Facial Image Generation from Bangla Textual Description using DCGAN and Bangla FastText

Noor Mairukh Khan Arnob, Nakiba Nuren Rahman, Saiyara Mahmud,
Md. Nahiyan Uddin, Rashik Rahman*, Aloke Kumar Saha*
Computer Science and Engineering, University of Asia Pacific, Dhaka, Bangladesh

***Abstract*—The synthesis of facial images from textual descriptions is a relatively difficult subfield of text-to-image synthesis. It is applicable in various domains like Forensic Science, Game Development, Animation, Digital Marketing, and Metaverse. However, no work was found that generates facial images from textual descriptions in Bangla; the 5th most spoken language in the world. This research introduces the first-ever system to generate facial images from Bangla textual descriptions. The proposed model comprises two fundamental constituents, namely a textual encoder, and a Generative Adversarial Network(GAN). The text encoder is a pre-trained Bangla text encoder named Bangla FastText which is employed to transform Bangla text into a latent vector representation. The utilization of Deep Convolutional GAN (DCGAN) allows for the generation of face images that correspond to text embedding. Furthermore, a Bangla version of the CelebA dataset, CelebA Bangla is created for this study to develop the proposed system. CelebA Bangla contains images of celebrities, their corresponding annotated Bangla facial attributes and Bangla Textual Descriptions generated using a novel description generation algorithm. The proposed system attained a Fréchet Inception Distance (FID) score of 126.708, Inception Score(IS) of 12.361, and Face Semantic Distance(FSD) of 20.23. The novel text embedding strategy used in this study outperforms prior work. A thorough qualitative and quantitative analysis demonstrates the superior performance of the proposed system over other experimental systems.**

***Keywords*—Bangla text-to-face synthesis; Natural Language Processing(NLP); Computer Vision(CV); GAN; text encoders**

I. Introduction

Generative Adversarial Networks (GANs) have been identified as an effective tool for producing lifelike images in diverse domains, encompassing natural landscapes and human countenances. The capacity to produce superior images from textual depictions has garnered considerable interest owing to its potential implications in virtual avatars, content creation, and tailored advertising.

The generation of an image from a given textual input is referred to as text-to-image generation. The Text-To-Face (TTF) technique is a subfield of the Text-To-Image (TTI) generation field, wherein a depiction of a human face is furnished, and a facial image is produced by utilizing the description. Generating images of faces is a more challenging task compared to text-to-image generation, primarily due to the intricate nature of facial attributes. The utilization of text-to-face synthesis holds significant potential in various practical domains, such as Forensic Science, Game Development, Animation, Digital Marketing, and the Metaverse. The generation of images and faces from text has emerged as a prominent area of research in recent times, resulting in a substantial body of literature on the subject. Notably, a majority of scholars have directed their attention towards image generation in the English language [1]. Although significant advancements have been achieved in the field of English-based text-to-image synthesis, there has been a dearth of research pertaining to non-English languages, specifically the Bangla language. The Bangla language possesses distinct linguistic and cultural subtleties, thereby posing distinctive obstacles to the synthesis of text-to-face. The process of generating facial images from Bangla text necessitates a profound comprehension of the language's phonological, syntactic, and semantic frameworks. The accurate representation of the visual heterogeneity and distinctive facial attributes of Bangla-speaking individuals is imperative in producing genuine and culturally appropriate facial depictions.

To mitigate this gap, in this paper, a novel GAN-based system, specifically for generating face images from Bangla textual input is proposed. The objective of the proposed system is to mitigate the challenges related to Bangla text and cultural diversity in multimodal synthesis research, thereby filling an existing gap in this field. The proposed model consists of two primary components, namely a text encoder and an image generator. The utilization of Bangla FastText [2] by the text encoder serves the purpose of encoding Bangla text into a latent vector representation that adeptly captures the semantic information that is intrinsic to the text. Subsequently, the image generator utilizes a Deep Convolutional Generative Adversarial Network (DCGAN) architecture to produce facial images that align with the encoded textual depiction. Modifications have been made to the CelebA dataset [3] to enhance the efficiency of our model's training and evaluation processes. Labels have been meticulously assigned to 40 distinct facial attributes using semantically accurate Bangla vocabulary. This has led to the creation of a novel dataset named CelebA Bangla. The CelebA Bangla dataset is a compilation of face images showcasing celebrities, accompanied by 40 facial attribute annotations in the Bangla language. Utilizing these attributes, textual depictions of Bangla faces are generated through our novel algorithm for Bangla facial description generation. By conducting thorough experimentation and utilizing both quantitative and qualitative evaluation metrics, the quality, diversity, and fidelity of the produced facial images are evaluted.Then, the performance of our proposed model is compared to that of the current leading models. The proposed system attained a Frechet Inception Distance (FID) score of 126.708, Inception Score(IS) of 12.361, and Face Semantic Distance(FSD) of 20.23.

*Corresponding authors

The major contributions of this paper are:

- A novel version of the CelebA dataset has been proposed entitled CelebA Bangla.
- A novel system for generating facial images from Bangla text descriptions has been developed, whereby meaningful images are produced in response to input in the Bangla language. The system under consideration attained an FID score of 126.708.

The subsequent segments of the document are structured in the following manner: Section II discusses the related works. The dataset is elaborated upon in Section III, while Section IV provides an overview of the methodology employed. Section V presents a thorough analysis of the qualitative and quantitative outcomes. Section VI establishes the limitations or constraints of the study followed by Section VIII containing the conclusion. The remaining portion comprises references.

II. Related Work

In this section, significant works of state-of-the-art Generative Adversarial Networks, text encoders, text-to-image, and face synthesis architectures are analyzed.

A. GANs

Generative Adversarial Network (GAN) [4] is an exceptional framework that can learn to generate new data based on the data of a specified training set. GANs are composed of two parts, the Generator and Discriminator respectively. There is a constant competition between these two parts where the generative network generates new data learning to map from a latent space of data distribution while the discriminative network differentiates the data produced by the generator from the actual data distribution. Deep Convolutional Generative Adversarial Network (DCGAN) [5] is an extension of GAN that incorporates convolutional and convolutional-transpose layers in the generator and discriminator accordingly. Self-Attention Generative Adversarial Network (SAGAN) [6] provides attention-driven modeling of long-range dependencies for image generation activities where its discriminator can verify the consistency of highly detailed features in distant portions of the image and attention mechanism can provide the generator and discriminator with more power to directly model the long-range dependencies in the feature maps and better approximate the original image's distribution.

Attentional Generative Adversarial Network (AttnGAN) enables multi-stage, attention-driven image generation from textual description [7], [8]. AttnGAN begins with a rudimentary low-resolution image which it then refines in multiple phases to produce a final image from the natural language description. StyleGAN [9]–[11] is another extension of the progressive GANs that enables generation of high-quality photorealistic images by means of the incremental development of discriminator and generator models beginning with a low resolution and expanding to a high resolution of 1024x1024 pixels. GigaGAN* synthesizes high-resolution images, such as ultra-high 4k resolution images in 3.66 seconds, and supports a variety of latent space editing options including latent interpolation, style blending, and vector arithmetic operations.

*https://github.com/lucidrains/gigagan-pytorch

B. Text to Image Synthesis

In paper [12], they proposed an efficient deep GAN architecture-based text-to-image synthesis of birds and flowers images from human-written descriptions. They utilized the Caltech-UCSD Birds dataset (CUB), Oxford-102, and MS COCO dataset to train and evaluate their model. Their proposed model showed substantial improvements in Text to image synthesis. Later on, the paper [7] suggested the first Bangla language-based Text-to-image generation method AttnGAN that analyzed Deep Attentional Multimodal Similarity Model and Attentional GAN to generate improved and realistic high-resolution images from Bangla text description surpassing the state-of-the-art (SOTA) image synthesis GAN models by an ideal inception score of 3.58 ± .06.

The author of [8], presented AttnGANTRANS which consists of Attentional GAN and transformer models such as Bidirectional Encoder Representations from Transformers (BERT), GPT2, and XLNet that were capable of extracting semantic information from text descriptions more accurately than the conventional AttnGAN. Gao *et al.* [13] proposed LD-CGAN comprised of one generator and two independent discriminators to regularize and generate 64x64 and 128x128 images. The generator includes three major components- Conditional Embedding (CE) which disentangles integrated semantic attributes in the text, Conditional Manipulating Modular (CM-M) used to continuously provide image features with compensation information and Pyramid Attention Refine Block (PAR-B) to enrich multi-scale features. The experiments were evaluated on CUB and Oxford-102 datasets achieving an Inception score of 3.64 ± 0.04 and 4.18 ± 0.06 on 64x64 and 128x128 images.

Zhang *et al.* [14], presented XMC-GAN comprised of several contrastive losses, an attentional self-modulation generator, and a contrastive discriminator to generate images of higher quality and closer correspondence to the input descriptions which was evaluated on three datasets demonstrating SOTA FID score of 9.33 on the MS-COCO dataset and an impressive benchmark FID score of 26.91 on the Open Image Data. Siddharth *et al.* [15] proposed AttnGAN with pre-trained text encoder RoBERTa using the Caltech-UCSD birds dataset for textual descriptions obtaining an FID score of 20.77.

The authors of [1], [16], [17] suggested DF-GAN that can directly synthesize high-resolution images without entanglements between different generators, improve TTI semantic coherence and make complete integration between text and synthesized features that was evaluated on the CUB and COCO datasets where yielded results surpassed SOTA models.

C. Text to Face Synthesis

It is a very challenging task to convert human-written descriptions into human faces. But many types of research [1], [9], [18]–[22] have been conducted in this field of Text-to-face synthesis.

Deorukhkar *et al.* [1], proposed to use Sentence Bidirectional Encoder Representations from Transformers (SBERT) to convert the textual descriptions (from their own dataset based on CelebA dataset) into embeddings and generated 128x128 sized images using DCGAN, SAGAN and DFGAN models. Recently, StyleGAN-based models [9], [19] have

advanced Text-to-face synthesis in terms of image quality and diversity. The author of [9] introduced a Multi-Modal CelebA-HQ dataset. They also introduced a framework containing the GAN inversion technique based on the multi-modal inputs. Finally, evaluating the model on the Multi-Modal CelebA-HQ dataset, they achieved an FID score of 106.37 and generated 1024x1024 sized images. The authors of [19] presented a two-stream framework combining CLIP visual concepts and StyleGAN using Multi-Modal CelebA-HQ and CelebAText-HQ [23] datasets for high-fidelity Text-to-face synthesis. Later, they evaluated the model on two datasets including the Multimodal CelebA-HQ dataset and the CelebAText-HQ dataset, and finally, achieved an FID score of 50.56 and 56.75, respectively.

Recently, StyleGAN2-based models [18], [20] have been introduced in the field of Text-to-face synthesis. The authors of [18], used a Text-to-face framework with StyleGAN2 and a sentence encoder named BERT and generated 1024x1024-shaped high-quality images. In the paper [20], they proposed a TTF-HD framework with StyleGAN2 using the CelebA dataset in order to generate high-quality facial images with a wide range of variations leading to generating 1024x1024 sized images.

Peng *et al.* [21] introduced a dynamic pixel synthesis network that can transform text features into dynamic knowledge embeddings and generate accurate Text-to-face images that were trained and evaluated on the Multi-Modal CelebA-HQ dataset achieving an excellent FID score of 13.48. The authors of [22], proposed a GAN model which can directly convert the text descriptions into pixel values. They conducted zero-shot experiments on Face2Text [24] then trained and evaluated their proposed model on Multi-Modal CelebA-HQ and managed to achieve an FID score of 14.45.

There are many existing works on English text-to-face synthesis, as discussed in this sub-section. However, there is no research work done on Bangla text-to-face synthesis. There are also some limitations in the prior English text-to-face synthesis works. For instance, when the textual descriptions of faces are long, some of the works failed in handling those long descriptions of faces. As a result, the models could not generate accurate facial images. In some other works the models are not robust enough, so the generated images do not match with the input text descriptions.

III. Dataset

The present study employs the CelebA dataset, which was introduced by the authors of [3]. The dataset comprises in excess of 200,000 images of celebrities' faces, each with a resolution of 128x128 pixels. Additionally, it contains annotations (in English) of 40 facial attributes for each image.

Nonetheless, the CelebA dataset is inadequate for creating a system that utilizes Bangla facial description as input and produces the corresponding image. Consequently, a novel iteration of the CelebA dataset, titled CelebA Bangla, has been created and presented in this paper. The proposed dataset consists of three distinct segments. The initial segment depicts a collection of images of notable celebrities, followed by a list of 40 attributes that have been manually annotated in the Bengali language. The third segment pertains to the Bangla

TABLE I. Facial Attribute Sample of the Proposed CelebA Bangla Dataset

image_file_name	হালকা_দাড়ি (5_o_Clock Shadow)	কুচকানো_ভ্রু (arched eyebrows)	আকর্ষণীয় (attractive)	.....	অল্পবয়স্ক (young)
 000001.jpg	-1	1	1	.....	1
 000002.jpg	-1	-1	-1	.....	1
....					
 202599.jpg	-1	1	1	.....	1

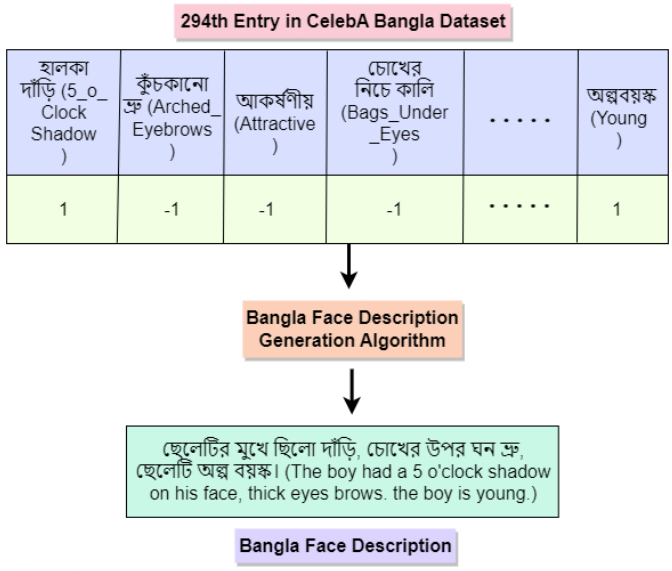

Fig. 1. Proposed Facial Description Generation Process.

descriptions, which are obtained from the second segment. Therefore, it can be observed that every image is associated with 40 distinct Bangla facial attributes and a corresponding Bangla facial description.

Table I represents 40 facial attributes corresponding to a single image. Here, the first column shows the image of celebrities and each of the following columns is facial attributes in Bangla. Considering i as row and j as the column

of Table I, if $TableI[i][j] = 1$, it implies that attribute j is present in image i. Otherwise, $TableI[i][j] = -1$ means attribute j is absent in image i. Face attributes were manually annotated into the most suitable Bangla attributes.

TABLE II. BANGLA TEXT DESCRIPTION SAMPLES OF THE PROPOSED CELEBA BANGLA DATASET

image_file_name	text_description
 000001.jpg	মেয়েটির ভ্রু কুচকানো ছিল। মেয়েটির সোনালী চুল ছিল। মেয়েটির মুখে ভারী মেকাপ ছিল। মেয়েটির উঁচু গালের হাড় ছিল। মেয়েটির মুখ কিছুটা খোলা ছিল। মেয়েটির চোখা নাক ছিল। মেয়েটির মুখে ছিল হাসি। মেয়েটির সোজা চুল ছিল। মেয়েটির কানে দুল পরা ছিল। মেয়েটির লিপস্টিক পরা ছিল। (The lady had arched eyebrows. She had blonde hair. She was wearing heavy makeup. The lady had high cheekbones. Her mouth was slightly open. She had a pointy nose. The lady was smiling. She was wearing earrings and lipstick.)
 000002.jpg	মেয়েটির চোখের নিচে কালি ছিল। মেয়েটির বড় নাক ছিল। মেয়েটির সোনালী চুল ছিল। মেয়েটির উঁচু গালের হাড় ছিল। মেয়েটির মুখ কিছুটা খোলা ছিল। মেয়েটির মুখে ছিল হাসি। (The woman had bags under eyes. She had a big nose. She had blonde hair. The woman had high cheekbones and her mouth was slightly open. She had a smile on her face.)
	....
 202599.jpg	মেয়েটির ভ্রু কুচকানো ছিল। মেয়েটির সোনালী চুল ছিল। মেয়েটির মুখে ভারী মেকাপ ছিল। মেয়েটির চেহারা ফ্যাকাশে। মেয়েটির চোখা নাক ছিল। মেয়েটির ঢেউ খেলানো চুল ছিল। মেয়েটির লিপস্টিক পরা ছিল। (The woman has arched eyebrows. She has blonde hair, heavy makeup and pale skin. Her nose is pointy. She has wavy hair. The woman was wearing lipstick.)

Algorithm 1 is utilized to generate a Bangla description for each image based on its facial attributes. Algorithm 1 depicts a partial segment of the comprehensive algorithm for generating Bangla facial descriptions. This particular segment outlines the process for generating textual descriptions pertaining to males and females of varying ages. The gender and age attributes of the dataset are represented by row[22] and row[41], respectively. The Bangla text descriptions in Table II have been generated through the utilization of the suggested description generation algorithm. In order to produce significant textual depictions in Bangla, the algorithm receives annotated Bangla attributes. Subsequently, the Bangla text description generation algorithm generates semantically accurate Bangla text descriptions that correspond to the images of faces. The aforementioned procedure is depicted in Fig. 1.

IV. METHODOLOGY

The proposed system utilized DCGAN+Bangla Fasttext to generate face images from the corresponding Bangla descriptions. Firstly the Bangla text description is fed to Bangla Fasttext [2] which returns a $[300 \times 1]$ shaped text embedding. A random noise vector with a shape of $[100 \times 1]$ along with the achieved text embedding is passed to the generator. The generator generates images with a resolution of 128x128, then the discriminator detects whether the generated images are real or fake by comparing the generated images with ground truth images. Based on the difference between generated and ground truth images the loss is calculated and is back-propagated through the generator and discriminator as shown in Fig. 2.

Algorithm 1 Bangla Face Description Generation Algorithm

```
CelebABangla        ▷ Dataset containing Bangla attributes
attributes          ▷ 40 facial attributes
for row in CelebABangla do
    description ← ""        ▷ textual description of face
    if row[22]==1 then
        gender ← "male"
    else
        gender ← "female"
    end if
    if row[41]==1 then
        age ← "old"
    else
        age ← "young"
    end if
    for i = 0 to length(attributes) do
        if gender=="male" then
            if age=="young" then
 description.append(YoungMaleSentence(attribute[i]))
            else
 description.append(OldMaleSentence(attribute[i]))
            end if
        end if
        if gender=="female" then
            if age=='young' then
 description.append(YoungFemaleSentence(attribute[i]))
            else
 description.append(OldFemaleSentence(attribute[i]))
            end if
        end if
    end for
end for
```

A. Embedding Strategy

The proposed dataset comprises image-text pairs, where a single pair contains textual description T of the facial image I. In this study, a text encoder denoted as TE was employed in conjunction with a Deep Convolutional Generative Adversarial Network ($DCGAN$). The text T is passed through TE in order to obtain the corresponding text embedding E. Ultimately, the $DCGAN$ was trained through the utilization of I and E.

$$E = TE(T) \tag{1}$$

In FGTD [1], text embeddings were generated for a given text description T consisting of a series of sentences S_i, and embeddings E_{Si} were computed, where $i \in \mathbb{N}$ and $\mathbb{N}$ is the set of all natural numbers. The arithmetic average of the embedding vectors was utilized as input for the conditional GAN, as depicted in Fig. 3(a). However, it was anticipated that the process of obtaining the mean of embeddings results in a reduction of significant semantic information that is initially present in the sentence embeddings, E_{Si}. In order to mitigate the loss of information, our proposed embedding strategy (as depicted in Fig. 3(b)) which involves the utilization of a text encoder to generate a text embedding, E, by processing the complete textual description T as shown in Equation 1. Thus a text embedding is obtained without losing semantic information caused by the mean operation. Since E_T is an

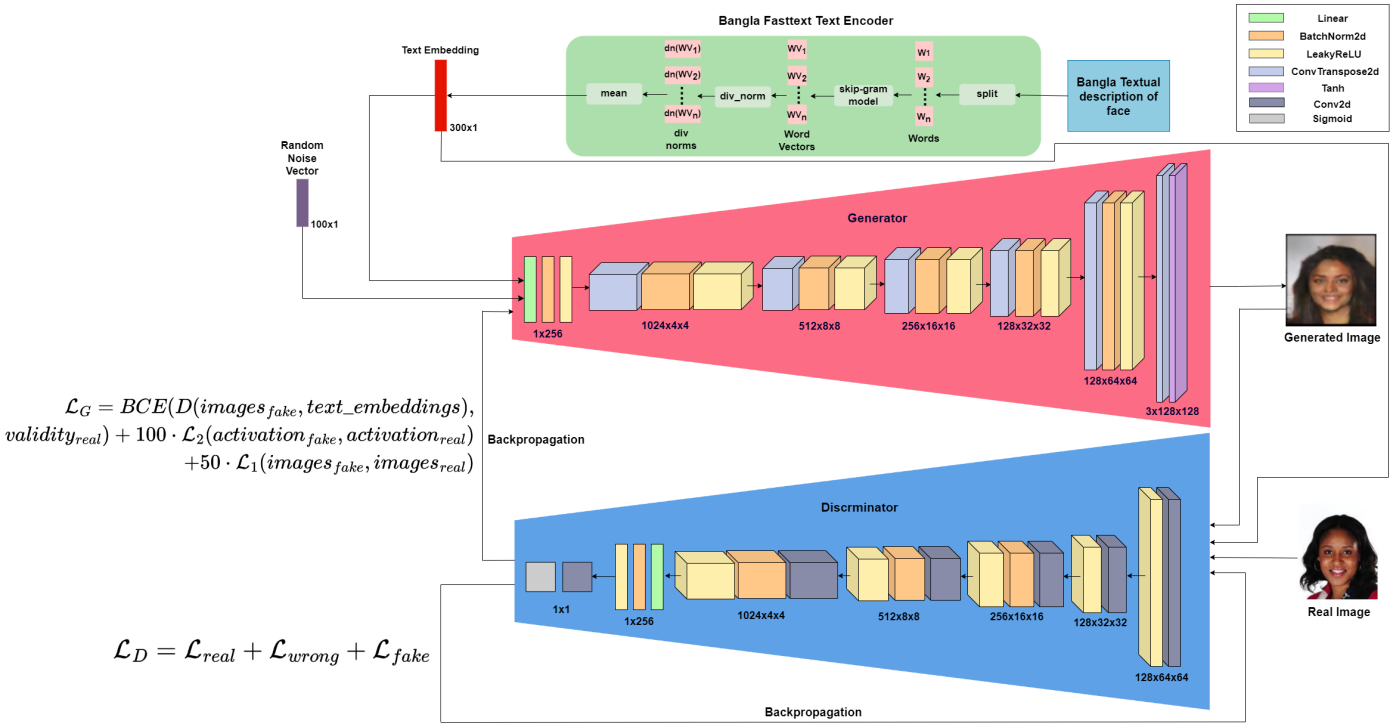

Fig. 2. Neural architecture of the proposed system: Bangla Fasttext + DCGAN.

embedding of the entire textual description T, it has a better one-to-one correlation between the text description and the final embedding. E_T is directly passed to DCGAN. Within the section pertaining to quantitative results, it is demonstrated that the proposed text embedding strategy exhibits superior performance in comparison to the strategy employed in FGTD [1].

B. Text Encoder

A work [8] from 2021 suggested that using a pre-trained text encoder enhances the performance of text-to-image synthesis. The proposed system employs Bangla FastText [2] as pretrained text encoder. Bangla FastText is trained using 20 million Bangla data. As demonstrated in Fig. 4(a), the Bangla Fasttext sentence encoder first splits the facial descriptions into words. It then computes word embeddings using a skip-gram model. Afterwards, the word embedding vectors go through an operation div_norm defined by Equation 2. div_norm essentially prevents a distribution from being dispersed by dividing a vector by its euclidian norm if the euclidian norm is greater than zero. Finally, the mean of div_norms is passed on as a $[300 \times 1]$ sentence embedding vector.

$$div\_norm(x) = \begin{cases} \frac{x}{\sqrt{\sum_{i=0}^{n} x_i^2}} & \text{if } \sqrt{\sum_{i=0}^{n} x_i^2} > 0 \\ x & \text{if } \sqrt{\sum_{i=0}^{n} x_i^2} \leq 0 \end{cases} \quad (2)$$

In our experimental models, two other pretrained Bangla text encoders were also used provided by sbnltk[†] : sbnltk sentence transformer hd (trained on 3,00,000+ human data) and sbnltk sentence transformer gd (trained on 3,00,000+ google translated data). Both of these models have the same neural architecture but were trained on different datasets. As depicted in Fig. 4(b), the sbnltk sentence transformer first generates tokens from sentences. The tokens are passed on to a pretrained multilingual model, XLM-RoBERTa [25] which has 12 hidden encoder layers. Finally, a pooling layer gives us the sentence embedding vector of length 768. Despite the superior neural architecture of XLM-RoBERTa, sbnltk sentence transformer is trained on lesser amount of Bangla text corpus, which may have led it to have learned an inadequate probabilistic distribution of text written in Bangla; compared to Bangla Fasttext. For this reason, Bangla FastText have been incorporated in our proposed system.

C. GAN Architecture

Our proposed method has DCGAN [5] as its GAN architecture. The generator of DCGAN has a sequence of transpose convolution, batch normalization, and LeakyReLU layers. In the end, a Tanh activation function gives us generated or fake images. Strided convolutions used in the generator allows the network to learn its own spatial upsampling. The Discriminator mainly has a sequence of blocks containing convolution, batch normalization, LeakyReLU layers, and a sigmoid activation in the end to classify real/fake images. The discriminator uses strided convolution to learn its own spatial downsampling. Batch normalization employed in both generator and discriminator normalizes the input to each unit to have zero mean and unit variance to stabilize the training process. The use

[†]https://github.com/Foysal87/sbnltk

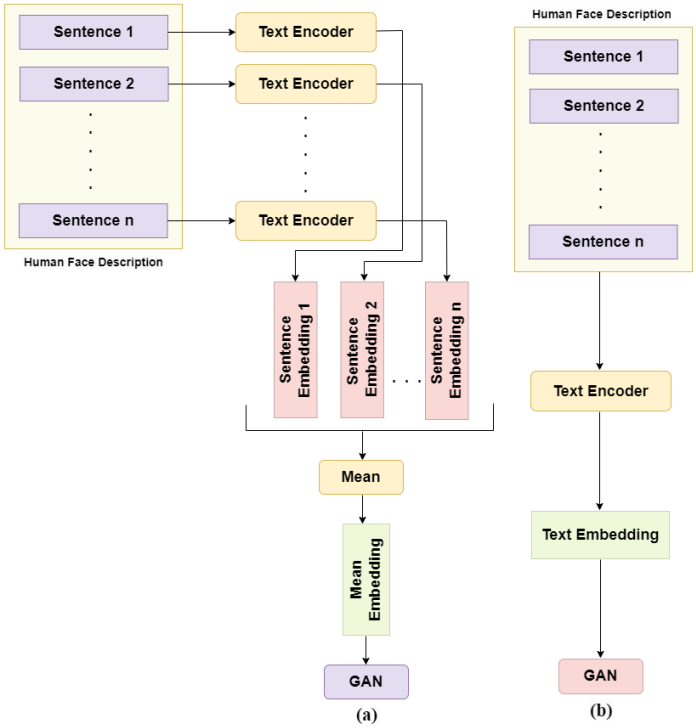

Fig. 3. (a) Text embedding strategy used in FGTD [1], (b) Proposed embedding strategy.

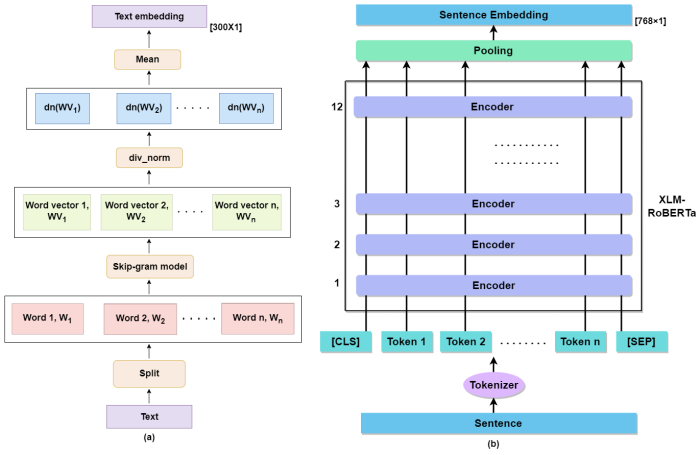

Fig. 4. (a) Bangla FastText architecture, (b) Sbnltk Sentence Transformer neural model.

of an unbounded activation, Leaky ReLU allows DCGAN to converge fast and learn the color space of the distribution of training images.

To train the Generator of DCGAN, Adam optimizer was utilized with learning rate, $\alpha = 0.0002$ and $\beta_1 = \beta_2 = 0.5$. For training the Discriminator of DCGAN, Adam optimizer was utilized with learning rate, $\alpha = 0.0001$ and $\beta_1 = \beta_2 = 0.5$.

D. Loss Functions

The loss function of the generator of our proposed DC-GAN+Bangla FastText method is shown in Equation 3.

$$\mathcal{L}_G = BCE(D(images_{fake}, text\_embeddings), validity_{real}) + 100 * \mathcal{L}_2(activation_{fake}, activation_{real}) + 50 * \mathcal{L}_1(images_{fake}, images_{real}) \quad (3)$$

Here BCE in Equation 3 is the Binary Cross Entropy Error. $images_{fake}$ are images generated from input noise and text embeddings passing through the Generator of DCGAN. $input\_noise$ is a 100-dimensional vector which comes from a standard normal distribution with mean 0 and variance 1. $text\_embeddings$ are generated from textual descriptions of faces which went through a text encoder (Equation 4). The dimensions of $text\_embeddings$ are $[300 \times 1]$. $validity_{real}$ is a vector where each element equals 1 (Equation 5). The dimensions of $validity_{real}$ are $[batch\_size \times 1]$. The BCE loss mentioned here uses discriminator D to assess how realistic the generated images are in response to text embeddings. Higher BCE loss penalizes the generator network more.

$$images_{fake} = Generator(input\_noise, text\_embeddings) \quad (4)$$

$$validity_{real} = \begin{bmatrix} 1 \\ 1 \\ . \\ . \\ . \\ 1 \end{bmatrix}_{[batch\_size \times 1]} \quad (5)$$

$\mathcal{L}_2$ loss is the Mean Square Error (MSE). By passing $images_{fake}$ and $text\_embeddings$ through the Discriminator of DCGAN in Equation 6, $activation_{fake}$ was obtained. By passing $images_{real}$ and $text\_embeddings$ through the Discriminator of DCGAN, $activation_{real}$ was obtained in Equation 7. The $\mathcal{L}_2$ loss is a comparison of how different the activations are from the discriminator with regards to real and fake images. Since this loss can potentially prove to be crucial in the training process, it is multiplied by 100 in Equation 3.

$$activation_{fake} = Discriminator(images_{fake}, text\_embeddings) \quad (6)$$

$$activation_{real} = Discriminator(images_{real}, text\_embeddings) \quad (7)$$

$\mathcal{L}_1$ loss is defined as the mean absolute error. in Equation 3, $\mathcal{L}_1$ loss measures how different the generated images are compared to real images. Since this loss has lower relevance than $\mathcal{L}_2$ loss and higher significance than BCE loss, it is given a multiplier 50 in Equation 3.

Performing a weighted sum of BCE loss, $\mathcal{L}_2$ loss and $\mathcal{L}_1$ loss in Equation 3 equips the generator of DCGAN with a

robust loss function to help it generate more realistic images which are semantically aligned with textual descriptions.

The loss function illustrated in Equation 8 was used to train the discriminator network of DCGAN.

$$\mathcal{L}_D = \mathcal{L}_{real} + \mathcal{L}_{wrong} + \mathcal{L}_{fake} \tag{8}$$

Where, For computing $\mathcal{L}_{real}$, $images_{real}$ and $text\_embeddings$ were passed through the Discriminator of DCGAN to get $output_{real}$ and $activation_{real}$ in Equation 9. $output_{real}$ is compared with $labels_{real}$ to compute BCE loss, which is our $\mathcal{L}_{real}$ loss(Equation 10). $labels_{real}$ are textual descriptions of faces corresponding to a batch of facial images. $\mathcal{L}_{real}$ loss essentially determines how close the outputs of the discriminator are compared to true labels.

$$output_{real}, activation_{real} = Discriminator(images_{real}, text\_embeddings) \tag{9}$$

$$\mathcal{L}_{real} = BCE(output_{real}, labels_{real}) \tag{10}$$

When calculating $\mathcal{L}_{wrong}$ in Equation 11, $validity_{fake}$, a vector of 1s and dimensions $[batch\_size \times 1]$ are taken. $output_{wrong}$ is obtained by passing $images_{wrong}$ and $text\_embeddings$ through the Discriminator of DCGAN. $images_{wrong}$ are images which are different from $images_{real}$ and do not correspond to $text\_embeddings$. BCE loss between $output_{wrong}$ and $validity_{fake}$ are calculated to acquire $\mathcal{L}_{wrong}$(Equation 12). The task of $\mathcal{L}_{wrong}$ is to ensure that the discriminator is classifying the wrong images correctly.

$$output_{wrong} = Discriminator(images_{wrong}, text\_embeddings) \tag{11}$$

$$\mathcal{L}_{wrong} = BCE(output_{wrong},\ validity_{fake}) \tag{12}$$

For the purpose of determining $\mathcal{L}_{fake}$, first $output_{fake}$ is obtained by passing $images_{fake}$ and $text\_embeddings$ through the Discriminator of DCGAN in Equation 13. In Equation 14, BCE loss between $output_{fake}$ and $validity_{fake}$ is calculated to get $\mathcal{L}_{fake}$. $\mathcal{L}_{fake}$ instructs the discriminator to classify fake images correctly.

$$output_{fake} = Discriminator(images_{fake}, text\_embeddings) \tag{13}$$

$$\mathcal{L}_{fake} = BCE(output_{fake}, validity_{fake}) \tag{14}$$

A linear combination of $\mathcal{L}_{real}$, $\mathcal{L}_{wrong}$ and $\mathcal{L}_{fake}$ in Equation 8 form a strong loss function to assist the discriminator of DCGAN to adjust its parameters to attain superior classification performance.

V. Result Analysis

In this section, a comprehensive discussion of the experimental details during training and validation of the proposed model is provided.

A. Experimental Setup

During the process of training and testing the model, the hardware configuration utilized comprised an Intel Core i7 7700K CPU, 16 GB of DDR4 RAM, and an Nvidia RTX 3060 GPU equipped with 12 GB of VRAM. The proposed system is implemented and developed using the Anaconda 22.11.1 environment, which runs on Windows 10 and has Python 3.9.15 installed.

Prior work related to text-to-face synthesis utilizes English text descriptions where they employ sBERT [1], Roberta [15], GPT2 and XLNet [8] etc. as text encoders. However, these text encoders are not usable when considering Bangla text description. Thus, in this study, Bangla FastText and sbnltk text encoders are used in combination with several GAN architectures to provide a comprehensive analysis of performance between our proposed system and other systems. Some details about the models used for comparison are presented in Table III where Model-1 is the porposed model. Keeping limited computational resources in mind, DCGAN [5] (30 Million Parameters), SAGAN [6] (18M Parameters) and DFGAN [16] (110M Parameters) architectures were chosen to perform the experiments. FGTD [1]'s implementation of DCGAN, SAGAN, and DFGAN were used in this research endeavor and sbnltk and Bangla FastText replaces the text encoder of FGTD to take Bangla text descriptions as input.

TABLE III. Configuration of Different Experimental Models

Experimental Models	Text encoder and GAN utilized	Batch size	VRAM consumption while training
Model-1 (Proposed system)	DCGAN + Bangla FastText	64	4.5 GB
Model-2	DCGAN + sbnltk HD	64	4.7 GB
Model-3	DCGAN + sbnltk GD	64	4.7 GB
Model-4	SAGAN + Bangla FastText	16	7.8 GB
Model-5	SAGAN + sbnltk HD	16	9 GB
Model-6	SAGAN + sbnltk GD	16	9 GB
Model-7	DFGAN + sbnltk GD	8	10 GB

To train the Generator of SAGAN, Adam optimizer was used with learning rate, $\alpha = 0.0001$ and $\beta_1 = 0$, $\beta_2 = 0.9$. For training the Discriminator of SAGAN, Adam optimizer was utilised with learning rate, $\alpha = 0.0004$ and $\beta_1 = 0$, $\beta_2 = 0.9$. For training the Generator of DFGAN, Adam optimizer was utilised with learning rate, $\alpha = 0.0001$ and $\beta_1 = 0$, $\beta_2 = 0.9$. For training the Discriminator of DFGAN, Adam optimizer was utilised with learning rate, $\alpha = 0.0004$ and $\beta_1 = 0$, $\beta_2 = 0.9$.

The aforementioned GAN architectures were paired with Bangla FastText [2], sbnltk sentence transformer HumanTranslated Data (HD), and sbnltk sentence transformer GoogleTranslated Data (GD) text encoders.

B. Evaluation Metrics

To evaluate our models, 5 different conventional evaluation metrics were utilised including Inception score (IS), Fréchet Inception distance (FID), Learned Perceptual Image Patch Similarity (LPIPS), Face Semantic Similarity (FSS) and Face Semantic Distance (FSD).

1) Inception score (IS): IS is used to measure the quality and diversity of the generated images where a higher score of IS suggests that the generated images are of high quality and diverse. IS is calculated using Equation 15.

$$\mathcal{I}nception\ Score = \exp\left(\mathbb{E}_x KL\left(p(y|x)||p(y)\right)\right) \quad (15)$$

Here, $p(y|x)$ is the conditional class distribution of the generated images, $p(y)$ is the marginal class distribution of the generated images, and KL is the Kullback-Leibler Divergence. PyTorch ignite's implementation[‡] of inception score was used.

2) Fréchet Inception Distance (FID): FID compares the similarity of generated images to the real ones. FID is a more accurate performance metric compared to IS and unlike IS, a lower FID score means that the generated images are more similar to the real images. FID is calculated using Equation 16.

$$d^2 = \|\mu_X - \mu_Y\|^2 + \mathrm{Tr}(\Sigma_X + \Sigma_Y - 2\sqrt{\Sigma_X \Sigma_Y}) \quad (16)$$

Where d^2 indicates the distance has squared units. μ_X is the feature-wise mean of the real image. μ_Y indicates the feature-wise mean of the generated image. Σ_X is the covariance matrix of the feature vector of the real image. Σ_Y is the covariance matrix of the feature vector of the generated image. Trace linear algebra operation is indicated by Tr.

3) Learned Perceptual Image Patch Similarity (LPIPS): LPIPS essentially measures the similarity between the activations of two image patches where the two images are the real image and the generated image, respectively. Like FID [26] score, a lower LPIPS score indicates that image patches are perceptually similar. The formula used to calculate LPIPS is in equation 17.

$$d(x, x_0) = \sum_l \frac{1}{H_l W_l} \sum_{h,w} \left\| \omega_l \bigodot (\hat{y}^l_{hw} - \hat{y}^l_{0hw}) \right\|_2^2 \quad (17)$$

Where d is the LPIPS distance between real image x and generated image x_0. Features are extracted from layer l of Alexnet. In the channel dimension, unit normalization is applied. $\bigodot$ indicates the Hadamard product. The number of activated channels are scaled by the vector ω_1. For calculating LPIPS, lpips-pytorch[§] was used to evaluate the generated images using a pre-trained Alexnet model.

4) Face Semantic Similarity (FSS): FSS measures the similarity between the generated face and the real face with regard to their facial features. In the case of FSS, a higher score of FSS means that the images are more similar. Equation 18 is utilized to compute FSS.

$$\mathcal{F}SS = \frac{1}{N} \sum_{i=0}^{N} cos(Facenet(F_{Gi}) - Facenet(F_G T_i)) \quad (18)$$

5) Face Semantic Distance (FSD): FSD is used to measure the dissimilarity between the generated face and the real face with respect to their facial features. Regarding FSD, a lower score implies that images are more similar. The formula for determining FSD is presented in Equation 19.

$$\mathcal{F}SD = \frac{1}{N} \sum_{i=0}^{N} |Facenet(F_{Gi}) - Facenet(F_G T_i)| \quad (19)$$

Here, $Facenet()$ indicates using a pre-trained Facenet model to extract a semantic vector of the input face. F_{Gi} is one of the generated faces, $F_G Ti$ is the ground truth of the synthesized face image. $cos()$ indicates calculating the cosine similarity of two vectors. For calculating FSS and FSD, a pretrained VGGFace2 model provided by facenet-pytorch[¶] was used.

C. Qualitative Results

Fig. 5 comprises images synthesized using the proposed system. The initial column denotes the input Bangla text descriptions, while the fourth column represents the corresponding translation of the stated Bangla descriptions. The images presented in the second column were produced through the utilization of the mean embedding approach of FGTD. On the other hand, the images displayed in the third column were generated by employing our proposed embedding strategy, which is elaborated in subsection IV-A. The figure presented provides clear evidence that the utilization of our proposed embedding strategy yields superior outcomes in generating accurate images that correspond to the input text. Specifically, the first, second, and fourth images of the third column accurately depict the gender as specified in the input text.

The visual representation in Fig. 6 illustrates that model-2 and model-3 have generated facial images that exhibit a degree of realism. The images generated by models 4, 5, and 6 exhibit noise and lack realism, which can be attributed to non-convergence, as noted in [27]. The potential reason for the absence of intricate features in almost all of the produced images may be attributed to the utilization of low-resolution images (solely 128x128) during the training process, coupled with the intricate structural composition of Bangla facial descriptions. Based on a visual assessment, it can be concluded that the proposed model, namely model-1, generated the most authentic and precise images. The model configuration details are presented in Table III, while their corresponding explanation can be found in subsection V-A.

[‡]https://pytorch.org/ignite/generated/ignite.metrics.InceptionScore.html
[§]https://github.com/S-aiueo32/lpips-pytorch
[¶]https://github.com/timesler/facenet-pytorch

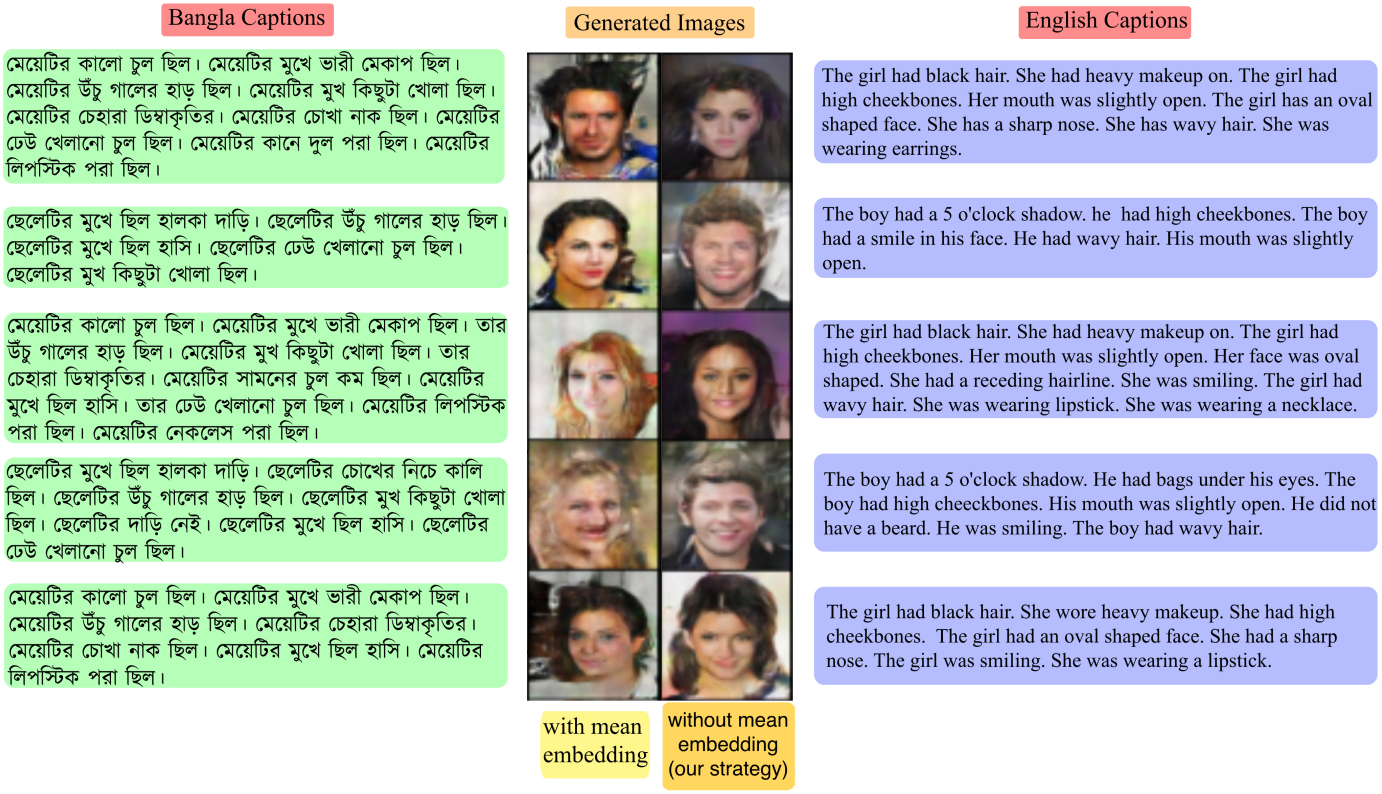

Fig. 5. Comparison of generated images between FGTD and proposed embedding strategies.

D. Quantitative Results

Table IV demonstrates that Model-1, which is proposed in this research, surpassed the other models in FID, Inception Score, and FSD metrics. This can be attributed to the stable training of DCGAN and the rich sentence embeddings of Bangla FastText. However, Model-7 shows moderate improvement in LPIPS and FSS performance metrics due to its matching-aware gradient penalty policy [16]. Nonetheless, the proposed model has a better overall performance.

TABLE IV. PERFORMANCE ANALYSIS AMONG DIFFERENT MODELS AND OUR PROPOSED MODEL

Experimental Models	FID ↓	IS ↑	LPIPS ↓	FSD ↓	FSS ↑
Model-1 (Proposed system)	**126.71**	**12.361**	21.8291	**20.23**	0.343
Model-2	165.87	11.676	21.6	20.35	0.34
Model-3	145.36	7.82	26.6	22.3	0.25
Model-4	184.17	9.05	6.25	21.81	0.303
Model-5	191.26	8.76	7.11	20.28	0.272
Model-6	210.93	8.28	6.6	21.93	0.233
Model-7	155.16	4.78	**3.22**	20.37	**0.42**

Bangla FastText performed better compared to sbnltk sentence transformer likely due to its robust pretraining procedure. Furthermore, it can be observed from Fig. 8 that the utilization of the proposed embedding strategy results in a superior FID score. The performance of our models is less significant in comparison to the state-of-the-art English text-to-face models, which can be caused by the intricate nature of Bangla textual descriptions. The FID score graph presented in Fig. 7 indicates that the proposed DCGAN+Bangla FastText method exhibits superior performance.

VI. DISCUSSION

It is clear from Fig. 9 that Both DCGAN and SAGAN suffered from non-convergence [27]. After about 47 epochs, DFGAN fell into mode collapse. Moreover, None of our models reached Nash equilibrium [27]. The last blocks of both generator and discriminator were omitted in the implementation of DFGAN in FGTD [1]; which may have made DFGAN more prone to mode collapse [27] and unstable training [27]. Although Transformer based models generally perform better than FastText-based models, Bangla FastText [2] performs better than sbnltk sentence transformer due to the superior training dataset, hyperparameter tuning, and preprocessing strategy used in Bangla FastText.

VII. LIMITATIONS OF OUR WORK

The quality of the generated images is low due to the proposed system's inability to map the text space to the generated facial image space accurately. Larger GAN architectures are relatively more capable of generating more realistic images. Further research can be done by employing such architectures to enhance the results of Bangla text-to-face synthesis.

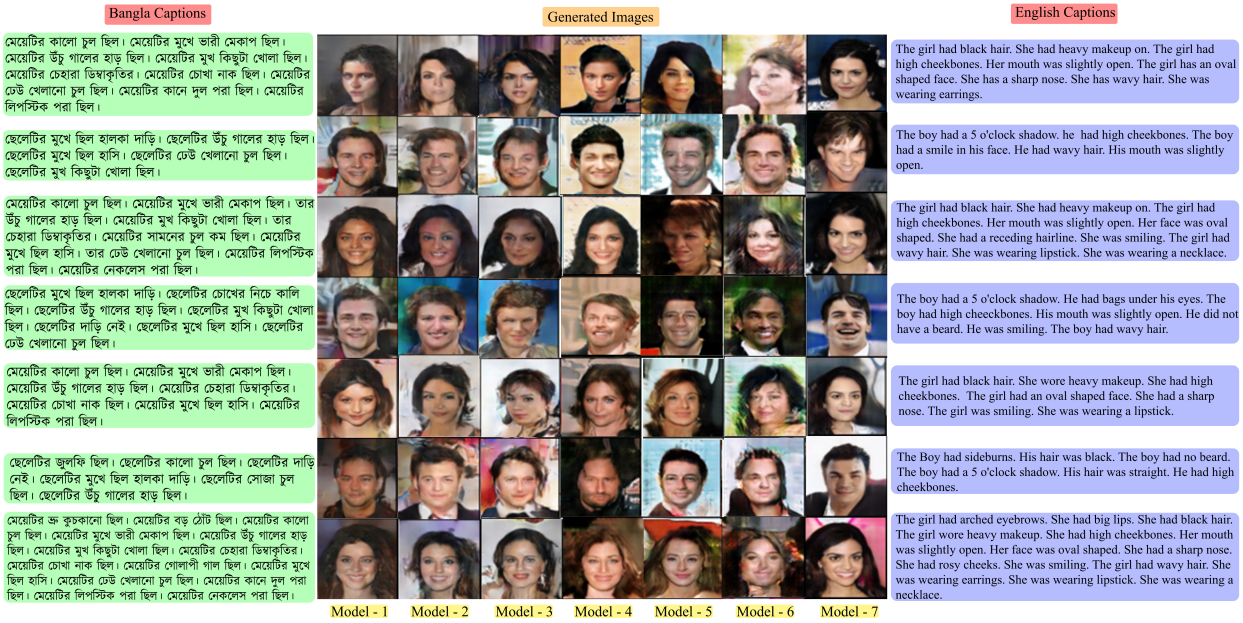

Fig. 6. Comparison of generated images among various experimental models.

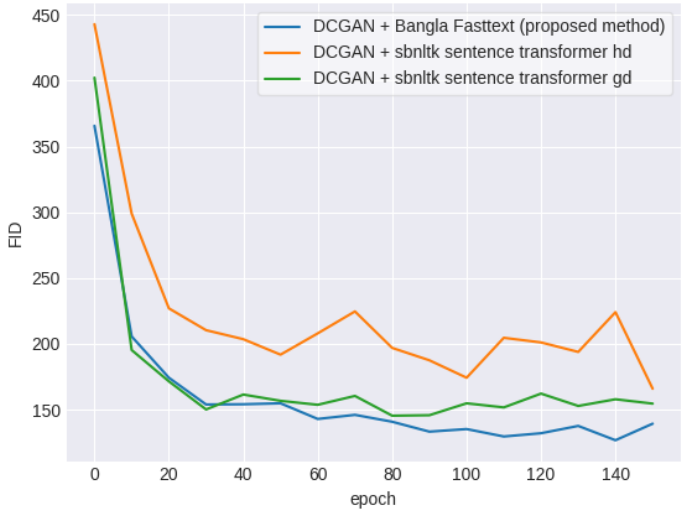

Fig. 7. Comparison of FID between our proposed model and other models considering FID.

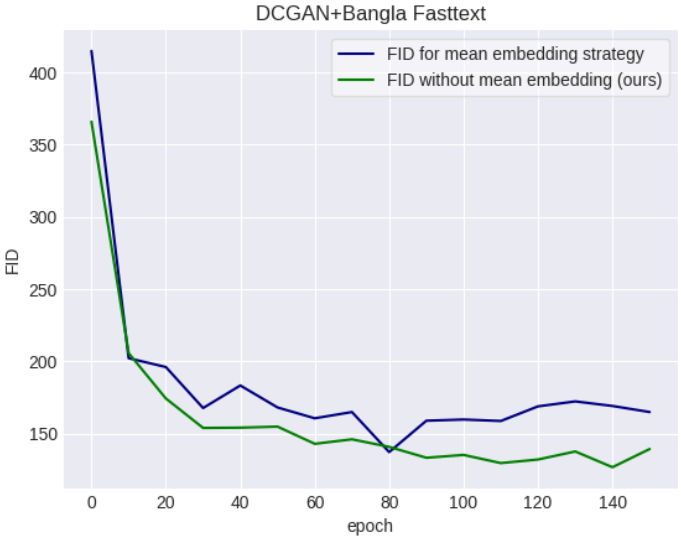

Fig. 8. Comparison of FID scores between FGTD [1] and our proposed embedding strategies.

VIII. Conclusion

This study presents a new approach that employs DCGAN + Bangla FastText to produce facial images based on textual descriptions in the Bangla language. A comprehensive performance comparison is provided between our proposed model and various utilizations of DCGAN, SAGAN, and DFGAN models in conjunction with Bangla FastText, sbnltk sentence transformer hd and sbnltk sentence transformer gd pre-trained Bangla text encoders. Furthermore, a new textual embedding approach is suggested. The superiority of the suggested embedding approach is established through both qualitative and quantitative results. The models presented in Table III were trained and evaluated using the CelebA Bangla dataset that is proposed in this research. The evaluation of generated face images involves the utilization of five distinct performance metrics, specifically FID, IS, LPIPS, FSS, and FSD. The study revealed that among all the models tested, the proposed model (DCGAN + Bangla FastText) exhibited the highest performance, attaining an FID, IS and FSD score of 126.71, 12.361 and 20.23 respectively. The proposed system's performance is moderate, likely due to the intricate structure of textual descriptions in the Bangla language. The use of diffusion models, variational autoencoders, pre-trained GANs, generating higher resolution images (256x256, 512x512, or

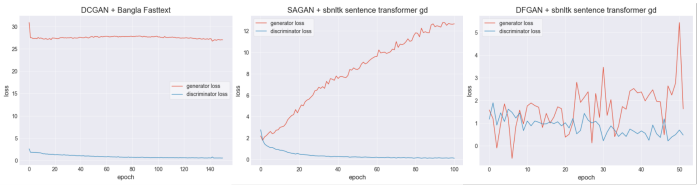

Fig. 9. Losses at different epochs of our experimental models.

1024x1024) in conjunction with large pre-trained language models can be investigated in future research for the task of Bangla text-to-face synthesis.

ACKNOWLEDGMENT

We are grateful to the Institute of Energy, Environment, Research, and Development (IEERD, UAP) and the University of Asia Pacific for their financial support.

REFERENCES

[1] K. Deorukhkar, K. Kadamala, and E. Menezes, "Fgtd: Face generation from textual description," in *Inventive Communication and Computational Technologies: Proceedings of ICICCT 2021*. Springer, 2022, pp. 547–562.

[2] M. Kowsher, M. S. I. Sobuj, M. F. Shahriar, N. J. Prottasha, M. S. Arefin, P. K. Dhar, and T. Koshiba, "An enhanced neural word embedding model for transfer learning," *Applied Sciences*, vol. 12, no. 6, p. 2848, 2022.

[3] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 3730–3738.

[4] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[5] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[6] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," in *International conference on machine learning*. PMLR, 2019, pp. 7354–7363.

[7] M. A. H. Palash, M. A. Al Nasim, A. Dhali, and F. Afrin, "Fine-grained image generation from bangla text description using attentional generative adversarial network," in *2021 IEEE International Conference on Robotics, Automation, Artificial-Intelligence and Internet-of-Things (RAAICON)*. IEEE, 2021, pp. 79–84.

[8] S. Naveen, M. S. R. Kiran, M. Indupriya, T. Manikanta, and P. Sudeep, "Transformer models for enhancing attngan based text to image generation," *Image and Vision Computing*, vol. 115, p. 104284, 2021.

[9] W. Xia, Y. Yang, J.-H. Xue, and B. Wu, "Tedigan: Text-guided diverse face image generation and manipulation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 2256–2265.

[10] Y. Ma, H. Yang, B. Liu, J. Fu, and J. Liu, "Ai illustrator: Translating raw descriptions into images by prompt-based cross-modal generation," in *Proceedings of the 30th ACM International Conference on Multimedia*, 2022, pp. 4282–4290.

[11] X. Hou, X. Zhang, Y. Li, and L. Shen, "Textface: Text-to-style mapping based face generation and manipulation," *IEEE Transactions on Multimedia*, 2022.

[12] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," in *International conference on machine learning*. PMLR, 2016, pp. 1060–1069.

[13] L. Gao, D. Chen, Z. Zhao, J. Shao, and H. T. Shen, "Lightweight dynamic conditional gan with pyramid attention for text-to-image synthesis," *Pattern Recognition*, vol. 110, p. 107384, 2021.

[14] H. Zhang, J. Y. Koh, J. Baldridge, H. Lee, and Y. Yang, "Cross-modal contrastive learning for text-to-image generation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 833–842.

[15] M. Siddharth and R. Aarthi, "Text to image gans with roberta and fine-grained attention networks," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 12, 2021.

[16] M. Tao, H. Tang, F. Wu, X.-Y. Jing, B.-K. Bao, and C. Xu, "Df-gan: A simple and effective baseline for text-to-image synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16 515–16 525.

[17] M. Berrahal and M. Azizi, "Optimal text-to-image synthesis model for generating portrait images using generative adversarial network techniques," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 25, no. 2, pp. 972–979, 2022.

[18] D. Ayanthi and S. Munasinghe, "Text-to-face generation with stylegan2," *arXiv preprint arXiv:2205.12512*, 2022.

[19] J. Sun, Q. Deng, Q. Li, M. Sun, M. Ren, and Z. Sun, "Anyface: Free-style text-to-face synthesis and manipulation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18 687–18 696.

[20] T. Wang, T. Zhang, and B. Lovell, "Faces a la carte: Text-to-face generation via attribute disentanglement," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2021, pp. 3380–3388.

[21] J. Peng, X. Du, Y. Zhou, J. He, Y. Shen, X. Sun, and R. Ji, "Learning dynamic prior knowledge for text-to-face pixel synthesis," in *Proceedings of the 30th ACM International Conference on Multimedia*, 2022, pp. 5132–5141.

[22] J. Peng, H. Pan, Y. Zhou, J. He, X. Sun, Y. Wang, Y. Wu, and R. Ji, "Towards open-ended text-to-face generation, combination and manipulation," in *Proceedings of the 30th ACM International Conference on Multimedia*, 2022, pp. 5045–5054.

[23] J. Sun, Q. Li, W. Wang, J. Zhao, and Z. Sun, "Multi-caption text-to-face synthesis: Dataset and algorithm," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 2290–2298.

[24] A. Gatt, M. Tanti, A. Muscat, P. Paggio, R. A. Farrugia, C. Borg, K. P. Camilleri, M. Rosner, and L. Van der Plas, "Face2text: Collecting an annotated image description corpus for the generation of rich face descriptions," *arXiv preprint arXiv:1803.03827*, 2018.

[25] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, "Unsupervised cross-lingual representation learning at scale," *arXiv preprint arXiv:1911.02116*, 2019.

[26] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," *Advances in neural information processing systems*, vol. 30, 2017.

[27] A. Jabbar, X. Li, and B. Omar, "A survey on generative adversarial networks: Variants, applications, and training," *ACM Computing Surveys (CSUR)*, vol. 54, no. 8, pp. 1–49, 2021.