IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 16 February 2024, accepted 23 February 2024, date of publication 26 February 2024, date of current version 14 March 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3370674

RESEARCH ARTICLE

Okkhor-Diffusion: Class Guided Generation of Bangla Isolated Handwritten Characters Using Denoising Diffusion Probabilistic Model (DDPM)

MD. MUBTASIM FUAD[1], A. FAIYAZ[1], NOOR MAIRUKH KHAN ARNOB[1], M. F. MRIDHA[2], (Senior Member, IEEE), ALOKE KUMAR SAHA[1], AND ZEYAR AUNG[3,4], (Senior Member, IEEE)

[1]Department of Computer Science and Engineering, University of Asia Pacific, Dhaka 1205, Bangladesh
[2]Department of Computer Science and Engineering, American International University-Bangladesh, Dhaka 1216, Bangladesh
[3]Center for Secure Cyber-Physical Systems (C2PS), Khalifa University, Abu Dhabi, United Arab Emirates
[4]Department of Computer Science, Khalifa University, Abu Dhabi, United Arab Emirates

Corresponding author: M. F. Mridha (firoz.mridha@aiub.edu)

ABSTRACT Bangla has a unique script with a complex set of characters, making it a fascinating subject of study for linguists and cultural enthusiasts. Unique in some of its similar characters which are only distinguishable by subtle differences in their shapes and diacritics, there has been a notable increase in research on Bangla character recognition and classification using machine learning-based approaches. However, Handwritten Bangla Character Recognition (HBCR) training requires an adequate amount of data from a diversely distributed dataset. Making diverse datasets for HBCR training is a challenging and tedious task to carry out. Yet, there is limited research on the automatic generation of handwritten Bangla characters. Motivated by this open area of research, this paper proposes a novel approach 'Okkhor-Diffusion' for class-guided generation of Bangla isolated handwritten characters using a novel Denoising Diffusion Probabilistic Model (DDPM). No prior research has used DDPM for this purpose, making the proposed approach novel. The DDPM is a generative model that uses a diffusion process to transform noise-corrupted data into diverse samples; despite being trained on a small training set. In our experiments, StyleGAN2-ADA had notably inferior performance compared to Okkhor-Diffusion in generating realistic isolated handwritten Bangla characters. Experimental results on the BanglaLekha-Isolated dataset demonstrate that the proposed Okkhor-Diffusion model generates realistic isolated handwritten Bangla characters, with a mean Multi-Scale Structural Similarity Index Measure (MS-SSIM) score of 0.178 compared to 0.177 for the real samples. The Fréchet Inception Distance (FID) score for the synthetic handwritten Bangla characters is 5.426. Finally, the newly proposed Bangla Character Aware Fréchet Inception Distance (BCAFID) score of the proposed Okkhor-Diffusion model is 10.388. The code for the proposed Okkhor-Diffusion framework is available at https://github.com/MubtasimFuad10/Okkhor-Diffusion.

INDEX TERMS Deep learning, handwritten character generation, generative model, denoising diffusion probabilistic model.

I. INTRODUCTION

Recent developments in deep learning have enabled generative models to accurately imitate real images. Moreover, generative models have shown tremendous potential for synthesizing realistic and diverse images; garnering attention for their potential applications in synthetic data, visual content, and more.

The Bangla writing system is the fifth most widely used in the world [1]. Over 210 million people speak it as a first

The associate editor coordinating the review of this manuscript and approving it for publication was Maria Chiara Caschera.

TABLE 1. Diacritics, dots and shapes.

Shape	অ	আ
	ও	ঔ
	এ	ঐ
	য	ষ
Dot	ব	র
	য	য়
	ঢ	ঢ়
	ড	ড়

or second language, and over 100 million Bengali speakers in Bangladesh and approximately 85 million in India use it as their primary language [2]. The Bangla script consists of 50 basic characters, 10 numerals, and 334 compound characters created by combining these basic characters [2].

There are several similar-looking Bangla characters that can only be distinguished by dots, shapes, and strokes. As an illustration, the characters ''ব'' and ''র'', ''য'' and ''য়'', ''ঢ'' and ''ঢ়'', ''ড'' and ''ড়'' are only distinguishable from each other by the presence of a dot below the primary strokes shown in Table 1. The letter changes depending on whether or not these dots are present. Furthermore, the characters ''অ'' and ''আ'', ''য'' and ''ষ'', have subtle differences between them regarding shape shown in Table 1. These subtle differences between identical characters and the inherent variation between individual handwriting make it difficult to discern handwritten Bangla characters. The Denoising Diffusion Probabilistic Model (DDPM) has the potential to be applied in several real-life scenarios, such as font generation, typography animation, text resizing, and generating stylistic variations for artistic expressions. This can be accomplished by generating diverse and realistic handwritten characters.

Deep learning models have revolutionized handwritten character recognition research [3], [4], [5], yet Bangla handwritten character datasets are limited in size and availability [9]. This work uses a novel approach to generate synthetic handwritten Bangla characters to solve data scarcity in handwritten Bangla character recognition. Researchers have recently acknowledged synthetic data as a valid and effective solution to data availability and model generalization issues.

Generative modeling of data using machine learning originated in 1980 to generate similar data to improve classifier efficiency. Generative modeling is used for image synthesizing, super-resolution, in-painting, image-to-image conversion, and text-to-image. There are various types of generative models, including Variational Auto Encoders, Generative Adversarial Networks (GAN), and Score-Based Generative models, among others. One popular technique to generate synthetic data is through GANs. In recent years, a denoising diffusion model emerged as a dominant model in the generative modeling space and outperformed GANs in terms of image relevancy and quality.

''Okkhor'' is a Bangla word which means symbolic letters or characters. As we are working with Bangla characters, every individual character is termed as Okkhor. Additionally, we are using a model named diffusion model for generating isolated handwritten Bangla characters. Therefore, by combining the two key concepts of our proposed system, we named our framework ''Okkhor-Diffusion''.

The main contribution of our paper is four fold, summarized as follows:

- This paper introduces Okkhor-Diffusion, a novel framework for generating isolated Bangla handwritten characters.
- Our paper presents an enhanced evaluation metric referred to as Bangla Character Aware Fréchet Inception Distance (BCAFID). This metric is more appropriate for assessing the quality and diversity of synthetically generated Bangla chracters compared to FID (Fréchet Inception Distance).
- A novel Bangla character interpolation algorithm is proposed. This algorithm can smoothly transition between any two Bangla characters.
- A detailed quantitative and qualitative performance analysis of synthetic Bangla character generation is provided. Images generated by DDPM and StyleGAN2-ADA along with Fréchet Inception Distance, Multi-Scale Structural Similarity Index Measure, Learned Perceptual Image Patch Similarity and BCAFID metrics are shown and compared for all experimental models.

The rest of the paper is structured as follows: Section II highlights previous studies that are relevant to the topic of our paper. Section III describes the approach and process used to develop the Okkhor-Diffusion model. Section IV focuses on the datasets used in this research and it provides details about the three specific datasets that were utilized to train the Okkhor-Diffusion model. After that, Section V discusses the hardware and software configurations used during the experiments. Section VI explores Bangla character interpolation, i.e., the generation of new samples lying between existing samples in the latent space. In Section VII, our paper presents the results obtained from the experiments and provides an analysis of those results. Section VIII states the limitation and the future work. Finally, IX summarises the main findings of the research and provides a concise summary of the paper.

II. RELATED WORKS

This work can be considered to be novel because, to the best of our knowledge, no prior research has been conducted on class-based handwritten Bangla character generation using DDPM. However, it should be noted that there has been some research on handwritten Bangla character generation using GANs. The following section provides an overview of the previous studies related to this work, Okkhor-Diffusion.

A. BANGLA HANDWRITTEN CHARACTER GENERATION USING GAN

Bangla Handwritten character generation was mostly done by using GAN (Generative Adversarial Networks). One of the most prominent of these works was done by the paper [18]

where the authors generated Bangla handwritten characters of the desired class using a variant of Conditional Generative Adversarial Network. This variant was constructed by combining the original implementation of Conditional Generative Adversarial Network (cGAN) with Deep Convolutional Generative Adversarial Network (DCGAN) and claims to achieve ''visually appealing'' results. One of the limitations of this work was that the authors did not provide scores of any performance metrics to actually evaluate the quality and diversity of the images; although the generated images visually looked similar to the training data.

Another study aims to produce image-to-image characters, the paper [19] developed Variational AutoEncoder Generative Adversarial Network (VAEGAN) to generate Bangla printed characters from Bangla handwritten character images. They used CMATERdb as their dataset and created a printed handwritten character based on Sutonnymj font which has 231 classes. The authors used handwritten character images to guide the process of handwritten printed character generation. The authors were able to generate 28 × 28 resolution Bangla printed character images. They ran their model for 100 epochs and the experimental results produced were quite diverse and accurately represented the classes but they did not mention any image quality metrics such as FID.

B. DENOISING DIFFUSION PROBABILISTIC MODEL (DDPM)

The Denoising Diffusion Probabilistic Model (DDPM) has gained significant popularity in the field of computer vision for its exceptional performance in generating high-quality images [10]. DDPM reconstructs the data distribution by employing two parameterized Markov chains and variational inference. DDPMs have shown impressive capabilities to generate high-quality and high-diversity images, surpassing other popular generative models such as GANs [10], [11]. Moreover, recent research indicates that Diffusion Models outperform GANs on image synthesis tasks, demonstrating the superior capabilities of diffusion models in this domain [12]. Diffusion models have been utilized in numerous fields [13], [14], including high-resolution image generation [15], image inpainting [16], and natural language processing [17], among others. A paper [34] uses DDPM for Chinese character generation and interpolation.

C. LIMITATION OF PREVIOUS WORK

In previous work, the researchers did not provide any quantitative metrics to evaluate the generated images of Bangla Handwritten characters, relying solely on visual comparison with real images. In addition, the generated images exhibit low quality and limited diversity. Furthermore, their proposed model was unable to generate high-quality images despite being trained for 1500 epochs [18] and 100 epochs [19]. Although FID [26] is generally used for evaluating images produced by generative models, FID does not cover handwritten characters. There is a need for a novel, standardized, robust performance metric for evaluating synthetic Bangla character images objectively.

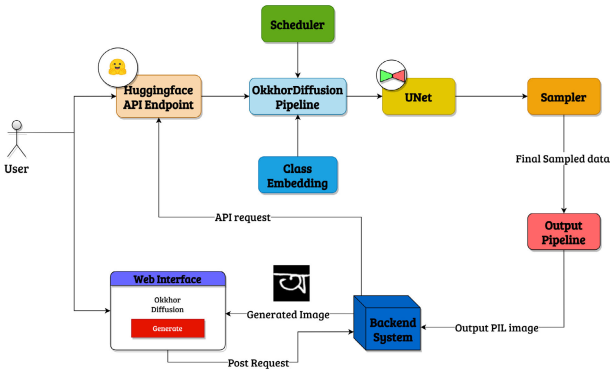

FIGURE 1. Proposed framework of Okkhor-Diffusion system.

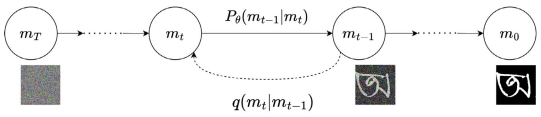

FIGURE 2. The Markov chain representing forward and reverse diffusion process illustrating noise added and then removed for generating Bangla handwritten character samples. Adapted from [10].

III. METHODOLOGY

Figure 1 describes our proposed system called ''Okkhor Diffusion'' that utilizes a DDPM model. Users interact with a web interface to provide input and receive output. The user's input is sent via a POST request to the backend system. The backend system communicates with the HuggingFace API through API request to access the Okkhor Diffusion pipeline. This pipeline incorporates a scheduler, time embedding, and class embedding, and passes the data to the UNet architecture for processing. The sampling process is then initiated, and the final sampled data is sent to the Output Pipeline. The scheduler, UNet and sampler follow a reverse diffusion process explained in Section III, Subsection III-B for generating images. The Output Pipeline converts the data into a PIL image, which is sent back to the backend system. Finally, the generated image is displayed in the web interface for users to view.

The diffusion model represents a new methodology for data generation, hence we used Diffusion Model to generate Bangla handwritten characters. Diffusion Models are generative models that introduce a cutting-edge method of generating synthetic data. The diffusion model works by adding noise to the data step by step which is defined by a Markov chain and then learns to reverse the diffusion process to construct the desired samples from the noise [22]. The basic principles of the DDPM used in our proposed Okkhor-Diffusion method, Forward Diffusion and Reverse Diffusion are explained in Subsections III-A and III-B under Section III.

A. FORWARD DIFFUSION PROCESS

In the isolated handwritten Bangla character generation process using Okkhor-Diffusion, the process begins with a procedure known as forward diffusion shown in Figure 2 that destroys the data by adding Gaussian noise successively. Hence, a character image from the real distribution $\mathbf{m}_0 \sim q(\mathbf{m})$ is selected and in the forward diffusion process, small amounts of gaussian noise are added to the sample in T steps to produce a sequence of noisy samples $\mathbf{m}_1, \ldots, \mathbf{m}_t$ according to the Equation (1). The step sizes are controlled by a variance schedule $\{\omega_t \in (0,1)\}_{t=1}^{T}$ [10]

$$q(\mathbf{m}_t|\mathbf{m}_{t-1}) = \mathcal{N}(\mathbf{m}_t; \sqrt{1-\omega_t}\mathbf{m}_{t-1}, \omega_t\mathbf{I}) \tag{1}$$

The data sample $\mathbf{m}_0$ gradually loses its features as the step t becomes bigger when $t \to \infty$, $\mathbf{m}_0$ is equivalent to an isotropic Gaussian distribution. A useful property of the above process is that it can sample $\mathbf{m}_t$ at any arbitrary time step t using the reparametrization technique [22]. Let $\phi_t = 1 - \omega_t$ and $\bar{\phi}_t = \prod_{i=1}^{t} \phi_i$:

$$\begin{aligned}\mathbf{m}_t &= \sqrt{\phi_t}\mathbf{m}_{t-1} + \sqrt{1-\phi_t}\boldsymbol{\psi}_{t-1} \\ &= \sqrt{\phi_t\phi_{t-1}}m_{t-2} + \sqrt{1-\phi_t\phi_{t-1}}\bar{\boldsymbol{\psi}}_{t-2} \\ &= \ldots \\ &= \sqrt{\bar{\phi}_t}\mathbf{m}_0 + \sqrt{1-\bar{\phi}_t}\boldsymbol{\psi}\end{aligned}$$

Note:
$\boldsymbol{\psi}_{t-1}, \boldsymbol{\psi}_{t-2}, \ldots\ldots \sim \mathcal{N}(0, I)$
$\bar{\boldsymbol{\psi}}_{t-2}$ merges two Gaussians.

$$q(\mathbf{m}_t|\mathbf{m}_0) = \mathcal{N}(\mathbf{m}_t; \sqrt{\bar{\phi}_t}\mathbf{m}_0, (1-\bar{\phi}_t)\mathbf{I})$$

which means it can sample noisy versions of the Bangla handwritten character image of sample $\mathbf{m}_0$ at any time step just by using the initial sample $\mathbf{m}_0$ and pre-computed $\bar{\phi}_t$ at t time step.

for $t = 1, \ldots, T$. The variance schedule can be defined as a small linear schedule to increase linearly from $\omega_1 = 10^{-4}$ to $\omega_T = 0.02$ [10]. Alternatively, the authors of [12] found a cosine schedule that performs well. Cosine schedule was used in the experiments which are defined in terms of $\bar{\phi}_t$:

$$\bar{\phi}_t = \frac{g(t)}{g(0)}, \quad g(t) = \cos\left(\frac{t/T + s}{1+s} \cdot \frac{\pi}{2}\right)^2$$

B. REVERSE DIFFUSION PROCESS

In the reverse diffusion process of the Okkhor-Diffusion framework, to recreate the true sample from a Gaussian noise $\mathbf{m}_t \sim N(0, I)$ the reversal of the above process and sample from $q(\mathbf{m}_{t-1}|\mathbf{m}_t)$ is needed. If ω_t is small enough, $q(\mathbf{m}_{t-1}|\mathbf{m}_t)$ will also be a Gaussian. So it can be approximated with a parameterized model p_θ as shown in Equation (2) which is adapted from [12].

$$\begin{aligned}p_\theta(\mathbf{m}_{0:T}) &= p(\mathbf{m}_T)\prod_{t=1}^{T} p_\theta(\mathbf{m}_{t-1}|\mathbf{m}_t) \\ p_\theta(\mathbf{m}_{t-1}|\mathbf{m}_t) &= \mathcal{N}(\mathbf{m}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{m}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{m}_t, t))\end{aligned} \tag{2}$$

Since the reverse conditional probability has a closed form expression when conditioned on m_0

$$q(\mathbf{m}_{t-1}|\mathbf{m}_t, \mathbf{m}_0) = \mathcal{N}(\mathbf{m}_{t-1}; \tilde{\boldsymbol{\mu}}(\mathbf{m}_t, \mathbf{m}_0), \tilde{\omega}_t\mathbf{I}) \tag{3}$$

where,

$$\begin{aligned}\tilde{\omega}_t &= 1/(\frac{1}{1-\bar{\phi}_{t-1}} + \frac{\phi_t}{\omega_t}) \\ &= 1/(\frac{\phi_t - \bar{\phi}_t + \omega_t}{\omega_t(1-\bar{\phi}_{t-1})}) \\ &= \frac{1-\bar{\phi}_{t-1}}{1-\bar{\phi}_t} \cdot \omega_t \\ \tilde{\boldsymbol{\mu}}_t(\mathbf{m}_t, \mathbf{m}_0) &= (\frac{\sqrt{\phi_t}}{\omega_t}\mathbf{m}_t + \frac{\sqrt{\bar{\phi}_{t-1}}}{1-\bar{\phi}_{t-1}}\mathbf{m}_0)/(\frac{\phi_t}{\omega_t} + \frac{1}{1-\bar{\phi}_{t-1}}) \\ &= (\frac{\sqrt{\phi_t}}{\omega_t}\mathbf{m}_t + \frac{\sqrt{\bar{\phi}_{t-1}}}{1-\bar{\phi}_{t-1}}\mathbf{m}_0)\frac{1-\bar{\phi}_{t-1}}{1-\bar{\phi}_t} \cdot \omega_t \\ &= \frac{\sqrt{\phi_t}(1-\bar{\phi}_{t-1})}{1-\bar{\phi}_t}\mathbf{m}_t + \frac{\sqrt{\bar{\phi}_{t-1}}\omega_t}{1-\bar{\phi}_t}\mathbf{m}_0\end{aligned} \tag{4}$$

Here $\tilde{\omega}_t$ is the variance and $\tilde{\boldsymbol{\mu}}_t$ is the mean and again using reparameterization technique, $\mathbf{m}_0$ can be represented by $\mathbf{m}_0 = \frac{1}{\sqrt{\bar{\phi}_t}}(\mathbf{m}_t - \sqrt{1-\bar{\phi}_t}\boldsymbol{\psi}_t)$ and by plugging it into Equation (4) [10], the mean becomes

$$\begin{aligned}\tilde{\boldsymbol{\mu}}_t &= \frac{\sqrt{\phi_t}(1-\bar{\phi}_{t-1})}{1-\bar{\phi}_t}\mathbf{m}_t + \frac{\sqrt{\bar{\phi}_{t-1}}\omega_t}{1-\bar{\phi}_t}\frac{1}{\sqrt{\bar{\phi}_t}}(\mathbf{m}_t \\ &\quad - \sqrt{1-\bar{\phi}_t}\boldsymbol{\psi}_t) \\ &= \frac{1}{\sqrt{\phi_t}}\left(\mathbf{m}_t - \frac{1-\phi_t}{\sqrt{1-\bar{\phi}_t}}\boldsymbol{\psi}_t\right)\end{aligned}$$

So a neural network $\psi_{\theta(m_t)}$ can be trained to approximate ψ_t. Eventually the mean μ parameterized by θ becomes

$$\boldsymbol{\mu}_\theta(\mathbf{m}_t, t) = \frac{1}{\sqrt{\phi_t}}\left(\frac{\phi_t - 1}{\sqrt{1-\bar{\phi}_t}}\boldsymbol{\psi}_\theta(\mathbf{m}_t, t) + \mathbf{m}_t\right)$$

The loss function in Equation (5) as proposed on [10] is used for training the U-Net model is

$$\begin{aligned}L_t &= \mathbb{E}_{\mathbf{m}_0,\boldsymbol{\psi}}\left[\frac{1}{2\|\boldsymbol{\Sigma}_\theta(\mathbf{m}_t, t)\|_2^2}\|\tilde{\boldsymbol{\mu}}_t(\mathbf{m}_t, \mathbf{m}_0) - \boldsymbol{\mu}_\theta(\mathbf{m}_t, t)\|^2\right] \\ &= \mathbb{E}_{\mathbf{m}_0,\boldsymbol{\psi}}\left[\frac{(1-\phi_t)^2}{2\phi_t(1-\bar{\phi}_t)\|\boldsymbol{\Sigma}_\theta\|_2^2}\|\boldsymbol{\psi}_t - \boldsymbol{\psi}_\theta(\mathbf{m}_t, t)\|^2\right] \\ &= \mathbb{E}_{\mathbf{m}_0,\boldsymbol{\psi}}\left[\frac{(1-\phi_t)^2}{2\phi_t(1-\bar{\phi}_t)\|\boldsymbol{\Sigma}_\theta\|_2^2}\|\boldsymbol{\psi}_t - \boldsymbol{\psi}_\theta(\sqrt{\bar{\phi}_t}\mathbf{m}_0 \right. \\ &\quad \left. + \sqrt{1-\bar{\phi}_t}\boldsymbol{\psi}_t, t)\|^2\right]\end{aligned} \tag{5}$$

Ho et al. [10] also discovered that training the model works better by ignoring the weighting term and proposed a simplified version of the loss function:

$$L_t^{\text{simple}} = \mathbb{E}_{t\sim[1,T],\mathbf{m}_0,\boldsymbol{\psi}_t}\left[\|\boldsymbol{\psi}_t - \boldsymbol{\psi}_\theta(\mathbf{m}_t, t)\|^2\right] = \mathbb{E}_{t\sim[1,T],\mathbf{m}_0,\boldsymbol{\psi}_t}\left[\|\boldsymbol{\psi}_t - \boldsymbol{\psi}_\theta(\sqrt{\bar{\phi}_t}\mathbf{m}_0 + \sqrt{1-\bar{\phi}_t}\boldsymbol{\psi}_t, t)\|^2\right] \quad (6)$$

In this paper a U-Net model is trained to predict $\psi_\theta(m_t)$ with the proposed loss function in Equation (6) which is adapted from [10].

C. ARCHITECTURE

The proposed U-Net architecture in Figure 3 is based on the publicly available library known as Diffusers [29]. The input image is of 64 × 64 resolution. There are six resolution stages in the encoder and decoder respectively. The downsampling blocks in the encoder each consist of 2 ResNet blocks followed by an average pooling layer and the fifth downsampling block has self-attention mechanism. On the other hand, the upsampling blocks in the decoder stage contain 3 ResNet blocks and the 2nd upsampling block has self-attention mechanism. The bottleneck combines a ResNet block, and Attention block, followed by another ResNet block. We modified the block out sizes of the U-Net architecture to be 64,64,128,128,256,256 for both the downsampling in the encoder section and the upsampling blocks in the decoder section. The timesteps are encoded using sinusoidal positional embedding and for conditioning the output to our desired Bangla character class, we concatenated the class embedding with the timestep embedding and then added it to each Residual block.

In Figure 4, it illustrates the training process of the UNet neural network and getting it to predict noise. Hence the goal of the neural network is for it to predict noise, and really it learns the distribution of what is noise on the image, but also what is not noise. We take a Bangla character image (e.g., "অ") from our training data, and we actually add noise to it. We add noise to it, and we give it to the neural network, and we ask the neural network to predict that noise. And then we compare the predicted noise against the actual noise that was added to that image, and that's how we compute the loss. And that backprops through the neural network, so then the neural network learns to predict that noise better.

The sampling process starts with pure Gaussian noise along with time embedding and class embedding, sampled from a known prior distribution. The trained UNet neural network is used to iteratively remove noise from the noisy samples, moving towards the original data distribution. This is visualized in the image as the gradual denoising of a noisy image. The iterative process is going on until we get the clear and diverse image which is X_0 as demonstrated in Figure 4.

IV. DATASET

A variety of Bangla handwriting datasets containing Bangla digits, alphabets, and other characters have been utilized for the task of automated generation of handwritten Bangla characters over the past years. In Table 2, a list of regularly used Bangla handwriting datasets is given along with additional details.

The BanglaLekha-Isolated [6], Ekush dataset [7] and CMATERdb [8] databases, as shown in Table 2, are the only publicly accessible relevant databases for Bangla handwritten characters. While other lists of datasets exist, they are not deemed suitable for training the model. It is apparent that BanglaLekha-Isolated is the dataset with the highest level of standardization and widespread use in research when compared to the other datasets being considered.

The BanglaLekha-Isolated dataset comprises 1,66,105 images representing 84 individual characters. These characters consist of 50 basic Bangla characters, 10 numerals, and 24 compound characters. 2000 handwriting samples from each of the 84 character classes were collected, digitized, and preprocessed.

Ekush Dataset is another dataset available for Bangla handwritten characters, which contains isolated handwritten characters for Bangla modifiers, vowels, consonants, compound letters, and numerical digits. The dataset comprises 367,018 characters written by 3,086 unique writers from Bangladesh. However, for our purposes, we excluded the modifiers, resulting in a total of 334,636 images included in the analysis section.

Another relevant dataset is the CMATERdb dataset, consisting of three distinct datasets for numerals, basic characters, and compound characters. Specifically, CMATERdb 3.1.2 provides data for basic characters, while CMATERdb 3.1.3.3.7z is dedicated to compound characters, and CMATERdb 3.1.1 contains data for numerals. For our purposes, we only used the basic and compound character datasets because the CMATERdb numerals dataset has images with a resolution of 28 × 28, which is too low for our needs as we are working with 64 × 64 images.

V. EXPERIMENTAL SETUP

The experimental model, DDPM+AdamW, and the proposed model Okkhor-Diffusion, which applies DDPM+Lion, were trained and evaluated on a system equipped with a 13th Gen Intel(R) Core(TM) i7-13700K processor with a clock speed ranging from 3.40 GHz to 5.40 GHz, 64 gigabytes of DDR4 RAM. Additionally, Nvidia RTX 4090 GPU with 24 gigabytes of VRAM was utilized. The models were designed and implemented utilizing the Anaconda 23.3.1 environment, which operates on Windows 10 and contains Python 3.10.11.

The DDPM+AdamW model was trained using a batch size of 32 and a learning rate of 1e-4. Also, the Lion Optimizer [24] is applied on the proposed model Okkhor-Diffusion trained on BanglaLekha-Isolated dataset, using a batch size of 128, resulting in slightly better results compared to

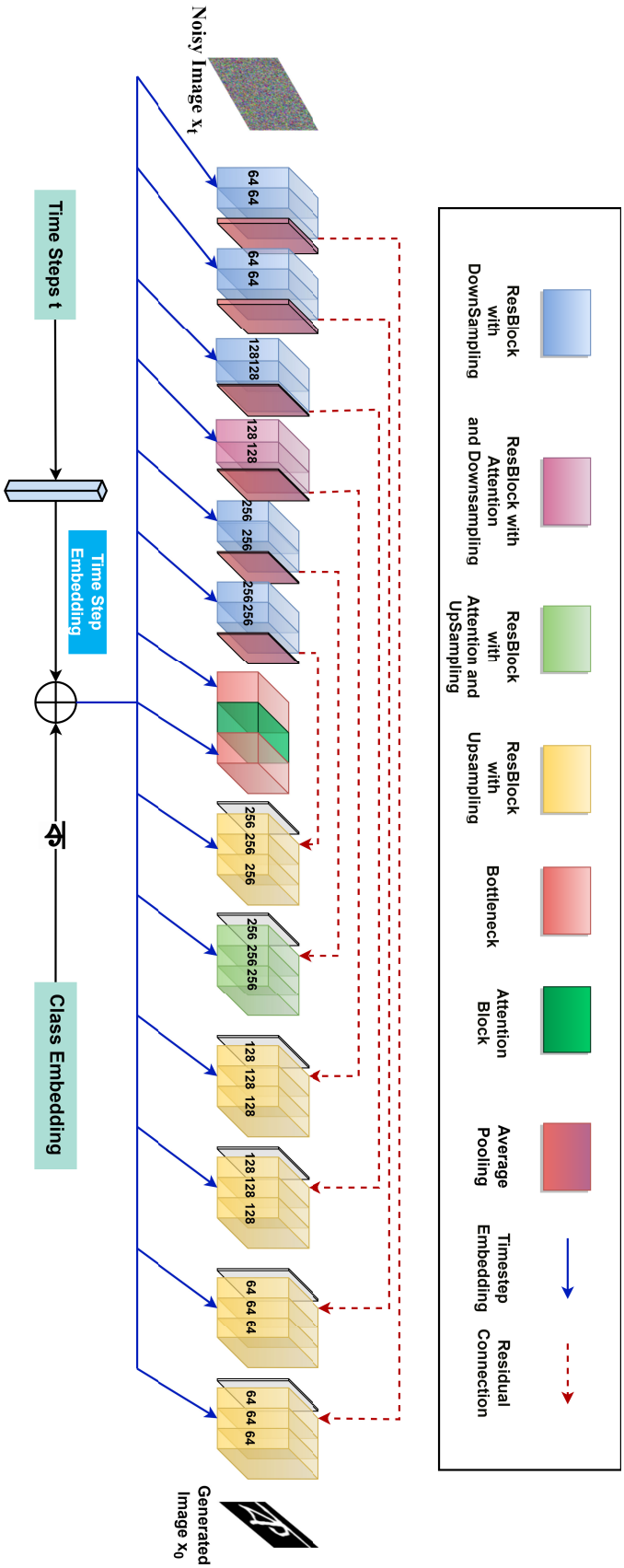

FIGURE 3. U-Net architecture of the proposed Okkhor-Diffusion model.

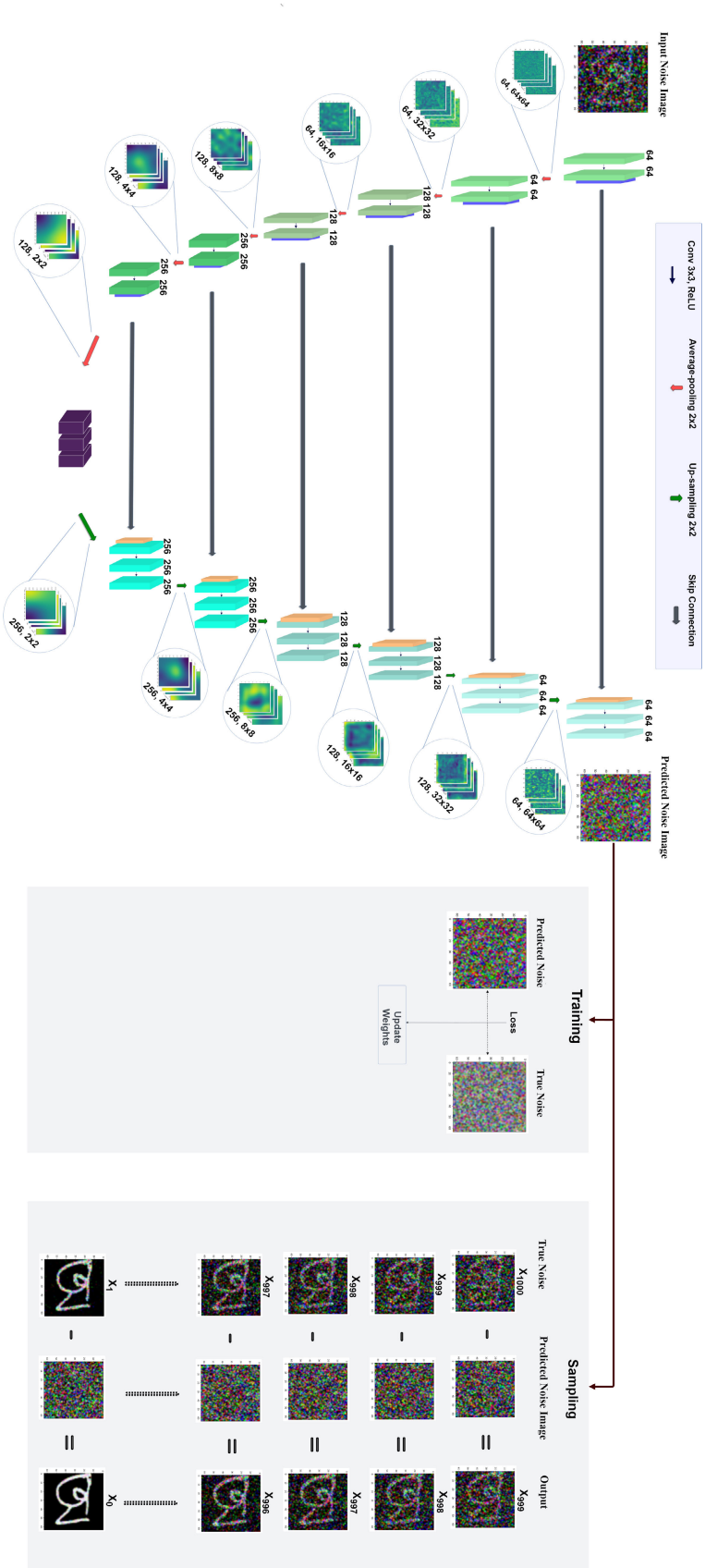

FIGURE 4. Inner workings of the training and sampling process of the proposed Okkhor-Diffusion model.

TABLE 2. Image statistics of different datasets.

Datasets	BanglaLekha-Isolated	Eksuh Dataset	CMATERdb
Basic Characters	98,950	152,584	15,001
Numerals	19,748	30,691	6,000
Compound Characters	47,407	151,361	34,439

those obtained with DDPM+AdamW(training configuration shown in Table 3. The generated images have a resolution of 64 × 64 pixels. The Okkhor-Diffusion model was trained for 50 epochs, resulting in the generation of high-quality and diverse images shown in Figure 6.

Generative models like GigaGAN [36] (1 Billion parameters),VQGAN [38] (1.4 Billion parameters), CogView [37] (4 Billion parameters) have high number of parameters; deeming such models unsuitable for comparing with our DDPM(28 Million trainable parameters). Therefore, for comparing with our DDPM, a state-of-the-art GAN based model, StyleGAN2-ADA [33] (25 Million trainable parameters) was trained on BanglaLekha-Isolated, Ekush and CMATERdb datasets.

VI. BANGLA CHARACTER INTERPOLATION

We can interpolate or gradually transition between Bangla character images x_1 to x_2 in latent space. For that We have combined denoised interpolation and character class embedding space interpolation. Two Bangla character images x_1 and x_2 are considered as the source images and their respective class embeddings are c_1 and c_2. We first sample x_1' and x_2' from Equation (7).

$$x_1' \sim q(x_t|x_0), x_2' \sim q(x_t|x_0) \tag{7}$$

Here q is forward diffusion and then we used this diffused source x_1' and x_2' to interpolate noise from Equation (8).

$$x_t = (1-\lambda) \times x_1' + \lambda \times x_2' \tag{8}$$

Here $\lambda = 0 \ldots 1$ is the interpolation factor. Equation (8) is the equation for denoised interpolation. Then we need to sample $x_T \sim p(x_0|x_t)$. Here $p(x_0|x_t)$ is the reverse diffusion process. While sampling $x_T \sim p(x_0|x_t)$ we also use the class embeddings c_1 and c_2 for interpolation by passing new class embedding vector defined in Equation (9).

$$c = (1-\lambda) \times c_1 + \lambda \times c_2 \tag{9}$$

Equation (9) is the equation for class embedding space interpolation. This new class embedding vector c along with x_T as pure noise is passed into the model(ϵ_θ) and is used in the denoising process. The Bangla Character Interpolation process is described in Algorithm 1.

Figure 5 shows that our algorithm smoothly transitions between the numerical characters "১" and "৪", vowels "এ" and "ও", compound characters "ধ্ব" and "ন্দ". Here x_1 and x_2 are images with classes c_1 and c_2 that we want to interpolate in between. Normally if we do pixel space interpolation we will not be able to generate new samples. But using the algorithm described in 1 we can generate new unseen characters,i.e: while interpolating from "১" to "৪", "৯" is generated when $\lambda = 0.5$.

Algorithm 1 Algorithm for Bangla Character Interpolation

$x_1' \sim q(x_t|x_0)$
$x_2' \sim q(x_t|x_0)$
$T \leftarrow 1000$
$images \leftarrow []$
$\delta\lambda \leftarrow \frac{1}{\text{frames}}$
for $\lambda = 0; \lambda \leq 1; \lambda = \lambda + \delta\lambda$ **do**
  $x_T = (1-\lambda) \times x_1' + \lambda \times x_2'$
  $c = (1-\lambda) \times c_1 + \lambda \times c_2$
  for $t = T; t \geq 1; t = t-1$ **do**
    $x_{t-1} = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{1-\alpha_t}{\sqrt{1-\alpha_t}} \times \epsilon_\theta(x_t, t, c))$
    $x_t = x_{t-1}$
  end for
  $images \leftarrow x_0$
end for

VII. RESULTS AND ANALYSIS

A. QUALITATIVE RESULTS

The qualitative results of the generated and real training samples for the BanglaLekha-Isolated, Ekush dataset and CMATERdb dataset are provided here. Figure 6, Figure 8, and Figure 10 depicts a selection of randomly chosen samples from the training (left) and generated (right) datasets, encompassing images from various classes. The generated samples were chosen randomly and demonstrated a level of visual distinguishability across all classes that is comparable to that of the training samples, without any indication of selection bias. The generated images for the dataset exhibit higher visual quality comparable to the training samples, indicating the success of the proposed DDPM in capturing the underlying data distribution and generating diverse samples. By visually comparing Figure 6 and Figure 7, Figure 8 and Figure 9, Figure 10 and Figure 11 we observe that StyleGAN2-ADA generates unrealistic samples compared to Okkhor-Diffusion accross all datasets in consideration.

B. QUANTITATIVE RESULTS

Although the qualitative results produced by Okkhor-Diffusion are visually plausible, a qualitative analysis is not enough to accurately assess the model. Therefore, the generated sample quality was compared using quantitative evaluation metrics.

TABLE 3. Experimental setup of the models trained.

Type	DDPM+AdamW	Okkhor-Diffusion
Images trained	556,181	556,181
Batch size	32	128
Activation function	silu	silu
Number of normalization group	32	32
ResBlock in downsampling	2	2
ResBlock in upsampling	3	3
Diffusion steps	1000	1000
Blockout Size	64, 64, 128, 128, 256, 256	64, 64, 128, 128, 256, 256
Noise Schedule	cosine scheduler	cosine scheduler
Learning rate	1e-4	1e-4
Optimizer	AdamW	Lion
Time embedding type	positional	positional
Hyperparameter	Initial learning rate α = 0.0001, $\beta_1 = 0.9, \beta_2 = 0.99$, weight decay = 0.01	Initial learning rate α =0.0001, $\beta_1 = 0.9, \beta_2 = 0.99$, weight decay = 0.0
Number of total trainable parameters	28,468,739	28,468,739

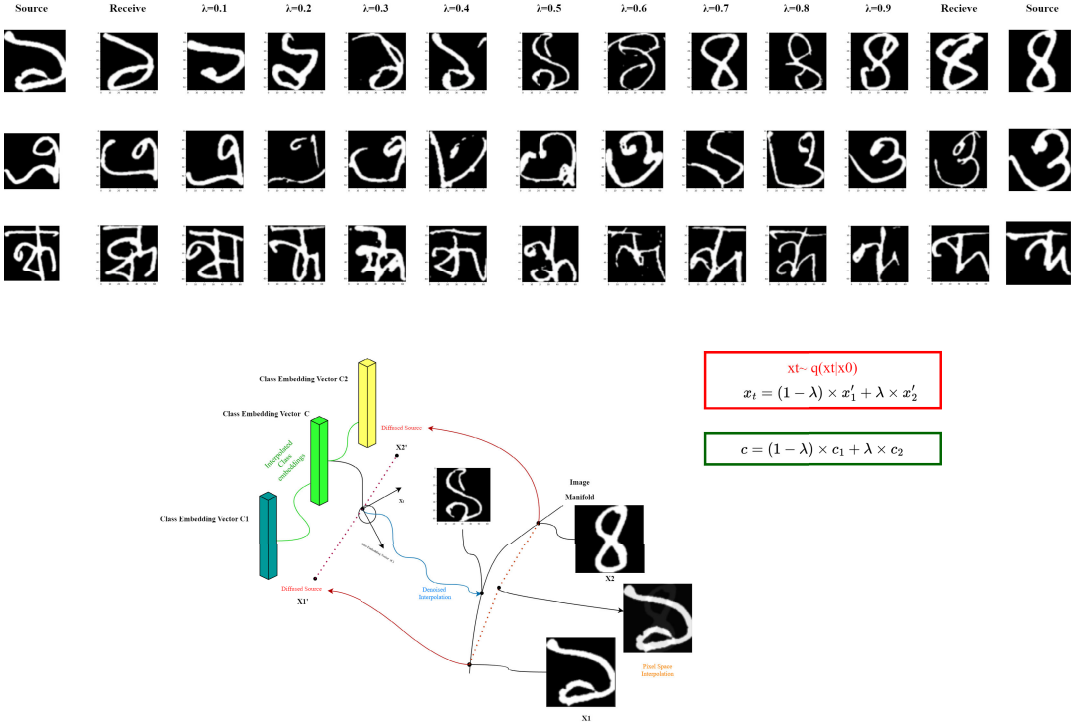

FIGURE 5. Interpolation of Banglalekha-Isolated images with 1000 timesteps of diffusion.

One of the most used performance metrics for evaluating generated images are FID (Fréchet Inception Distance), MS-SSIM (Multi-Scale Structural Similarity Index Measure), and LPIPS (Learned Perceptual Image Patch Similarity). These metrics are particularly effective when assessing the quality of generated images that can vary based on the input noise vector.

1) FRÉCHET INCEPTION DISTANCE (FID)

To represent the diversity of generated images, Fréchet Inception Distance (FID) was proposed by [26], claiming that it is compatible with human judgment. FID is a symmetric measure of the distance between two image distributions in the Inception-V3 [27] latent space. A lower FID score signifies better performance. FID was

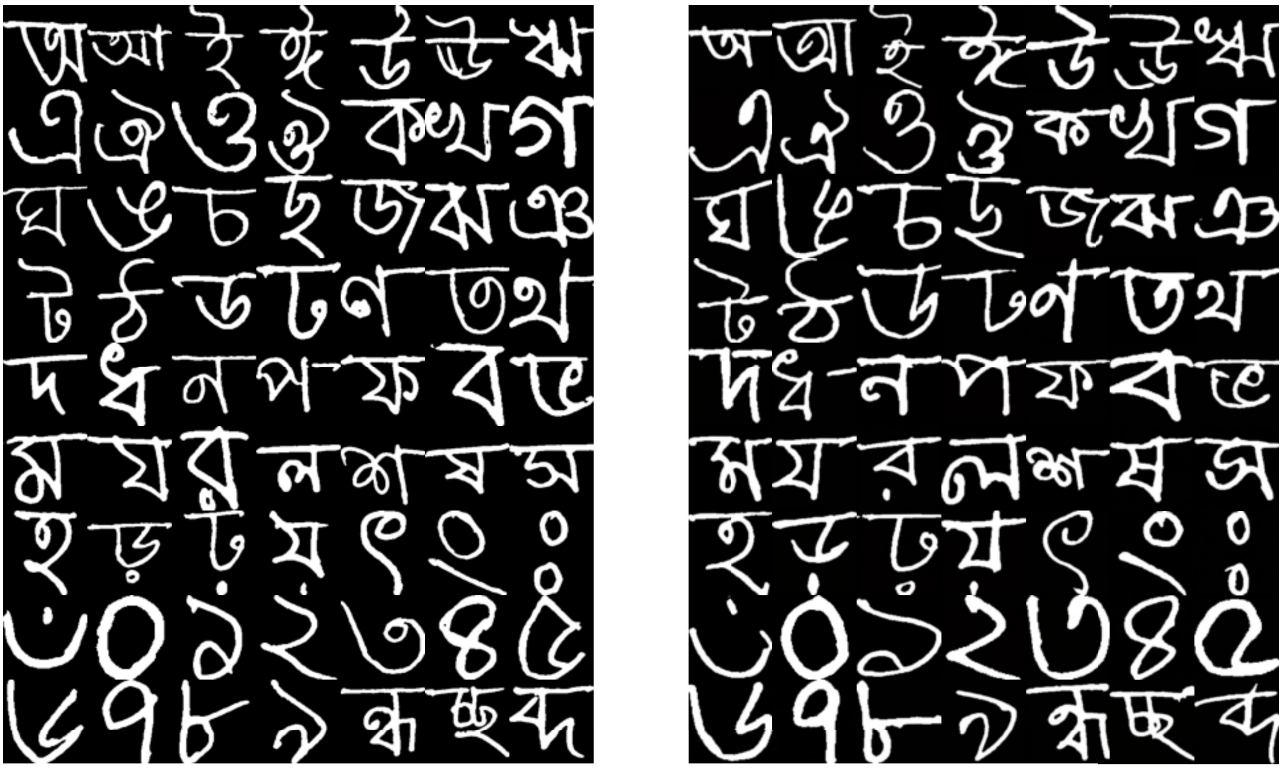

FIGURE 6. Comparing real character images and generated character images using Okkhor-Diffusion model from BanglaLekha-Isolated dataset: A visual analysis.

used as the default metric for overall sample quality comparisons.

To calculate the Fréchet Inception Distance (FID) score, the first step is to utilize a pre-trained Inception-V3 model. The output layer of the model is removed, and the resulting activations from the last pooling layer, which performs global spatial pooling, are taken as the feature representations of the images. The feature representations comprise a coding vector or feature vector, which is a vector with 2,048 activations assigned to each image. A feature vector consisting of 2,048 dimensions is subsequently generated to anticipate the representation of real images inside the problem domain, serving as a benchmark for understanding the visual characteristics of real images. Subsequently, feature vectors can be computed for synthetic images. The result is two sets of 2,048 feature vectors: one representing real images and the other representing generated images. These feature vectors allow for a quantitative comparison of the distributions of real and fake images using the FID score. The FID score is calculated using Equation (10) taken from the paper [26]:

$$d^2 = \|\mu_1 - \mu_2\|^2 + \text{Tr}(C_1 + C_2 - 2\sqrt{C_1 \cdot C_2}) \quad (10)$$

Following the paper [26], FID was calculated by randomly selecting 50,000 real images and 50,000 randomly sampled generated images from triad datasets individually. The evaluation of the experimental models on a triad of Bangla datasets shown in Figure 12, comprising the Bangla-Lekha Isolated dataset, the Ekush dataset, and the CMATERdb dataset, reveals that the Fréchet Inception Distance (FID) score for the Bangla-Lekha Isolated dataset is 6.225, while the FID score for the Ekush dataset is 9.86 and the FID score for the CMATERdb dataset is 10.346 while training using DDPM+AdamW. These scores provide a foundational benchmark that can serve as a guiding reference for future improvements. It is also clear from Figure 12 that the proposed Okkhor-Diffusion (DDPM+Lion) model outperforms DDPM+AdamW on the relatively standardized Bangla-Lekha Isolated dataset.

2) MULTI-SCALE STRUCTURAL SIMILARITY INDEX MEASURE (MS-SSIM)

Multi-Scale Structural Similarity Index Measure (MS-SSIM) was used to assess the intra-class diversity of the generated samples compared to the training samples [28]. Multi-Scale Structural Similarity Index Measure (MS-SSIM) has emerged as a reliable metric for measuring the perceptual diversity of image samples, as successfully applied in the papers of [20] and [21]. The score ranges from 0.0 to 1.0, with higher values indicating a greater degree of structural similarity between the two images [21]. Therefore, images with greater diversity ought to have lower MS-SSIM scores, whereas images with less diversity ought to have higher MS-SSIM scores. Multi-Scale Structural Similarity Index Measure (MS-SSIM) scores

FIGURE 7. Comparing real character images and generated character images using StyleGAN2-ADA model from BanglaLekha-Isolated dataset: A visual analysis.

are computed for 100 randomly selected image pairs per class, in total of 8400 images, and then illustrate the mean MS-SSIM scores for both training and generated samples.

The Multi-Scale Structural Similarity Index Measure (MS-SSIM) score has been employed as a quantitative metric to evaluate and compare the performance of two different models on three distinctive datasets. The databases comprise several types of characters in the Bengali language, particularly basic characters, numerals, and compounds. Figure 16, Figure 17 and Figure 18 demonstrate the efficacy of the Multi-Scale Structural Similarity Index (MS-SSIM) in accurately quantifying the structural similarity between pairs of images belonging to different classes of characters. These figures provide graphical illustrations that highlight how well the MS-SSIM metric measures the structural similarities among different types of characters, notably basic characters, digits, and compounds. The mean MS-SSIM scores have also been calculated for real and generated images, represented as blue and orange bars in Figure 14, exhibit high variability across the 84 classes of the BanglaLekha-Isolated dataset, 110 classes of Ekush dataset and 221 classes of CMATERdb dataset. By computing the mean MS-SSIM scores separately for digits, basic characters, and compound characters, insights into the generative model's creative abilities can be gained. The analysis is extended to include two other datasets, Ekush and CMATERdb, and found that the performance of our approach was exceptional. As it can be seen, the Okkhor-Diffusion model generated real and synthetic samples that exhibited different levels of diversity across the isolated Bangla handwritten characters.

The L2 Distance metric employs the comparison of pixel values to quantify the dissimilarities in structure between two images, particularly a sample image and a reference image. To surpass prior methods, the paper [35] devised a measure that emulates the discerning abilities of the human visual perception system, renowned for its proficiency in recognizing structural details within a given scene. The Structural Similarity Index Measure (SSIM) metric is capable of identifying three distinct metrics from an image: Luminance $l(x, y)$, Contrast $c(x, y)$, and Structure $s(x, y)$, as shown in Equation (11).

$$\text{SSIM}(x, y) = [l(x, y)]^{\alpha} \cdot [c(x, y)]^{\beta} \cdot [s(x, y)]^{\gamma} \tag{11}$$

where $\alpha > 0, \beta > 0, \gamma > 0$ denote the relative importance of each of the metrics to simplify the expression.

However, taking the reference and distorted image signals as the input, the system iteratively applies a low-pass filter and downsamples the filtered image by a factor of 2. The authors from the paper [28] index the original image as Scale 1, and the highest scale as Scale M, which is obtained after $M - 1$ iterations. At the j-th scale, the contrast comparison and the structure comparison are calculated and denoted as $c_j(x, y)$ and $s_j(x, y)$, respectively. The luminance comparison is computed only at Scale M and is denoted as $l_M(x, y)$.

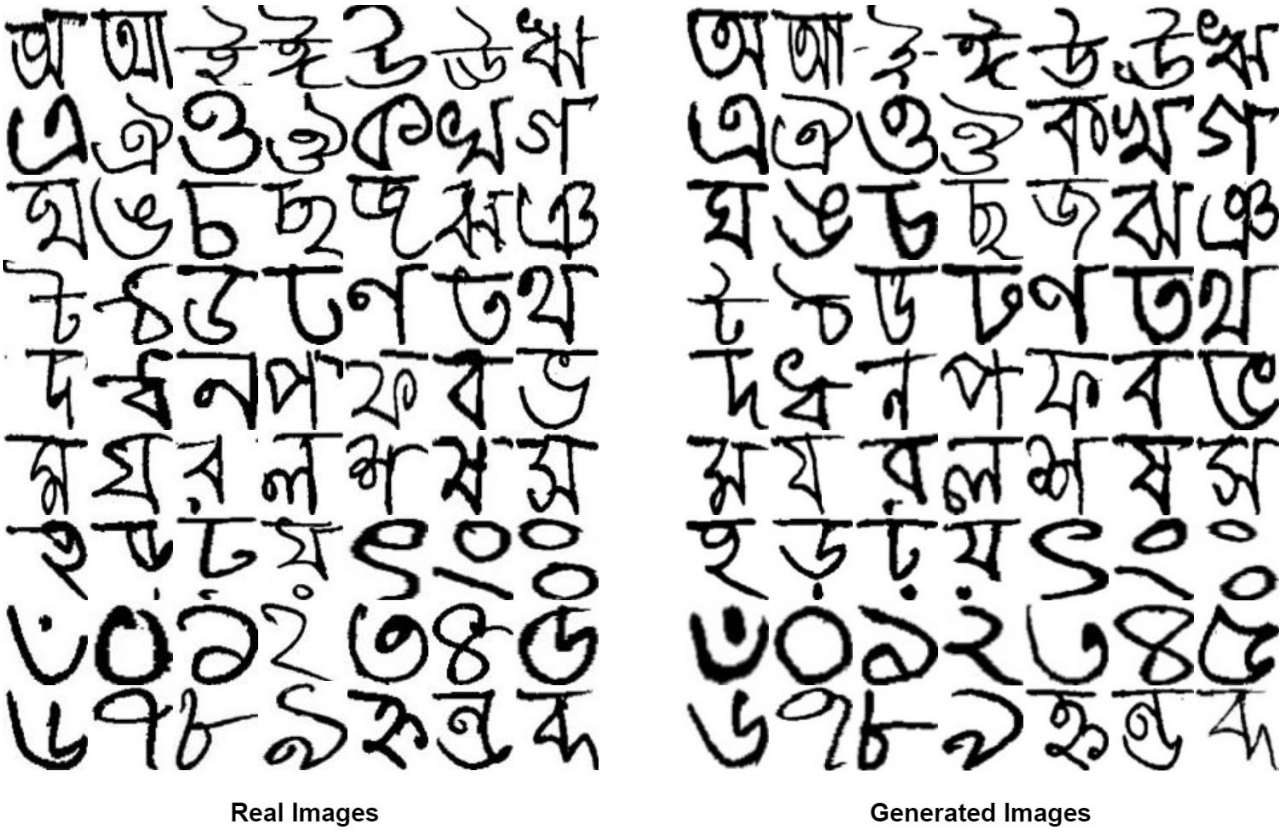

FIGURE 8. Comparing real character images and generated character images using Okkhor-Diffusion model from Ekush dataset: A visual analysis.

TABLE 4. Comparison between the models across different datasets for performance metrics: A comprehensive analysis.

Dataset	Model	FID↓	Mean MS-SSIM↓		LPIPS↓	BCAFID↓
			Real Images	Fake Images		
BanglaLekha-Isolated	**Okkhor-Diffusion**	**5.426**	**0.177**	**0.178**	0.3634	**10.388**
BanglaLekha-Isolated	DDPM+AdamW	6.744	0.185	0.178	0.3754	37.678
BanglaLekha-Isolated	StyleGAN2-ADA	87.519	0.180	0.892	**0.3254**	28.698
Ekush	**Okkhor-Diffusion**	**6.679**	**0.186**	**0.185**	**0.3079**	**28.762**
Ekush	DDPM+AdamW	9.862	0.184	0.186	0.3078	30.993
Ekush	StyleGAN2-ADA	70.709	0.184	0.926	0.3152	62.594
CMATERdb	**Okkhor-Diffusion**	**18.581**	**0.184**	**0.191**	**0.3126**	**29.063**
CMATERdb	DDPM+AdamW	20.696	0.187	0.190	0.3117	38.463
CMATERdb	StyleGAN2-ADA	26.744	0.186	0.585	0.3637	33.624

Equation (12) outlines the process of computing MS-SSIM.

$$\text{MS-SSIM}(x,y) = [l_M(x,y)]^{\alpha_M} \prod_{j=1}^{M} [c_j(x,y)]^{\beta_j} [s_j(x,y)]^{\gamma_j} \tag{12}$$

3) LEARNED PERCEPTUAL IMAGE PATCH SIMILARITY (LPIPS)

The LPIPS [32] score is computed by comparing the image patches of real and synthetic images. LPIPS utilizes activations of a layer from a pre-trained Alexnet model. A lower LPIPS score indicates higher performance. Since LPIPS has been proven to correlate well with human perceptual judgment, LPIPS was used to evaluate the results quantitatively. Human evaluation is subject to bias and human error. LPIPS provides us a mechanism to objectively assess the generated images. LPIPS was calculated by individually selecting 50,000 sample images and 50,000 generated images at random from the triad datasets. The results from Table 4 and Figure 13 indicate that the proposed Okkhor-Diffusion model performs competitively in terms of LPIPS across all datasets.

FIGURE 9. Comparing real character images and generated character images using StyleGAN2-ADA model from Ekush dataset: A visual analysis.

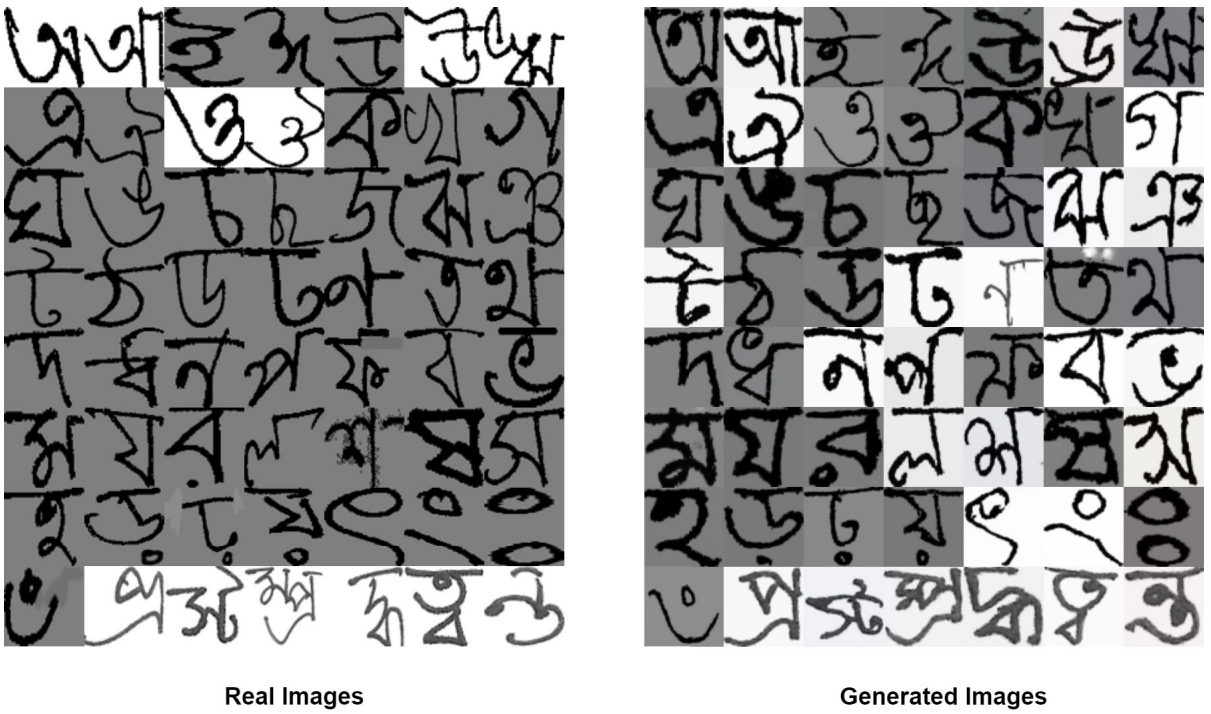

FIGURE 10. Comparing real character images and generated character images using Okkhor-Diffusion model from CMATERdb dataset: A visual analysis.

4) BANGLA CHARACTER AWARE FRÉCHET INCEPTION DISTANCE (BCAFID)

As the classical FID metric employs an Inception-V3 classifier trained on imagenet dataset; it is not adequate for evaluating generated Bangla Handwritten character images. The classes of the Imagenet dataset do not contain any types of handwritten English characters, let alone Bengali characters [30]. Therefore, the activation statistics from

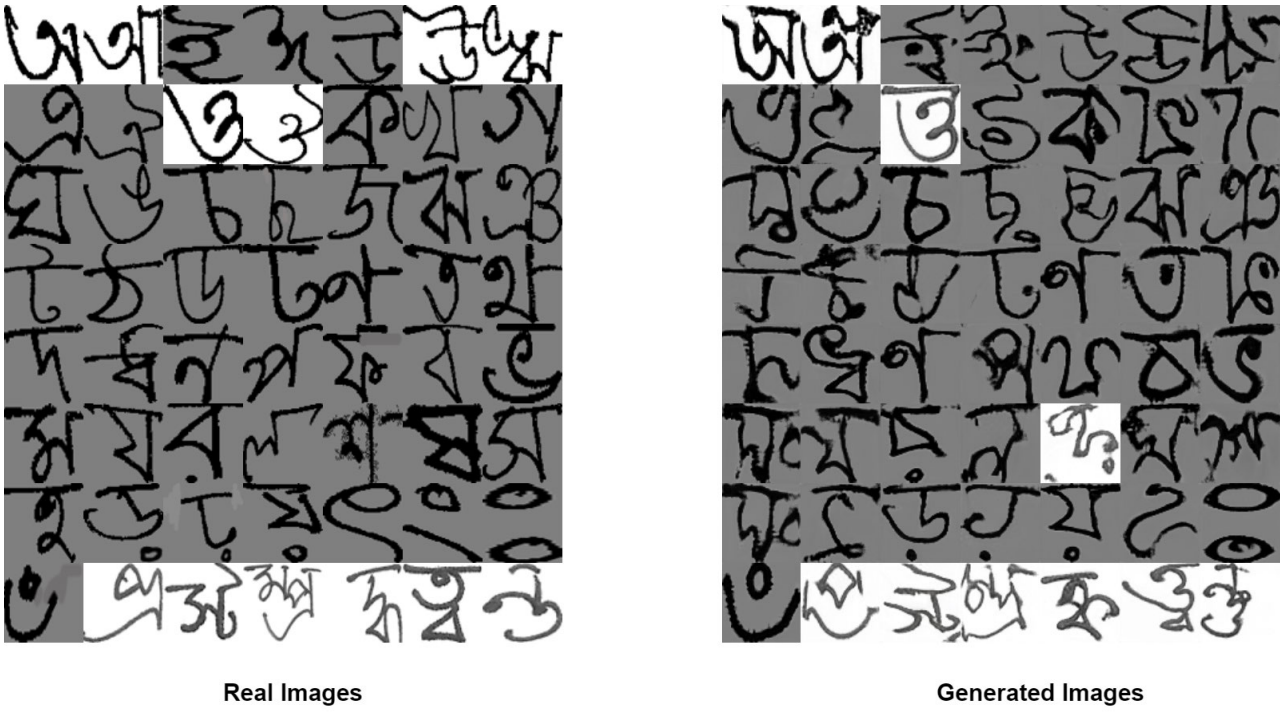

FIGURE 11. Comparing real character images and generated character images using StyleGAN2-ADA model from CMATERdb dataset: A visual analysis.

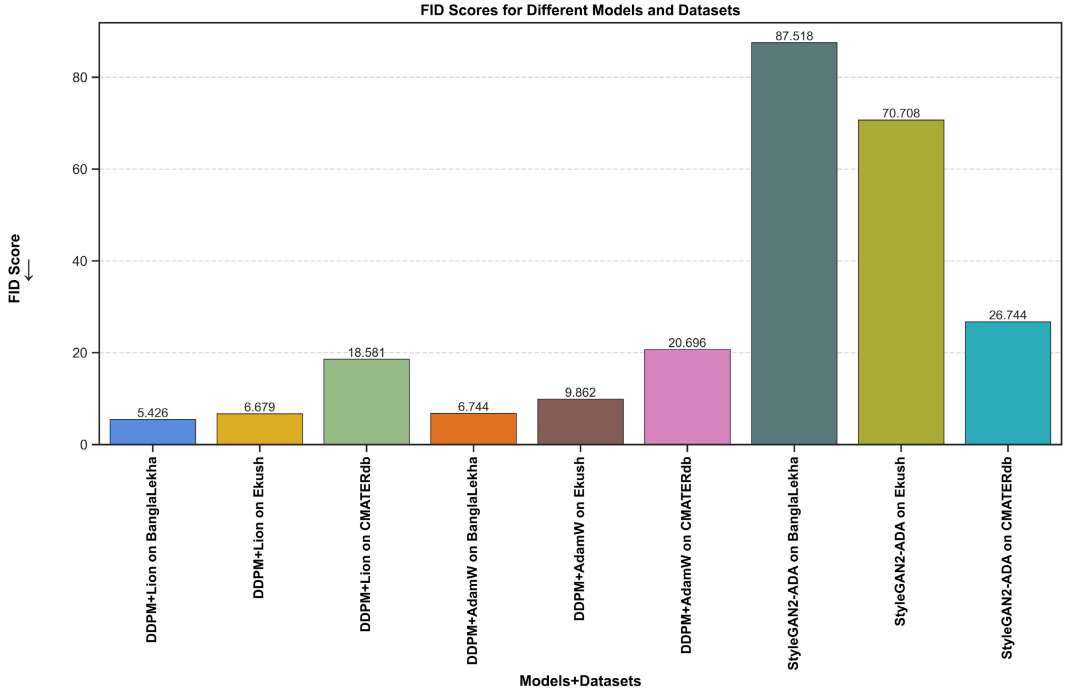

FIGURE 12. A comparison of the FID scores of models trained on the Bangla-Lekha Isolated, Ekush, and CMATERdb datasets.

the intermediate layers are biased towards images akin to Imagenet classes and are inaccurate for handwritten text. In order to address this issue, this paper proposed a modified variant of FID that is referred to as **Bangla Character Aware**

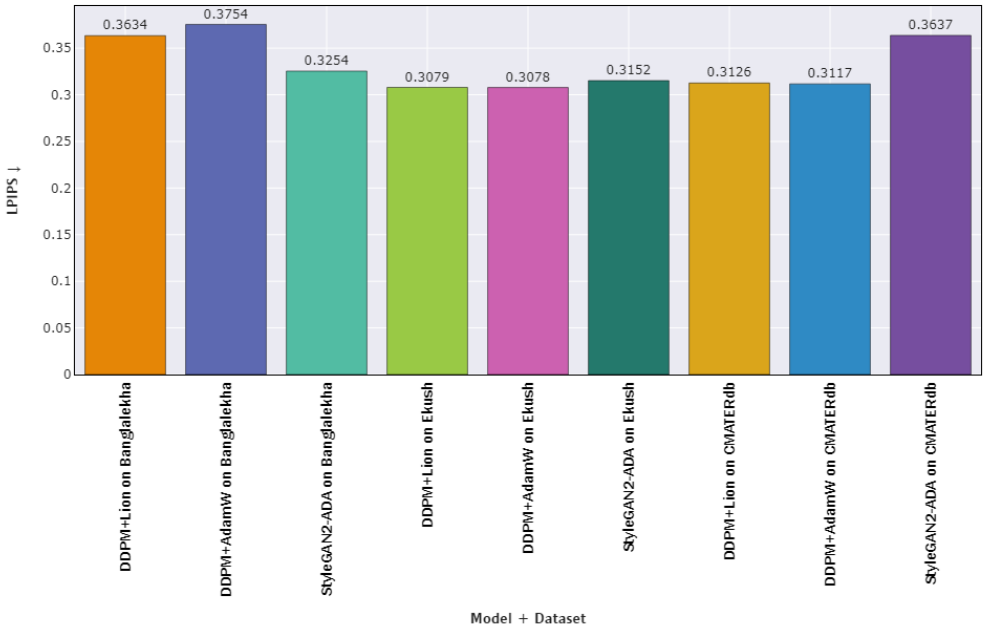

FIGURE 13. A comparison of the LPIPS scores of models trained on the Bangla-Lekha Isolated, Ekush, and CMATERdb datasets.

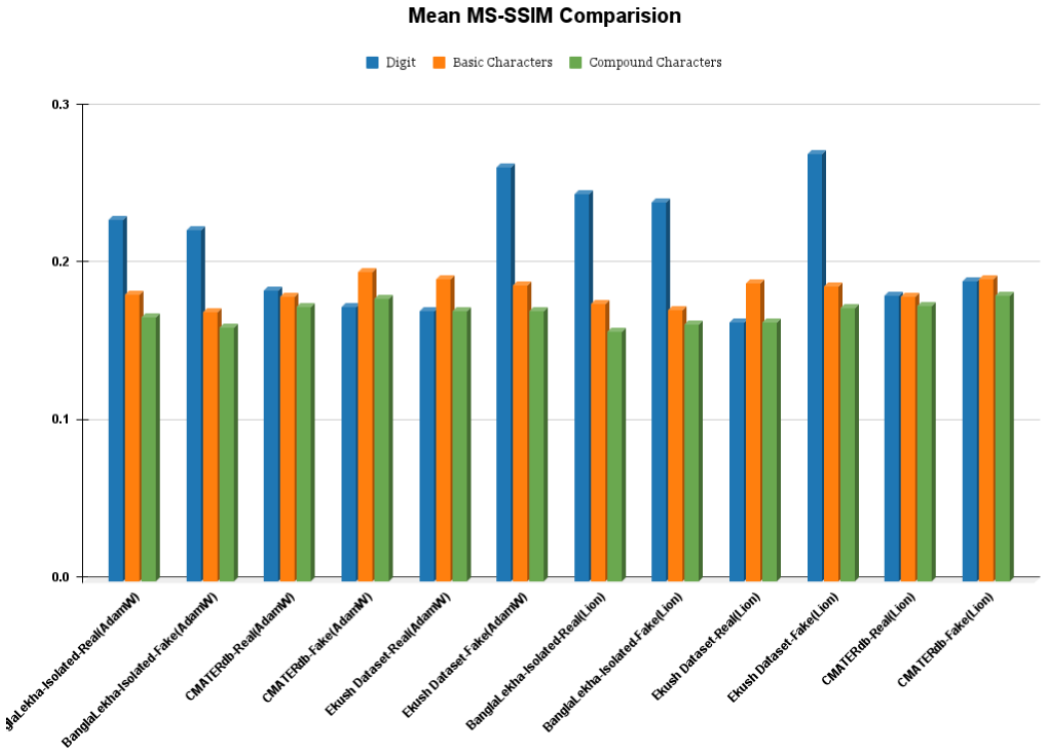

FIGURE 14. Mean MS-SSIM comparison of models on all datasets: digits, characters, and compound characters.

Fréchet Inception Distance (BCAFID). The pre-trained Inception-V3 model was utilized and fine-tuned using the BanglaLekha-Isolated dataset. As FID compares a sample of actual images to a sample of generated images, the distance between the distribution of activations for specific deep layers in an Inception-V3 is calculated. If activation distributions are similar, then image distributions are assumed to be similar as well. Training the model with BanglaLekha-Isolated ensures

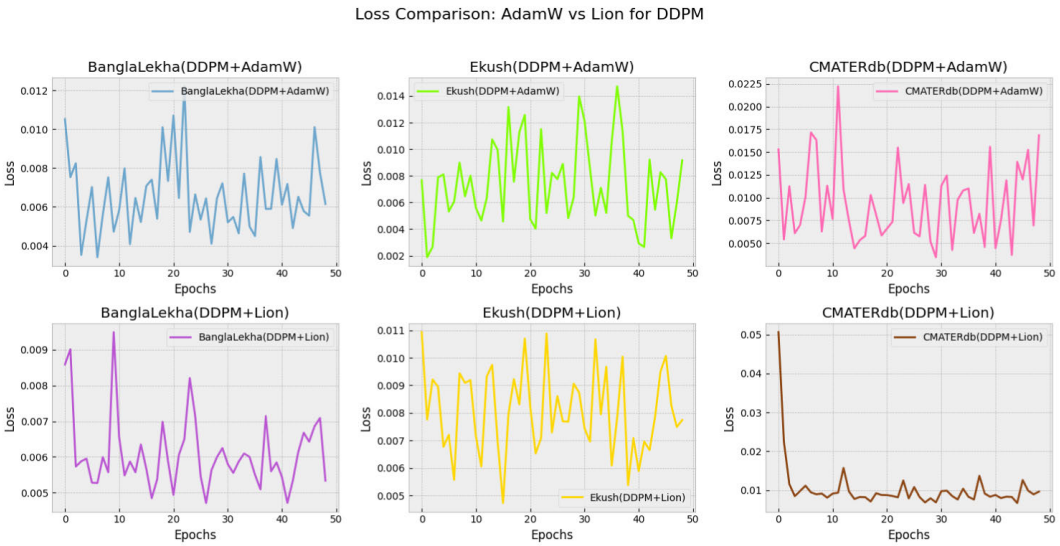

FIGURE 15. Plot of training loss at different epochs for different datasets and models.

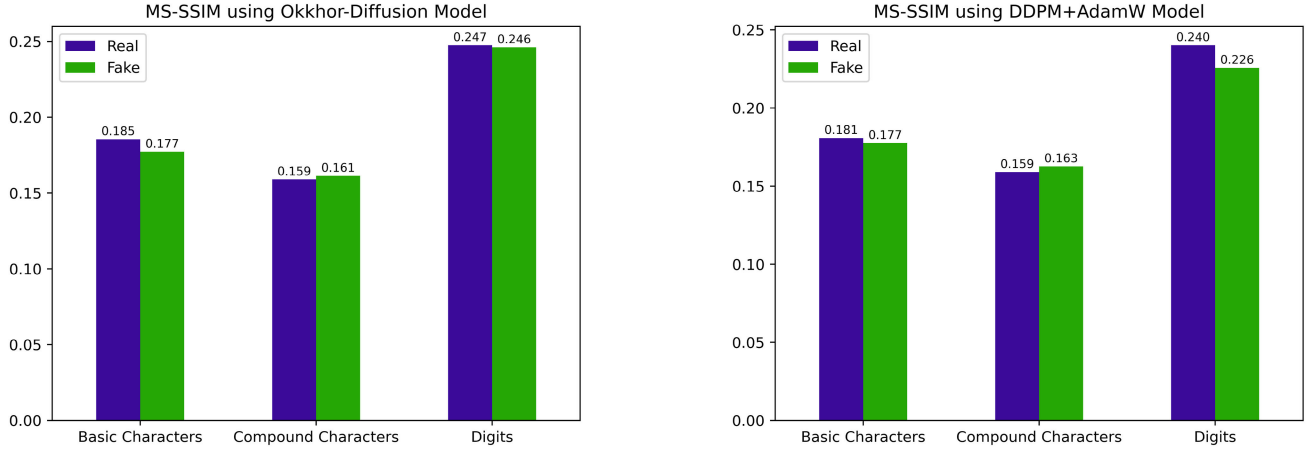

FIGURE 16. MS-SSIM on BanglaLekha-Isolated dataset.

that it can capture the similarity of activation distribution between actual and generated Bangla handwritten character images, making it a meaningful variant of the FID metric for Isolated Bangla Handwritten Character generation. For computing BCAFID, real Bangla character images, B_R and fake Bangla character images, B_F are fed into the Bangla character trained Inception-V3 network to obtain features f_R and f_F from intermediate layers. The mean of f_R, μ_R and the mean of f_F, μ_F are calculated. Also, the covariance matrix of f_R, Σ_R and the covariance matrix of f_F, Σ_F are computed. The mean and covariance matrices obtained from the feature vectors are used to calculate BCAFID shown in Equation (13). Here, $\mathcal{L}_2$ represents the Euclidean distance and *Trace* sums up the elements e_{ij} of a matrix M where $i = j$.

$$BCAFID = \mathcal{L}_2(\mu_R, \mu_F) \odot \mathcal{L}_2(\mu_R, \mu_F) + Trace(\Sigma_R + \Sigma_F - 2(\Sigma_R \cdot \Sigma_F)^{(1/2)}) \quad (13)$$

Table 4 provides a comprehensive comparison of all the metrics used in this study to evaluate the DDPM+AdamW model and Okkhor-Diffusion model. The scores obtained by DDPM+AdamW across a triad of datasets show DDPM's competitive capability of producing high-quality and diverse Bengali Isolated Handwritten characters. The proposed Okkhor-Diffusion model resulted in a substantial improvement in all of the evaluation metrics. Okkhor-Diffusion is trained on the standardized BanglaLekha-Isolated dataset

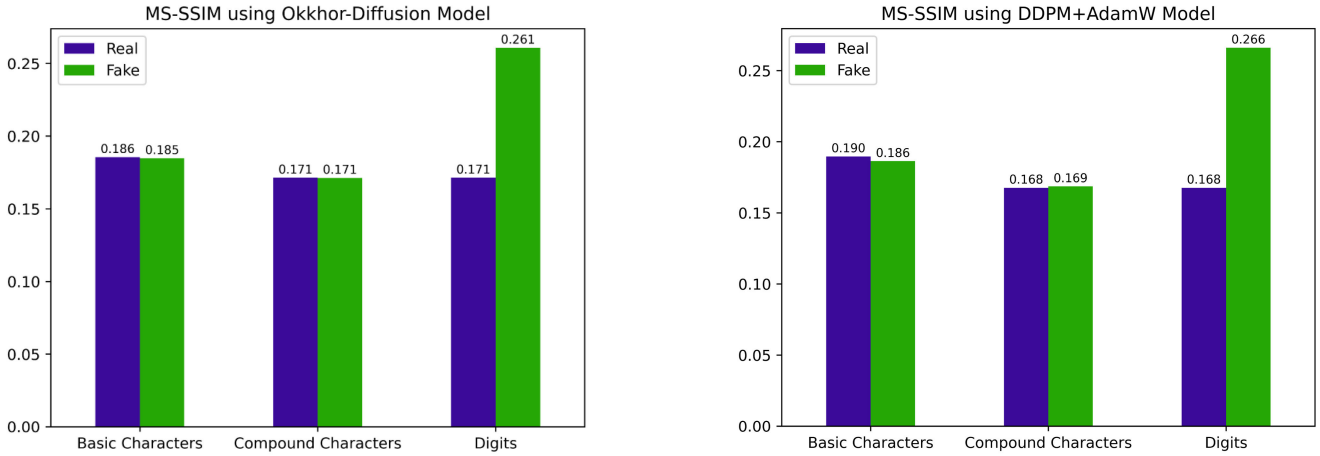

FIGURE 17. MS-SSIM on Ekush dataset.

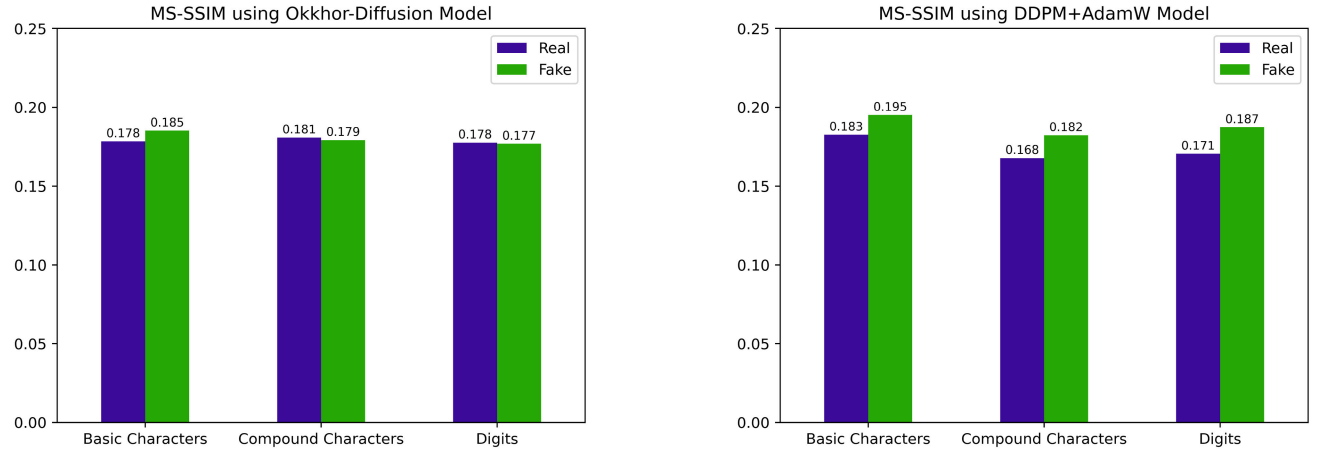

FIGURE 18. MS-SSIM on CMATERdb dataset.

using the Lion optimizer, and the results indicate that it performs relatively better than DDPM+AdamW.

From Figure 15, it is clear from the loss plot that Okkhor-Diffusion exhibits a more stable training performance compared to DDPM+AdamW.

VIII. DISCUSSION

Due to limited computational resources, this paper only explores the generation of 64 × 64 Bangla characters. The FID score of Okkhor Diffusion is not as great as the FID score obtained by state-of-the-art English handwritten character generation models. Our model requires an expensive GPU to run on a local device, limiting its reach to many users. In the future, generating higher resolution images like 128 × 128, 256 × 256, etc. using DDPM efficiently on less expensive hardware can be explored. Since the Okkhor-Diffusion model produces state-of-the-art images, it can be used to create a dataset with even greater variation and an abundance of handwritten Bangla characters. Such a dataset can potentially aid in improving the performance and acceptability of Handwritten Bangla Character Recognition (HBCR).

IX. CONCLUSION

This paper proposes a novel framework using a novel Denoising Diffusion Probabilistic Model architecture combined with Lion optimizer for generating 64 × 64 handwritten Bangla isolated character images. No previous literature was found that actually measures the qualitative results of Bangla handwritten isolated character generation using state-of-the-art image generation metrics such as FID. Therefore, the results are compared across three popular Bangla handwritten datasets: BanglaLekha-Isolated, CMATERdb, and Ekush. It is evident from Figure 6 and other quantitative measures at Table 4 that Okkhor-Diffusion has ensured remarkable results in terms of relevancy and image quality compared to the state-of-the-art GAN model, StyleGAN2-ADA. This paper also introduced and used a new metric called BCAFID to address the limitations of evaluating generated Bangla handwritten

character images. It can be concluded from the results at Table 4 that DDPM with Lion optimizer yields high-quality results for synthesizing Bangla handwritten character images.

ACKNOWLEDGMENT

The authors acknowledge hardware support from the Department of Computer Science and Engineering, University of Asia Pacific. They would like to thank the Advanced Machine Intelligence Research Laboratory (AMIR Lab) for research support and supervision.

REFERENCES

[1] D. Vaughan, (Nov. 2020). *The World's 5 Most Commonly Used Writing Systems*. Encyclopedia Britannica. [Online]. Available: https://www.britannica.com/list/the-worlds-5-most-commonly-used-writing-systems

[2] N. Das, K. Acharya, R. Sarkar, S. Basu, M. Kundu, and M. Nasipuri, "A benchmark image database of isolated Bangla handwritten compound characters," *Int. J. Document Anal. Recognit. (IJDAR)*, vol. 17, no. 4, pp. 413–431, Dec. 2014, doi: 10.1007/s10032-014-0222-y.

[3] T. A. Tani, M. M. A. Shibly, M. S. Hasan, N. Yeasmin, and S. Ripon, "Autoencoder and deep convolutional generative adversarial network in improving the performance of bangla handwritten character recognition," in *Deep Learning Applications in Image Analysis*. Springer, 2023, pp. 1–26.

[4] M. Islam, S. Shuvo, M. Nipun, R. B. Sulaiman, M. Shaikh, J. Nayeem, Z. Haque, M. Sourav, and A. Kareem, "Efficient approach to using CNN-based pre-trained models in Bangla handwritten digit recognition," in *Proc. Comput. Vis. Bio-Inspired Comput.*, 2023, pp. 697–716.

[5] M. Mortuza, S. Islam, M. Kabir, and U. Chong, "A convolutional neural network-based approach to recognize Bangla handwritten characters," in *Proc. Comput. Vis. Image Anal. Ind.*, 2023, pp. 150–163.

[6] M. Biswas, R. Islam, G. K. Shom, M. Shopon, N. Mohammed, S. Momen, and A. Abedin, "BanglaLekha-isolated: A multi-purpose comprehensive dataset of handwritten Bangla isolated characters," *Data Brief*, vol. 12, pp. 103–107, Jun. 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352340917301117

[7] A. Rabby, S. Haque, M. Islam, S. Abujar, and S. Hossain, "Ekush: A multipurpose and multitype comprehensive database for online off-line Bangla handwritten characters," in *Proc. 2nd Int. Conf.*, Solapur, India, Dec. 2019, pp. 149–158.

[8] R. Sarkar, N. Das, S. Basu, M. Kundu, M. Nasipuri, and D. K. Basu, "CMATERdb1: A database of unconstrained handwritten Bangla and Bangla–English mixed script document image," *Int. J. Document Anal. Recognit. (IJDAR)*, vol. 15, pp. 71–83, Feb. 2012.

[9] M. A. Bansal, D. R. Sharma, and D. M. Kathuria, "A systematic review on data scarcity problem in deep learning: Solution and applications," *ACM Comput. Surveys*, vol. 54, no. 10s, pp. 1–29, Jan. 2022, doi: 10.1145/3502287.

[10] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 6840–6851.

[11] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," 2020, *arXiv:2011.13456*.

[12] P. Dhariwal and A. Nichol, "Diffusion models beat GANs on image synthesis," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 8780–8794.

[13] F.-A. Croitoru, V. Hondru, R. T. Ionescu, and M. Shah, "Diffusion models in vision: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 9, pp. 10850–10869, 2023, doi: 10.1109/TPAMI.2023.3261988.

[14] L. Yang, Z. Zhang, Y. Song, S. Hong, R. Xu, Y. Zhao, W. Zhang, B. Cui, and M.-H. Yang, "Diffusion models: A comprehensive survey of methods and applications," 2022, *arXiv:2209.00796*.

[15] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 10674–10685.

[16] T. Wang, T. Zhang, B. Zhang, H. Ouyang, D. Chen, Q. Chen, and F. Wen, "Pretraining is all you need for image-to-image translation," 2022, *arXiv:2205.12952*.

[17] J. Austin, D. Johnson, J. Ho, D. Tarlow, and R. Van Den Berg, "Structured denoising diffusion models in discrete state-spaces," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 17981–17993.

[18] Z. K. Nishat and M. Shopon, "Synthetic class specific Bangla handwritten character generation using conditional generative adversarial networks," in *Proc. Int. Conf. Bangla Speech Lang. Process. (ICBSLP)*, Sep. 2019, pp. 1–5.

[19] M. M.-H.-Z. Abedin, T. Ghosh, T. Mehrub, and M. A. Yousuf, "Bangla printed character generation from handwritten character using GAN," in *Soft Computing for Data Analytics, Classification Model, and Control*. Springer, 2022, pp. 153–165.

[20] I. B. Mustapha, S. Hasan, H. Nabus, and S. M. Shamsuddin, "Conditional deep convolutional generative adversarial networks for isolated handwritten Arabic character generation," *Arabian J. Sci. Eng.*, vol. 47, no. 2, pp. 1309–1320, Feb. 2022.

[21] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 2642–2651.

[22] L. Weng, (Jul. 2021). *What Are Diffusion Models?*. [Online]. Available: https://lilianweng.github.io/posts/2021-07-11-diffusion-models/

[23] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 1–9.

[24] X. Chen, C. Liang, D. Huang, E. Real, K. Wang, Y. Liu, H. Pham, X. Dong, T. Luong, C.-J. Hsieh, Y. Lu, and Q. V. Le, "Symbolic discovery of optimization algorithms," 2023, *arXiv:2302.06675*.

[25] S. Barratt and R. Sharma, "A note on the inception score," 2018, *arXiv:1801.01973*.

[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," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1–12.

[27] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.

[28] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *Proc. Thrity-Seventh Asilomar Conf. Signals, Syst. Comput.*, vol. 2, 2003, pp. 1398–1402.

[29] P. Platen, S. Patil, A. Lozhkov, P. Cuenca, N. Lambert, K. R. Davaadorj, and M. T. Wolf. (2022). *Diffusers: State-of-the-art Diffusion Models*. GitHub Repository. [Online]. Available: https://github.com/huggingface/diffusers

[30] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[31] S. Bond-Taylor, A. Leach, Y. Long, and C. G. Willcocks, "Deep generative modelling: A comparative review of VAEs, GANs, normalizing flows, energy-based and autoregressive models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, pp. 7327–7347, 2021.

[32] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 586–595.

[33] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila, "Training generative adversarial networks with limited data," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 12104–12114.

[34] D. Gui, K. Chen, H. Ding, and Q. Huo, "Zero-shot generation of training data with denoising diffusion probabilistic model for handwritten Chinese character recognition," 2023, *arXiv:2305.15660*.

[35] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[36] M. Kang, J.-Y. Zhu, R. Zhang, J. Park, E. Shechtman, S. Paris, and T. Park, "Scaling up GANs for text-to-image synthesis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 10124–10134.

[37] M. Ding, Z. Yang, W. Hong, W. Zheng, C. Zhou, D. Yin, J. Lin, X. Zou, Z. Shao, and H. Yang, "Cogview: Mastering text-to-image generation via transformers," in *Proc. Adv. In Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 19822–19835.

[38] P. Esser, R. Rombach, and B. Ommer, "Taming transformers for high-resolution image synthesis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 12868–12878.

MD. MUBTASIM FUAD received the Bachelor of Science degree (Hons.) in computer science and engineering from the University of Asia Pacific, Dhaka, Bangladesh, Spring 2023. Currently, he is a Teaching Assistant with the CSE Department, University of Asia Pacific. He is a dedicated individual who prioritizes delivering work on time with a focus on quality. He has a strong interest in new technologies and possesses a collaborative mindset. He is an enthusiastic machine learning enthusiast, who has actively worked with generative models, particularly focusing on diffusion models and generative adversarial networks (GANs). His research interests include the fields of deep learning and computer vision, with a focus on exploring advancements and applications within these domains. Also, he has a strong desire to travel to uncharted parts of the world and interact with diverse cultures and people in order to acquire new experiences. He believes that such experiences will broaden his horizons and provide valuable insights that can enrich his personal and professional growth. Throughout his academic journey, he was awarded the Honourable Vice Chancellor Award six times for his outstanding GPA in each semester. He has actively participated in numerous inter and intra-university contests, including the 5th Inter Department Software and Hardware Carnival 2022, CSE ROBO EXPO 1.0, the Robotronics 2.0-An Inter-University Tech Fest, and the Intra Department CTF Contest-2022, where he received various achievements and awards. Additionally, he volunteered in the tech teams during the ICPC World Finals held in Dhaka.

A. FAIYAZ received the Bachelor of Science degree in computer science and engineering from the University of Asia Pacific, Dhaka, Bangladesh, Spring 2023. He is currently a passionate individual with an eagle eye on the latest software development and deep learning advancements. He received the Honourable Vice Chancellor Award three times. He is highly active in the Competitive Programming Arena and participated in the ICPC Programming Contest. In addition to that, he also participated in numerous hackathons. Furthermore, he volunteered in the tech teams during the ICPC World Finals held in Dhaka.

NOOR MAIRUKH KHAN ARNOB received the B.Sc. degree (Hons.) in computer science and engineering from the University of Asia Pacific, Dhaka, Bangladesh, in 2023. He is currently a Lecturer with the Department of CSE, UAP. He is the author of two published journal articles. He participated in the Intra-UAP Programming Contest 2020, the 2021 ICPC Asia Dhaka Regional Contest, the Robi Datathon 2.0, and the Code Samurai Online Preliminary Contest 2022. He aims to inspire students to solve real-world problems by employing their mathematical and programming skills. His research work is directed toward building a fair and inclusive society with the help of AI. His research interests include deep learning, natural language processing, computer vision, and generative models.

M. F. MRIDHA (Senior Member, IEEE) received the Ph.D. degree in AI/ML from Jahangirnagar University, in 2017. He is currently an Associate Professor with the Department of Computer Science, American International University-Bangladesh (AIUB). Before that, he was an Associate Professor and the Chairperson of the Department of CSE, Bangladesh University of Business and Technology. He was a Faculty Member with the CSE Department, University of Asia Pacific, and the Graduate Head, from 2012 to 2019. His research experience within both academia and industry, which results in more than 120 journals and conference publications. His research work contributed to the reputed journals of *Scientific Reports* (Nature), *Knowledge-Based Systems*, *Artificial Intelligence Review*, IEEE Access, *Sensors*, *Cancers*, and *Applied Sciences*. For more than ten years, he has been with the master's and bachelor's students as a Supervisor of their thesis work. His research interests include artificial intelligence (AI), machine learning, deep learning, natural language processing (NLP), and big data analysis. He was a program committee member of several international conferences/workshops. He served as an Associate Editor for several journals, including *PLOS One* journal. He also served as a Reviewer for reputed journals and IEEE conferences, such as HONET, ICIEV, ICCIT, *IJCCI*, ICAEE, ICCAIE, ICSIPA, SCORED, ISIEA, APACE, ICOS, ISCAIE, BEIAC, IC3e, ISWTA, *Coasts*, icIVPR, ICSCT, 3ICT, and DATA21.

ALOKE KUMAR SAHA received the B.Sc. degree (Hons.) in applied physics amp and electronics and the M.Sc. degree (thesis) in computer science from the University of Dhaka, in 1995 and 1997, respectively, and the Ph.D. degree in computer science and engineering from Jahangirnagar University, Savar, Dhaka, Bangladesh. He was a Lecturer with Queens University, from June 1997 to March 1999. In March 1999, he joined the University of Asia Pacific (UAP), Dhaka, as a Lecturer. He was the Head of the CSE Department, UAP, from 2008 to 2018. For more than 26 years, he has been working with undergraduate and graduate students as a Course Teacher and a Supervisor of their research works. He is currently a Professor with the Computer Science and Engineering (CSE) Department and the Director of the Institutional Quality Assurance Cell (IQAC), UAP. He has authored or coauthored 32 journal articles and 25 conference papers. He usually teaches courses on digital logic amp, system design, numerical methods, data structures, discrete mathematics, and computer graphics. His current research interests include algorithms, artificial intelligence (AI), machine learning, and natural language processing (NLP). He was the Chair of the Organizing Committee of the International Conference on Computer and Information Technology (ICCIT), in 2017. He was the Contest Director of the National Collegiate Programming Contest (NCPC), in 2016. Under his leadership, UAP hosted the International Collegiate Programming Contest (ICPC), in 2016 and 2017. He is the Chief of the Organizing Committee of the International Journal of Computer and Information Technology (IJCIT) (Department of CSE, UAP). He is a reviewer of different conferences and journals.

ZEYAR AUNG (Senior Member, IEEE) received the Ph.D. degree in computer science from the National University of Singapore, in 2006. From 2006 to 2010, he was a Research Fellow with the Institute for Infocomm Research (I2R), Agency for Science, Technology and Research (A*STAR), Singapore. In 2010, he joined Masdar Institute, which later became a part of Khalifa University, United Arab Emirates, as an Assistant Professor. He is currently an Associate Professor with the Department of Computer Science, Khalifa University. He is also a member with the Center for Secure Cyber-Physical Systems. His past research interests were include bioinformatics and chemoinformatics. His current research interests include data analytics, machine learning, and their applications in various domains, such as cyber security, social media, financial systems, renewable energy, and environmental science.