERT Embedding , Transfer Learning, Fine-Tuning.**

# **Introduction**

The field of text-to-image generation has witnessed a remarkable transformation in recent years, owing to the advent of deep learning techniques. Prior to this, text-to-image models relied heavily on attention based recurrent neural networks[1], which often produced limited and unrealistic visual outputs. However, the exposure of deep learning opened up new possibilities for generating cohesive and lifelike visuals that closely align with textual descriptions.[2] This capability allows for the creation of composite images of suspects based on witness descriptions, aids in understanding textual narratives through image stories, and contributes to the development of realistic avatars for gaming and virtual reality.

Even as the AlignDRAW model underwent significant evolution, it demonstrated the ability to generalize to objects not encountered in its training data.[3] This capability was a testament to its capacity to go beyond mere data memorization.

In 2016, a pivotal moment occurred when Reed, Akata, Yan, and their collaborators pioneered the use of generative adversarial networks (GANs) for text-to-image generation.[4] They successfully generated visually credible images, including birds and flowers, from text descriptions. However, fine-grained details remained a challenge.

The landscape of text-to-image generation continued to evolve with the introduction of models like VQGAN+CLIP [5], XMC-GAN[6], and GauGAN2[7]. However, it was OpenAI's DALL-E, a transformer-based system launched in 2021[8], that truly captured widespread attention. Its successor, DALL-E 2, introduced in 2022[9], promised even more detailed and lifelike graphics, further advancing the field. Additionally, Stable Diffusion became accessible to the public in August 2022[10], further fueling innovation in this area.

Despite these advancements, generating facial images from text inputs remains a complex task, primarily due to the nuanced nature of facial features. Existing models require comprehensive datasets mapping facial features to textual descriptions to create meaningful embeddings for accurate image generation. The challenge lies in precise text descriptions that specify the exact facial attributes, ensuring the generated images closely match the input.

DCGAN is one of the models that has stable training over the epochs, less computational complexity, and uses convolutional neural networks (CNNs) for both the generator and discriminator models. The generator model learns to generate realistic images from random noise, while the discriminator model learns to distinguish between real and fake images. In a paper titled Application of an Improved DCGAN for Image Generation, Mobile Information Systems by Liu, Bingqi, Lv, Jiwei, Fan, Xinyue, Luo, Jie, and Zou, Tianyi. (2022). The authors have justified their selection of DCGAN by stating that it has shown promising results in generating high-quality images in various image generation tasks. They have also mentioned that DCGAN is capable of learning complex image features and generating high-resolution images. Another Paper titled Face Generation using DCGAN for Low Computing Resources 2021 by W. Liu, Y. Gu and K. Zhang, Support DCGAN as a suitable choice for text-to-face image generation task.

Therefore the proposed solution leverages the power of Deep Convolutional Generative Adversarial Networks (DCGAN) in synergy with BERT embedding vectors. BERT, a widely acclaimed NLP model, serves as the text encoder, fine-tuned with extensive textual data to extract rich information for image generation. This integration of DCGAN and BERT promises to yield highly detailed and accurate facial images from textual descriptions, with potential applications in law enforcement, education, entertainment, and beyond.

In the subsequent sections, we delve into the technical details of our approach, highlighting the intricate process of encoding text descriptions, generating realistic facial images, and training our model through adversarial learning. Our aim is to contribute to the ongoing advancement of text-to-face generation, offering a novel and effective solution to a multifaceted challenge at the intersection of language and imagery.

# Literature survey

Text-to-face generation has been a significant milestone in the field of artificial intelligence and computer vision, enabling the synthesis of realistic human facial images from textual descriptions. Traditional systems in this domain often relied on rule-based or template-based approaches, where predefined facial features and attributes were combined to generate facial images.[11] While these systems had the advantage of being computationally efficient, they suffered from limited flexibility and the inability to capture the full range of human facial diversity. Disadvantages included the generation of generic and unrealistic faces, with little scope for personalization or fine-grained control over generated images.

In our comprehensive research survey, we have explored various state-of-the-art approaches in text-to-face generation. These advancements have significantly improved the quality, diversity, and realism of generated facial images. Han Zhang et al. [12] StackGan++ addressed the challenges of image resolution and multi-distribution, resulting in high-resolution images aligned with textual descriptions. Timo Aila et al [13]. progressive growing of GAN introduced dynamic layer addition, enhancing image quality and stability. Muhammad Zeeshan Khan et al. [14] leveraged textual representations for expression-driven image generation, albeit with diversity challenges. Tero Karras et al. [15] style-based generator regulated image synthesis, though lacking an encoder for input image-to-latent code mapping. Brian Lovell et al. integrated BERT-based NLP and StyleGAN2 for Text to Face HD, mitigating mode collapse and diversifying high-resolution image output[16]. A. Kumar et al.[17] LSTM-based approach encoded text into semantic vectors for natural image generation.

These advancements have led to widespread applications in forensic investigations, entertainment, medicine, and gaming. However, limitations persist. High-resolution facial image generation remains computationally intensive, limiting real-time applications. Challenges arise when specific facial characteristics must be altered, and there is a tendency to generate non-human entities more efficiently. The training of GANs demands substantial data and resources, while transformers are memory-intensive, restricting deployment on resource-constrained devices.

Text-to-image generation is a challenging task in the field of generative deep learning. Various approaches have been proposed to tackle this problem using DCGAN (Deep Convolutional Generative Adversarial Networks). In this comparative study, we will explore and contrast three prominent methods for text-to-image generation using DCGAN to understand their respective strengths and limitations.

| **GAN** | **Inception Score** | **Fréchet Inception Distance** | **Clean FID** |
| --- | --- | --- | --- |
| DCGAN | 2.84 ± 0.062 | 87.146 | 87.58 |
| SAGAN | 2.342 ± 0.039 | 114.512 | 115.256 |
| DFGAN | 2.865 ± 0.041 | 109.14 | 106.453 |

Table 1: Comparative analysis of DCGAN,SAGAN & DFGAN

In conclusion, while significant progress has been made in text-to-face generation, there are several gaps and challenges that need to be addressed. The computational demands for high-resolution image generation and the tendency to produce non-human faces remain prominent concerns. Additionally, the need for more efficient models that can handle specific facial attribute modifications is evident. Bridging these gaps will be crucial for the broader adoption of text-to-face generation in various applications, ensuring both realism and efficiency in facial image synthesis. Further research is needed to develop more resource-efficient algorithms and improve the diversity and controllability of generated facial images.

1. **Proposed System**
   1. **Dataset**

The "CelebA" dataset contains a total of 202,599 face images. These images belong to 10,177 unique identities or individuals. However, the actual names of these identities are not provided. Each image is annotated with 40 binary attributes, indicating the presence or absence of specific facial attributes.

These attributes include characteristics such as hair color (e.g., brown hair), facial expressions (e.g., smiling), and the presence of eyeglasses. A value of "1" indicates the presence of the attribute, while "-1" indicates the absence of the attribute.

Overall, the CelebA dataset provides a diverse and comprehensive collection of labeled face images, making it an essential resource for advancing the field of facial analysis in computer vision.

We created a large text corpus data related to face description, which was used to enhance the bert embedding model, to increase its domain understanding knowledge to interpret the input commands more efficiently and in a broader view.

For all the celebA images we prepared a paired text description file containing the image filename and respective textual description. For such a large dataset we incorporated a function to take all the attributes of an image from the attributes csv and generate a natural language text using openAI saved in paired\_text\_description csv.

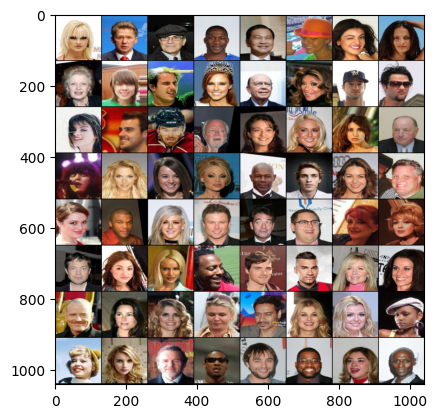


Fig 1: Data visualization of data in a batch of size 80

1. **System Architecture**

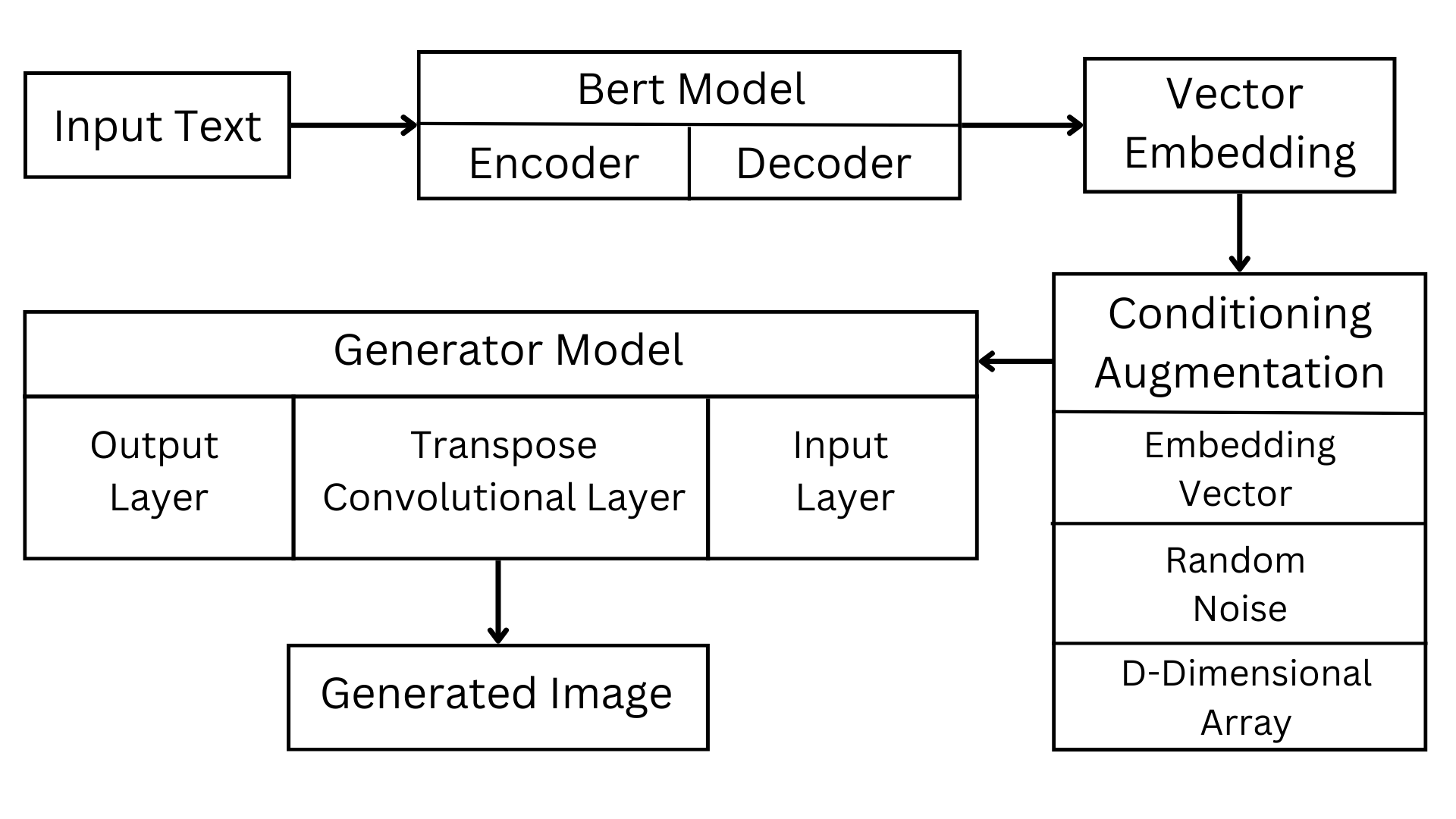
****

Fig. 2. System Architecture Diagram : Explaining how the system must work**.**

During the system's initial phase, textual input in the form of facial features is sent into the BERT module, which transforms the text into embedding vectors. The encoder and decorator in the BERT module take an input, transform it into mask symbols in the encoder, and then pass it to the decoder to create an embedding that preserves the semantic information. The d-dimensional array is formed by concatenating the BERT embeddings with random noise in order to prepare the input data for the generator.

This array combines the rich semantic information from BERT embeddings with stochastic variations from random noise. This step is vital for introducing diversity into the generated images.

The heart of the system is the DCGAN-based generator. This generator takes the d-dimensional array as input, where BERT embeddings play a pivotal role in providing textual context. The generator employs a series of transpose convolutional layers to upsample the data and generate fake images. By blending BERT embeddings with random noise, the generator creates images that align with the textual descriptions provided as input.

The final step involves a discriminator, which evaluates the authenticity of the generated images. Pairs of real images and fake images, produced by the generator, are presented to the discriminator for classification. The discriminator is modified to accept BERT embeddings as part of the input data, ensuring that the semantic content of the text is considered during the discrimination process. The discriminator's role is to distinguish between real and fake images, providing feedback to the generator to improve image quality.

1. **Methodology**
   1. **Data Collection And Preprocessing**

The CelebA dataset with over 200,000 celebrity images is the primary data source. Data collection involves unzipping files to access images and metadata. During preprocessing, images are loaded using PIL, transformed (resized, converted to tensors, and normalized), extracting true text, and randomly selecting "wrong" images for triplet creation. Divided into training and validation sets for model learning and evaluation.

The \_\_getitem\_\_ method returns true images, true text, and wrong images. The data\_processing method reads data from a CSV file, eliminates certain columns from the dataset ("Bags\_Under\_Eyes," "Bangs," "Blurry," "No\_Beard"), and returns a cleaned DataFrame with only relevant characteristics and classes. The weight\_generation function generates a list of weights for each data point by calculating class counts and weights based on label occurrences in attribute data. The method get\_weighted\_dataloader produces random indices for a dataset, processes it using weight\_generation, builds a weighted DataLoader, and provides its iterator.

* 1. **Bert Embedding Model**
     1. The bert-base-uncased model from hugging face is trained over large text corpus data which is unknown to its end users. and generally consists of general knowledge across a domain. We tried to make a face domain sentence embedding model using bert-base-uncased. We create one data.txt file containing more than 100000 sentences related to faces descriptions based on face attributes. We focused on detailed description taken from books, internet sources, many of them AI generated and rest were employed from existing celebA text description file. We performed Masked Language Model (MLM) pre-training on BERT (Bidirectional Encoder Representations from Transformers) using the Hugging Face Transformers library.
     2. The code initializes a BERT tokenizer and a BERT model for Masked Language Modeling using the Hugging Face model hub. Face descriptions are tokenized using the tokenizer.encode, and a random 15% of tokens are selected to be masked. These masked tokens are predicted by the model during training. The masked input sequences are converted to PyTorch tensors, and a PyTorch TensorDataset is created. A PyTorch DataLoader is set up for batch handling and It sets up an AdamW optimizer and a CrossEntropyLoss criterion for training.. The model is trained using a masked language modeling objective, iterating through the data loader, computing the model's loss, and performing backpropagation to update parameters. The tqdm library provides a progress bar for training iterations. After training, the pre-trained BERT model is saved to a specified directory .
     3. In the Defined SentenceEncoder class which leverages the Fine-tuning Bert-based model over a face description corpus data along with SentenceTransformer model to convert batches of text sentences into dense vector embeddings. The class constructor initializes the model and specifies the computation device. The convert\_text\_to\_embeddings method splits input sentences into sub-sentences, encodes them into embeddings, and computes the mean embedding for each sentence. These embeddings are then concatenated to form a matrix, which is returned as a PyTorch tensor. This class is useful for extracting meaningful representations of text sentences, which can be employed for various natural language processing tasks.
  2. **DCGANs Model**

1. **Model Architecture :**The DCGAN consists of two main components: the generator and the discriminator. These networks are initialized using the `Generator` and `Discriminator` classes defined in the code.

The Generator begins with a linear projection of text embeddings, reducing their dimensionality. It then utilizes transposed convolutional layers for up-sampling, progressively increasing the spatial dimensions of the input noise and projected text embeddings. Batch normalization and leaky ReLU activation functions enhance training stability and feature diversity.

The Discriminator comprises a series of convolutional layers for down-sampling real and generated images. Additionally, it linearly projects and concatenates text embeddings, leveraging both image and text information. The discriminator's final layer produces a probability map indicating the authenticity of the input. Both Generator and Discriminator employ spectral normalization for weight stability. The overall architecture emphasizes proper weight initialization and embraces Batch Normalization for normalization. Despite not incorporating certain advanced techniques like progressive growing or attention mechanisms, the DCGAN demonstrates effectiveness in generating high-quality facial images with a focus on stability and realism.

1. **Loss Functions:**The binary cross-entropy loss (`nn.BCELoss()`) is used. This loss measures the difference between the predicted probabilities (output of the discriminator) and the target labels (real or fake).

In addition to BCE loss, the generator also uses MSE loss (`nn.MSELoss()`) and L1 loss (`nn.L1Loss()`) to encourage generated images to be similar to real images. MSE loss measures the pixel-wise mean squared difference between generated and real images, while L1 loss measures the absolute pixel-wise difference.

1. **Optimization Algorithm:**

The Adam optimizer is used for both the generator and discriminator. The optimizers are defined as `torch.optim.Adam` objects with specified learning rates and betas (momentum terms). The discriminator optimizer uses a learning rate of 0.0001, while the generator optimizer uses a learning rate of 0.0002.

* 1. **Training**

The training loop follows a GAN training procedure and alternates between training the discriminator and training the generator:

1. **Discriminator Training:**

Real images and their associated text descriptions are loaded from the dataset. The discriminator is trained to classify real images as real (label=1) and fake images (generated by the generator) as fake (label=0). Real images are assigned labels with a small negative value (-0.1) to encourage the discriminator not to be too confident.

1. **Generator Training:**

The generator is trained to generate images that can deceive the discriminator. Random noise vectors are generated and concatenated with text embeddings. The generator's output is passed to the discriminator, and the loss is computed using BCE loss. Additional losses, including MSE loss and L1 loss, are used to encourage image quality and similarity to real images.

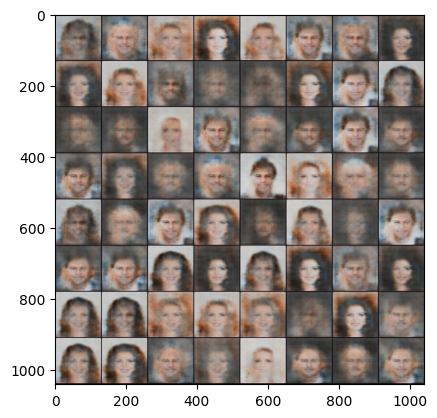
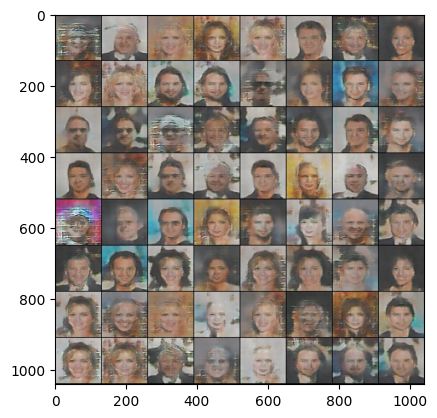
 

Fig.3: .Epoch 1 Fig. 4: .Epoch 10

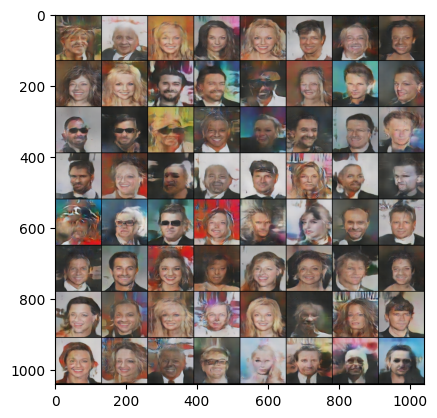
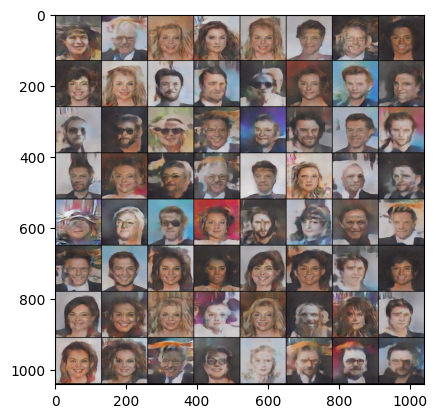


Fig.4: .Epoch 20 Fig.5: .Epoch 30

Losses for both the discriminator and generator are recorded during training. The training progress is monitored using a progress bar (`tqdm`).

Optionally, generated images are visualized during training to monitor the progress of image generation.

The training loop runs for a specified number of epochs (defined by `cfg.epochs`), and at the end of each epoch, it may visualize the current state of generated images using the `plot\_output` function.

* 1. **Deployment**

The final model deployed on an Anvil server, hosted on Google Colab for easy accessibility.

Our deployed solution offers a user-friendly interface for generating face images based on user-provided text prompts. The core of our system is the generate\_face function, which is callable via the Anvil server. This function takes a text\_prompt as input, where users describe the desired facial features or attributes they want to see in the generated image.

The provided text\_prompt is processed through our sentence\_encoder, which converts the text into embeddings. These embeddings capture the semantic meaning of the text, enabling the model to align the generated image with the textual description effectively.

Then we load back our saved DCGAN model by using loaded\_model. The loaded\_model combines the random noise and text embeddings to create an image that closely matches the textual description. This generated image aims to represent the facial features described in the text prompt.

We employ the Matplotlib library to process and display the generated image. The generated image is saved as a PNG format image in memory using the BytesIO object. The PNG image is further encoded as a base64 string. This encoding step prepares the image for efficient transmission over a network.

Finally, the base64-encoded image is returned as the output of the generate\_face function. Users can access the generated image, which closely matches their textual description.

* 1. **Result And Analysis** The training was done over 30 epochs, with a learning rate of 0.0002 for both generator and discriminator.

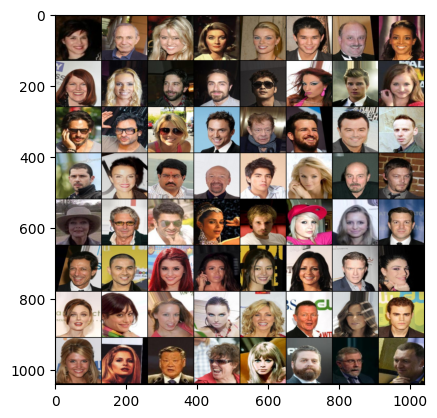


Fig.6: .Real Images Batch

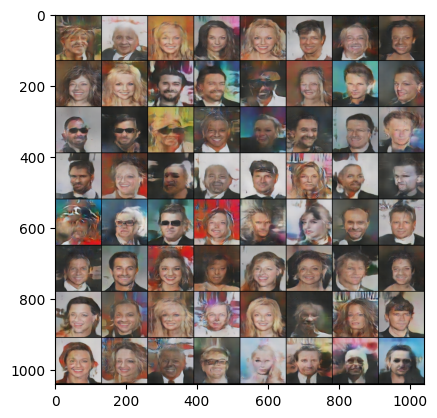


Fig7:Generated Images Batch of Our Model

Data augmentation techniques like random cropping and horizontal flipping were used to enhance the model's robustness. The generated images showed the model could capture main attributes but sometimes lacked fine details. The experiment suggests that, better pre-trained sentence encoders, and more advanced GAN architectures could improve image quality.

We compared our implemented improved model with its based model demonstrated in [paper]. And in a very small epoch run of 30 iterations we can see a significant improvement demonstrated by the Face Domain Specific Bert Based Model with Improve DCGAN model.

| **Text Input** | **Generated Image** | |
| --- | --- | --- |
|  | Bert-Based Model With Normal DCGAN | Domain Specific Bert- Based Model With Improve DCGAN |
| A young woman with striking brown eyes, long lashes, and perfect arched eyebrows. Her hair is a rich chestnut, and she has plump, coral-colored lips. Her laughter is infectious, and she wears minimal makeup, radiating confidence |  |  |
| A young man with a square face and a thick, well-groomed mustache. He sports a modern undercut hairstyle and has deep-set, contemplative brown eyes. His lips are full, and he often has a subtle, mysterious smirk. |  |  |

1. **Conclusion**

**In conclusion, our research has successfully explored the synergistic integration of a DCGAN with an improved BERT embedding model for the generation of realistic face images based on textual descriptions. Leveraging the extensive CelebA dataset and employing meticulous data collection and preprocessing, we established a robust foundation for training our DCGAN. The utilization of the bert-base-uncased model for sentence embeddings, enhanced through Masked Language Model pre-training, demonstrated its effectiveness in capturing intricate facial features from diverse textual descriptions.**

**The DCGAN architecture, meticulously designed with spectral normalization, batch normalization, and proper weight initialization, showcased its capability to generate high-quality facial images with stability and realism. Through a comprehensive training regimen spanning 30 epochs, we observed the progressive improvement in generated images, as evidenced by the visualizations and batch comparisons.**

**Limitations and future scope :**

**Thus to overcome the instability and challenges faced by DCGAN’S generative models we fine-tune a bert model embedding to increase its domain understanding of face textual data. resulting in better accuracy and realistic image generation than state of art technique of using DCGANs with generalized bert model.**

The challenging task of text-to-face generation, specifically focusing on generating facial images from textual descriptions has significant implications across various domains, including law enforcement, entertainment, and virtual reality. The limited availability of datasets for this task has prompted the development of innovative solutions. Our approach combines the power of Deep Convolutional Generative Adversarial Networks (DCGANs) and BERT-based text encoders to bridge the gap between language and imagery.

We utilized the CelebA dataset, containing a diverse collection of labeled face images, as our primary data source. Data preprocessing involved image resizing, conversion to tensors, and normalization, paving the way for subsequent model training. The integration of BERT-based embeddings into our SentenceEncoder class enabled the conversion of textual descriptions into meaningful vector representations.

We combine DCGANs and BERT embeddings to convert text descriptions into realistic facial images. In conclusion, this research contributes to the advancement of text-to-image generation, particularly for generating lifelike facial images from text inputs, with broad applications in diverse fields

# **References**

[1] Zia, Tehseen, Shahan Arif, Shakeeb Murtaza, and Mirza Ahsan Ullah. "Text-to-image generation with attention based recurrent neural networks." arXiv preprint arXiv:2001.06658 (2020).

[2] Shirin Nasr Esfahani, Shahram Latifi, "Image Generation with Gans-based Techniques: A Survey," International Journal of Computer Science and Information Technology, 2019.

[3] Zhu, Wenwu & Wang, Xin & Gao, Wen. (2020). Multimedia Intelligence: When Multimedia Meets Artificial Intelligence. IEEE Transactions on Multimedia. PP. 1-1. 10.1109/TMM.2020.2969791.

[4] Reed, Scott, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. "Generative adversarial text to image synthesis." In *International conference on machine learning*, pp. 1060-1069. PMLR, 2016.

[5] Crowson, Katherine & Biderman, Stella & Kornis, Daniel & Stander, Dashiell & Hallahan, Eric & Castricato, Louis & Raff, Edward. (2022). VQGAN-CLIP: Open Domain Image Generation and Editing with Natural Language Guidance.

[6] Zhang, Han, Jing Yu Koh, Jason Baldridge, Honglak Lee, and Yinfei Yang. "Cross-modal contrastive learning for text-to-image generation." In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 833-842. 2021.

[7] Wang, Lu-Yao, and Yi-Ping He. "Environmental landscape art design based on visual neural network model in rural construction." *Ecological Chemistry and Engineering S* 30, no. 2 (2023): 267-274.

[8] Ramesh, Aditya, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. "Zero-shot text-to-image generation." In *International Conference on Machine Learning*, pp. 8821-8831. PMLR, 2021.

[9] Ramesh, Aditya, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. "Hierarchical text-conditional image generation with clip latents." *arXiv preprint arXiv:2204.06125* 1, no. 2 (2022): 3.

[10] Gu, Shuyang, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. "Vector quantized diffusion model for text-to-image synthesis." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10696-10706. 2022.

Joseph Babcock, Raghav Bali, "Generative AI with Python and TensorFlow 2: Create images, text, and music with VAEs, GANs, LSTMs, Transformer models," Packt Publishing Limited, 2021.

[3]Li, Bowen, Xiaojuan Qi, Thomas Lukasiewicz, Philip Torr, "ManiGAN: Text-Guided Image Manipulation," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[4] Shoya Matsumori, Yuki Abe, Kosuke Shingyouchi, Komei Sugiura, Michita Imai, "LatteGAN: Visually Guided Language Attention for Multi-Turn Text-Conditioned Image Manipulation," IEEE Access, 2021.

[5] [6]Jian Zhao, Xie Xie, Lin Wang, Meng Cao, Miao Zhang, "Generating Photographic Faces From the Sketch Guided by Attribute Using GAN," IEEE Access, 2019.

[7] Zixu Wang, Zhe Quan, Zhi-Jie Wang, Xinjian Hu, Yangyang Chen, "Text to Image Synthesis With Bidirectional Generative Adversarial Network," 2020 IEEE International Conference on Multimedia and Expo (ICME), 2020.

[**17**]Anukriti Kumar, Anurag Mudgil, Nakul Dodeja, Dinesh Kumar Vishwakarma, "Realistic face generation using a textual description," 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), 2021.

[**14**]Muhammad Zeeshan Khan, Saira Jabeen, Muhammad Usman Ghani Khan, Tanzila Saba, Asim Rehmat, Amjad Rehman, Usman Tariq, "A Realistic Image Generation of Face From Text Description Using the Fully Trained Generative Adversarial Networks," IEEE Access, 2021

[**12**] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, Dimitris Metaxas, "StackGAN++: Realistic Image Synthesis with Stacked Generative Adversarial Networks," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2019.

[11] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, Ilya Sutskever, "Zero-Shot Text-to-Image Generation," International Conference on Machine Learning, 2021

[12]Mukesh Jha, Sahil Jobanputra, Divya Kamath, Varsha Hole, "Creation of Face for Aiding in Forensic Investigation based on Textual Description," 2020 Fourth International Conference on Inventive Systems and Control (ICISC), 2020.

[**15**] Tero Karras, Samuli Laine, Timo Aila, "A Style-Based Generator Architecture for Generative Adversarial Networks," International Conference on Machine Learning, 2019.

[**13**] Tianren Wang, Teng Zhang, Brian Lovell, "Faces à la Carte: Text-to-Face Generation via Attribute Disentanglement," 2021 IEEE Winter Conference on Applications of Computer Vision (WACV), 2021.

[15] Jichao Zhang, Aliaksandr Siarohin, Yahui Liu, Hao Tang, Nicu Sebe, Wei Wang, "Training and tuning generative neural radiance fields for attribute-conditional 3D-aware face generation," arXiv preprint arXiv:2208.12550, 2022.

[16] Gary Marcus, Ernest Davis, Scott Aaronson, "A very preliminary analysis of DALL-E 2," arXiv preprint arXiv:2204.13807, 2022.

[17]Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, Illia Polosukhin, "Attention is all you need," Advances in neural information processing systems, 2017.

[18] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, Luke Zettlemoyer, "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," arXiv preprint arXiv:1910.13461, 2019.

[19]Faezeh Gholamrezaei, Mohammad Manthouri, "cycle text2face: cycle text-to-face gan via transformers," arXiv preprint arXiv:2206.04503, 2022.

[**18**]Wenwu Zhu, Xin Wang, Wen Gao, "Multimedia Intelligence: When Multimedia Meets Artificial Intelligence," IEEE Transactions on Multimedia, 2020.

[21] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, Honglak Lee, "Generative adversarial text to image synthesis," International conference on machine learning, 2016.