Arabian Journal for Science and Engineering (2022) 47:1309–1320
https://doi.org/10.1007/s13369-021-05796-0

RESEARCH ARTICLE-COMPUTER ENGINEERING AND COMPUTER SCIENCE

Conditional Deep Convolutional Generative Adversarial Networks for Isolated Handwritten Arabic Character Generation

Ismail B. Mustapha[1] · Shafaatunnur Hasan[1] · Hatem Nabus[1] · Siti Mariyam Shamsuddin[1]

Received: 24 April 2020 / Accepted: 29 May 2021 / Published online: 3 July 2021

Abstract
Being the basis on which several languages of the world are built, the historical relevance of the basic Arabic characters cannot be overemphasized. Unique in its many similar characters which are only distinguishable by dots, Arabic character recognition and classification has witnessed notable increase in research in recent times, particularly using machine learning-based approaches. However, little or no research exists on automatic generation of handwritten Arabic characters. Besides, the available databases of labeled handwritten Arabic characters are limited. Motivated by this open area of research, we propose a Conditional Deep Convolutional Generative Adversarial Networks (CDCGAN) for a guided generation of isolated handwritten Arabic characters. Experimental findings based on qualitative and quantitative results show that CDCGAN produce synthetic handwritten Arabic characters that are comparable to the ground truth, given a mean multiscale structural similarity (MS-SSIM) score of 0.635 as against 0.614 in the real samples. Comparison with handwritten English alphabets generation task further shows the capability of CDCGAN in generating diverse yet high-quality images of handwritten Arabic characters despite their inherent complexity. Additionally, machine learning efficacy test using CDCGAN-generated samples shows impressive performance with about 10% performance gap between real and generated handwritten Arabic characters.

Keywords Deep learning · Handwritten character generation · Generative adversarial networks · Arabic character recognition

1 Introduction

Although optical character recognition has been an active area of research for many years, automatic recognition of handwritten texts still remains a hard and open problem [1]. Addressing this category of problems is advantageous in that it facilitates the transcription of archived manuscripts into digital forms and enhances compact storage and search for contents within scanned documents. Like English and other Latin handwritten character recognition research, studies on handwritten Arabic recognition have increased considerably in recent years, considering the historical relevance of the 28 basic Arabic characters (see Table 1) to the Malay, Persian, Urdu and Turkish languages among others.

The Arabic characters are unique in their many similar looking characters that are only distinguishable by dots above or below a master stroke. For instance, characters "jīm" (ج), "ḥā'" (ح) and "khā'" (خ) are only separable by the presence or absence of a dot above or below the master stroke "ح" (see Table 1). The presence or absence of these dots affects the pronunciation and meaning these letters add to words. These minor differences between highly similar characters and the inherent variation in individual hand writing make handwritten Arabic character recognition challenging [2].

Recent advancements in computational technologies and algorithms have increased the application of deep neural networks in many research domains; consolidating past efforts by achieving state-of-the-art results in many challenging discriminative tasks. An essential requirement for optimal

✉ Ismail B. Mustapha
bmismail2@live.utm.my; ismail.b.mustapha@gmail.com

Shafaatunnur Hasan
shafaatunnur@utm.my

Hatem Nabus
hatemrun@gmail.com

Siti Mariyam Shamsuddin
mariyam@utm.my

1 Computer Science Department, School of Computing, Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia

Table 1 Arabic alphabets

خ	ح	ج	ث	ت	ب	ا
khāʾ	*ḥāʾ*	*jīm*	*thāʾ*	*tāʾ*	*bāʾ*	*ʾalif*
ص	ش	س	ز	ر	ذ	د
ṣād	*shīn*	*sīn*	*zayn*	*rāʾ*	*dhāl*	*dāl*
ق	ف	غ	ع	ظ	ط	ض
qāf	*fāʾ*	*ghayn*	*ʿayn*	*ẓāʾ*	*ṭāʾ*	*ḍād*
ي	و	ه	ن	م	ل	ك
yāʾ	*wāw*	*hāʾ*	*nūn*	*mīm*	*lām*	*kāf*

performance and generalization of deep learning models is the availability of large labeled data. However, data annotation and labeling remain expensive and time-consuming.

Deep learning for handwritten Arabic character recognition has equally received increased research attention of late [1–3], albeit the available databases of handwritten Arabic characters remain limited, both in accessibility and size [2, 4, 5]. An interesting unexplored research direction that this study explores is the generation of synthetic handwritten Arabic characters to alleviate concerns related to data access. More so that researchers have in recent time seen synthetic data as a safe and useful alternative to addressing barriers encountered in data access [6, 7]. In addition to offering an alternative data source to existing machine learning-based handwritten recognition systems, synthetically generated samples can be used to augment training for improved model generalization.

While generative models have been widely studied in machine learning, it has had less of an impact prior to the emergence of Generative Adversarial Networks (GANs) [8], a powerful framework with the capability of learning highly complex data distributions. Inspired by game theory, GANs consist of two competing models: a generative model which generates data similar to a known data distribution and a discriminative model which tries to differentiate the generated data samples from the generative model from real samples. Several GANs variants have been proposed in the wake of its advent with many real-life applications in many real-life tasks in several fields like computer vision, where it has been used for image synthesis, image translation, image super-resolution and image tagging among others [9, 10]. Despite its widespread use in other handwritten character generation tasks [11, 12], its performance on isolated handwritten Arabic character generation is yet to be reported in any literature. The paucity in large annotated handwritten Arabic character database also provides a strong motivation for us to formulate an isolated Arabic handwritten character generation problem. Additionally, pertinent literatures have shown that it is difficult for GANs to generate quality samples when trained on data samples with high variability as inherent in handwritten characters. Meanwhile, incorporation of class labels in GANs training has been found to produce models with improved stability and sample quality [9]. Thus, a class-conditioned generative adversarial network is proposed in this study. The contributions of this research summarized as follows;

- A guided generation of isolated handwritten Arabic character using Conditional Deep Convolutional Generative Adversarial Networks (CDCGAN).
- Qualitative and quantitative results show that CDCGAN generate Arabic characters that capture the intrinsic diversity that exist in the ones written by humans.
- Machine learning model trained on CDCGAN-generated synthetic data performs reasonably well on real test samples.

The remaining part of this paper is organized as follows. A listing of the available Arabic handwriting databases is given in Sect. 2 followed by a review of pertinent works in Sect. 3. Section 4 introduces GANs, conditional GANs and the proposed CDCGAN model. The handwriting datasets and the experimental setup are also described in this section. In Sect. 5, the experimental results are presented before discussion in Sect. 6. Finally, Sect. 7 concludes the paper.

2 Arabic Handwriting Databases

Several Arabic handwriting databases, ranging from Arabic digits to words databases, have been used for automatic recognition tasks over the past years. However, compared to its English and Chinese handwriting counterpart, research in Arabic handwriting recognition has witnessed lesser research attention [4, 13]. A listing of the commonly used Arabic handwriting databases is provided in Table 2 alongside other details. The table shows that accessibility to most of the available databases is limited with only a few being freely accessible while majority require permission to be sought before access can be granted. The size of the available databases also remains limited except the AHDBase/MADBase which is comparable to MNIST [14], a popular benchmark for Latin handwritten digits. In all, a freely

Table 2 Arabic handwriting databases

Source	Databases	Type (size)	Availability	Remark
Pechwitz [43]	IFN/ENIT	Word (26,459) images	Accessible with permission	These are only word images of city names
El-Sherif and Abdelazeem [44]	AHDBase/ MADBase	Digit (70,000) images	Accessible	Suitable only handwritten digits recognition tasks
Das [45, 46]	CMATERDB 3.3.1	Digit (3000) images	Freely Accessible	This dataset is only suitable for handwritten Arabic digit recognition tasks
Elzobi [13]	IESK-ArDB	Manuscript pages (285), Word (6000) and Segmented character (8000) images	Accessible with permission	The characters are unlabeled and not in their isolated form
Lawgali [47]	HACDB	Character (6600) images	Inaccessible URL	Limited sample size and include characters in their connected form
Che [7], Torki [15]	AIA9K	Character (8737) images	Accessible with permission	Limited sample size, requires permission to access
El-Sawy [1]	AHCD	Character (16,800) images	Freely Accessible	Written by adult and highly referred
Altwaijry and Al-Turaiki [4]	Hijja	Character (47,434) images	Freely Accessible	Written by children and also includes each character in their connected form

accessible unified benchmark for handwritten Arabic character recognition tasks still remains elusive.

The HACDB, AIA9K [15], AHCD [1] and Hijja [4] databases, as shown in Table 2, are the only relevant databases for Arabic handwritten characters. However, only AHCD and Hijja are freely available without accessibly constraints. The AHCD database has been preferred over Hijja for reasons that include: 1. The Hijja database is a collection of children handwriting as against adult in AHCD. In fact, it was specifically designed for children handwriting character recognition problem 2. Hijja database includes the connected forms of each character which further makes several different characters highly indistinguishable. Additionally, the AHCD database has been highly referred in deep learning-based handwritten character recognition research.

3 Related Works

Handwritten text generation has, in the past, been achieved through generative sequential models, notably among which is [16] where the synthesis of handwritten texts with recurrent neural networks was introduced. A key drawback of these approach is that they are not guaranteed to produce natural looking text written by human. Besides, getting the required stroke data is practically more challenging compared to images of handwriting. Alternatively, generating handwritten character/text using handwriting images have become increasingly rife due to recent advances in deep learning methods. Following the advent and initial application of GANs to the generation of Latin handwritten digits [8, 9], several recent studies have focused on handwritten character/text generation.

Kong and Xu [17] conducted a comparative analysis of convolutional GANs (cCGAN), convolutional variational autoencoder (cCVAE) and a hybrid of the two, all conditioned on GBK encoding for the generation handwritten Chinese characters. Qualitative results reveal that cCGAN produces sharp images but suffers mode collapse while cCVAE preserves the diversity of generated images. However, training the hybrid of the two models was unstable and resulted in blurry images. Chang et al. [11] employed cycleGAN [18] in the synthesis of isolated handwritten Chinese characters from an existing printed font using a style transfer approach. In another closely related study, an hierarchical GANs featuring a hierarchical generative transfer network and hierarchical discriminative network for typeface transformation of Chinese handwritings was proposed in [19]. Experimental findings show the proposed model transfers styles from varying font styles to human handwritings impressively. Liu et al. [20] also presented a multi-scale multi-class GANs that is conditioned on writing styles and shapes for the generation of Chinese handwritten characters. While experimental findings show the generated images as realistic and natural, the diversity of the dataset appears fairly limited given that it was written by one person.

One of the earliest works on GANs-based handwritten English text generation is [12] where an adversarial generation of images of handwritten words conditioned on their embedding was proposed. In addition to the to the integration of an auxiliary recognizer to the network, a bidirectional long short-term memory (LSTM) recurrent neural network was used to extract the embedding of words to be generated and fed as conditional information alongside noise input to the generator networks. While the quantitative results are modestly impressive, generated images of Arabic and French words were notably blurry. Contrary to images of handwritten character generation, handwriting GANs (HWGANs) were proposed for generating handwriting stroke data in [21]. A stacked architecture of Convolutional Neural Networks-LSTM-Feedforward Neural Networks (CNN-LSTM-FNN) was used as discriminator while the recurrent neural networks handwriting generator proposed by [16] was used to synthesis stroke data. Input to the discriminator takes the form of Path Signature Features. Experimental findings show the efficacy of this approach in the generation of realistic handwritten text data. A semi-supervised GANs framework called ScrabbleGAN was proposed in [22]. In addition to the discriminator, this GANs framework has an added auxiliary supervised recognizer which also conditions inputs to the generator with character filters to allow generation of varying lengths of handwritten texts. Experimental findings show that given an input French and English texts, the proposed approach is capable of rendering them in different styles. In another related study, [23] proposed a GANs variant conditioned on latent style vectors and texts of arbitrary length to generate entire lines of handwriting. This model incorporates a number of discriminative networks among which is the spacing network used for predicting the space between texts. Comparison of the qualitative and quantitative results of this approach with results from ScrabbleGAN [22] and [12] shows the superior performance of this method. Similarly, a GANs conditioned on textual content and calligraphic style features (GANwriting) was presented in [24] for the synthesis of realistic word images is different styles. Apart from the discriminator network, two auxiliary classifiers, an attention-based word recognizer and style classifier, are incorporated into this GANs architecture to guide the generation of highly realistic samples of handwritten words. More recently, [25] proposed Handwriting Imitation GANs (HiGAN) that generates arbitrary-length handwritten text through imitation of calligraphic styles of both randomly sampled and reference images. This GANs framework incorporates a style encoder to disentangle calligraphic styles from reference handwriting. Although it is not clear if HiGAN is capable of inserting spaces between texts as in [23], the qualitative and quantitative results show it outperforms ScrabbleGAN and GANwriting.

Similarly, generation of Bangla handwritten digits and characters has been reported in a couple of relevant literatures with impressive qualitative findings [26, 27]. Summarily, representative generative models of varying architectures have been proposed for the generation of handwritten texts, notably for English and Chinese handwriting, with impressive results. However, little or no work has explored Arabic character generation. Besides, generation of plausible yet diverse images of handwritten texts remains an active area of research [24]. Thus, a Conditional Deep Convolutional Generative Adversarial Network is proposed here for generation of diverse yet natural looking Arabic handwriting characters.

4 Materials and Methods

4.1 Generative Adversarial Networks

Generative Adversarial Networks [8] are a generative modeling framework which uses a game-theoretic approach in its data generation. The framework in its basic form consists of two feed forward neural networks, a generative model G and an adversary D, also known as the discriminative model. While G maps random samples, z, from a known prior $P(z)$ to the given data space $P\text{data}(x)$ using function $G(z)$, $D(x)$ outputs a probability that x is a data point from the given data space rather than the generative model, G. G is trained to lead D into believing that the samples it generates are from the data distribution. Thus, G and D are trained via an adversarial objective as in a two-player min–max in Eq. 1.

$$\min_G \max_D f(D,G) = \mathbb{E}_{x\sim P\text{data}(x)}\left[\log D(x)\right] + \mathbb{E}_{x\sim P(z)}\left[\log\left(1 - D(G(z))\right)\right] \tag{1}$$

4.2 Conditional Generative Adversarial Networks

Given that the data being generated by the basic GANs are random with no control on the mode or order of generated data, a conditional variant of GANs was proposed in [9]. Any auxiliary/conditional information y can be used to condition the generative process by feeding y as input to G and D, respectively. Thus, the prior input z is combined with the additional information y and fed into G while the x input to D is also combined with y. The conditional version for the adversarial objective is as in Eq. 2;

$$\begin{aligned}\min_G \max_D f(D,G) = {} & \mathbb{E}_{x\sim P\text{data}(x)}\left[\log D(x|y)\right] \\ & + \mathbb{E}_{x\sim P(z)}\left[\log\left(1 - D(G(z|y))\right)\right]\end{aligned} \tag{2}$$

By conditioning the generative and discriminative models G and D, respectively, on additional information such as class labels or some data from other modalities, the data generation process can be guided.

Deep Convolutional Neural Networks (CNNs) have achieved state of the art results in many image recognition tasks; however, adapting CNN architectures to GANs context has not come without challenges [28]. Hence, Radford, Metz [29] proposed a class of architecturally constrained convolutional GANs called Deep Convolutional GANs (DCGAN) that are reported to be stable while training deeper GANs. Some of the important architectural modifications are as follows:

- Replacing pooling layers with strided convolutions in both generator and discriminator.
- Using Batch normalization in all layers of both generator and discriminator.
- Respectively using ReLU and LeakyReLU activation functions in all layers of the generator and discriminator except the output.
- Avoiding the use of fully connected layers over convolutional features.

A conditional version of DCGAN that we call CDCGAN is proposed for the handwritten Arabic character generation by including the class label of each character as the conditional information.

Several approaches have been taken in conditioning DCGAN. While the auxiliary information y has generally been fed as input to the conditioned generator alongside the input noise z, the point at which y is introduced to the conditioned discriminator has varied. Gauthier [30] has, however, reasonably argued that y be made to interact with the foundational information to the decision of both the generator and discriminator which is the dense code. It is on this basis that we choose to feed y to the discriminator alongside the dense code produced after convolution in our proposed CDCGAN discriminator.

4.3 CDCGAN Architecture

The model architecture, as illustrated in Fig. 1, comprises of two convolutional neural networks, Discriminator D and Generator G, that both learn through competition. Each network is described in what follows.

The CDCGAN discriminator D is a discriminative Deep CNN architecture comprising of a sequence of four convolutional layers and a fully connected layer (Fig. 2). Each convolutional layer has a batch normalization operation, dropout rate of 25% with Leaky rectified activation [31]. The high-level output of the series of convolutions is flattened, concatenated with the one-hot encoded conditional formation y and fed to a dense/fully connected layer before the output. Contrary to what is suggested in [29], we found this to be beneficial as it allowed room for more interaction between the high-level representations and the class labels y, resulting in higher-quality class conditioned outputs unlike when the concatenated features were fed directly to the output unit.

Rather than simply distinguish between real and fake samples, the CDCGAN is unique in that D is a multiclass classifier having a SoftMax output with $K + K'$ possible classes, where K represents the real 28 possible classes of the Arabic characters and K' corresponds to the 28 possible classes of the generated samples.

The model architecture for the CDCGAN generator G, as illustrated in Fig. 3, can be best described as an inverted$D$. The conditional information, y, corresponding to the class label of each character, is initially one-hot encoded into a 28-dimensional vector and concatenated with a 100 dimensional uniformly distributed noise input z before being fed as input to a fully connected layer. Given that the task for

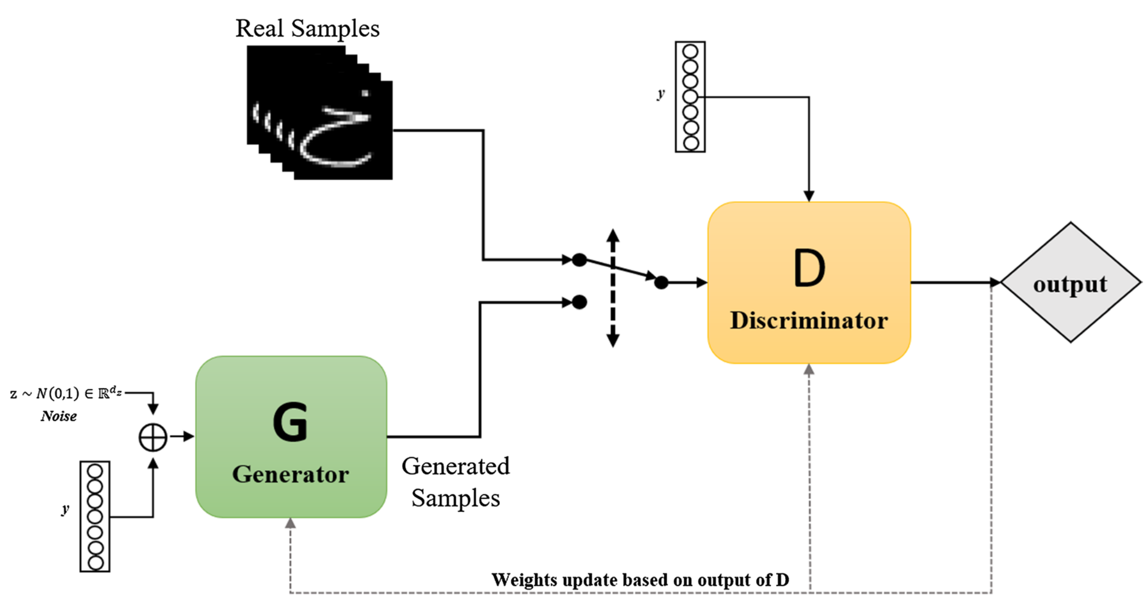

Fig. 1 CDCGAN Model Architecture

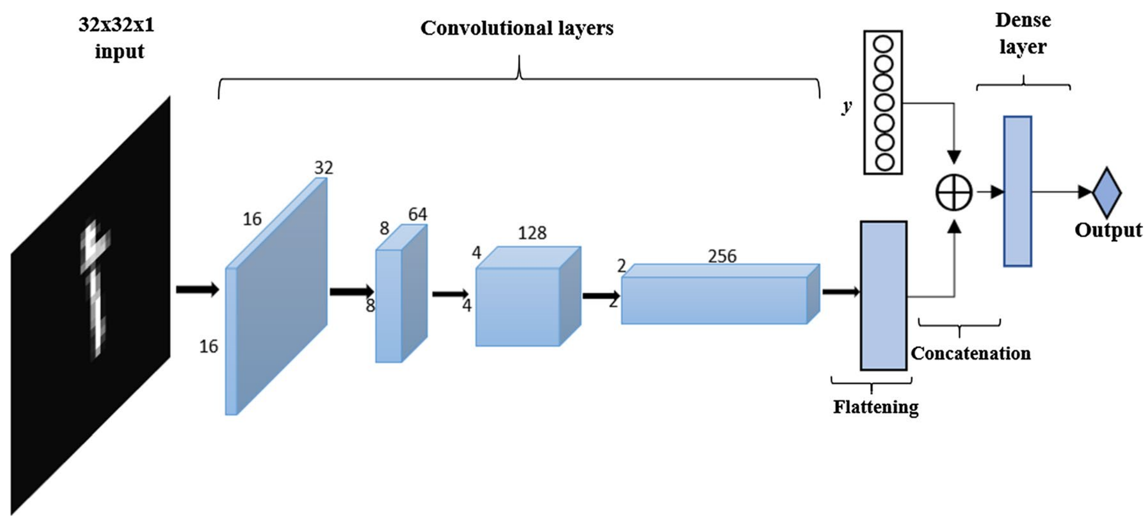

Fig. 2 CDCGAN Discriminator Network

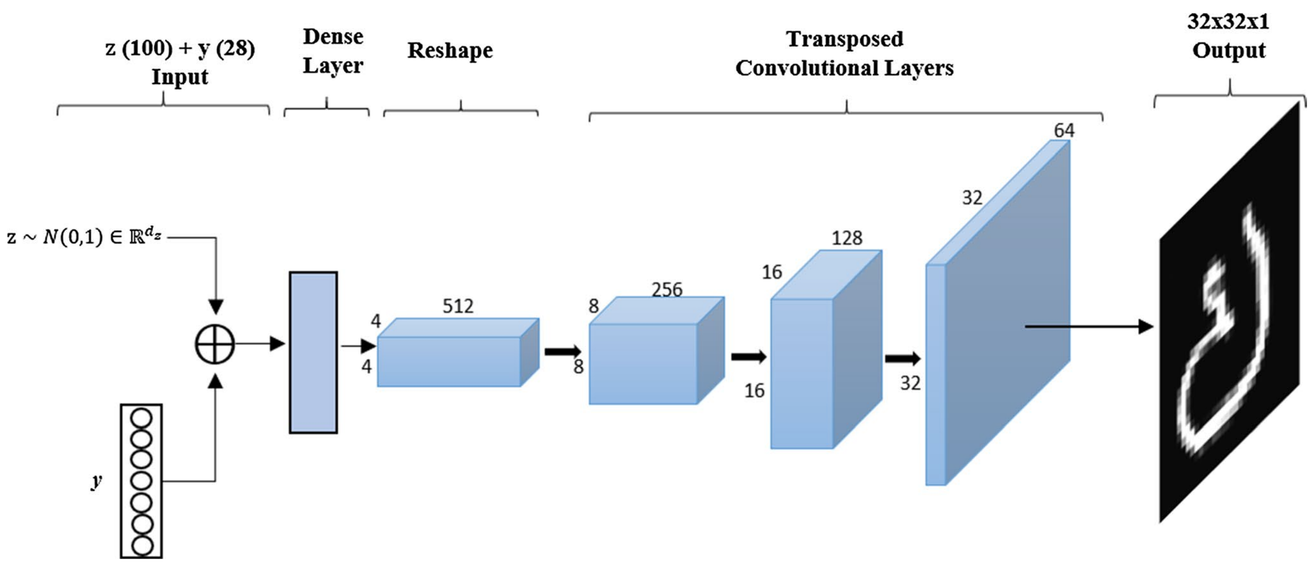

Fig. 3 CDCGAN Generator Network

G is a generative one with the aim of predicting each pixel values of an image, there is the need to increase the height and width of the input to the desired shape. Thus, the output of the first layer is reshaped and passed through three successive layers of transposed/fractionally strided convolutions [32] from which a single-channeled 32×32 image is predicted. Each transposed convolutional layer is batch normalized with ReLU activation.

$$\mathcal{L}_D^{CDCGAN} = -\mathbb{E}_{x \sim P\mathrm{data}(x)}\left[\log D(x|y)\right] - \mathbb{E}_{x \sim P(z)}\left[\log\left(1 - D(G(z|y))\right)\right] \tag{3}$$

$$\mathcal{L}_G^{\mathrm{CDCGAN}} = -\mathbb{E}_{x \sim P(z)}\left[\log D(G(z|y))\right] \tag{4}$$

The non-saturation version of GAN loss as in Eqs. 3 and 4 is used. Given the differentiable nature of the model, Adam optimizer [33] with a learning rate of 0.0002 and momentum of 0.5 was used in model training for both D and G as reported in [29]. In addition, after trying several values, a mini-batch Stochastic Gradient Decent (SGD) with batch size of 32 was used as this brought more stability to the model training. More details of the hyperparameters used for CDCGAN are presented in supplementary material.

4.4 Machine Learning Modeling

To evaluate the machine learning efficacy of CDCGAN-generated data, a Lenet-5 [34] deep learning with batch

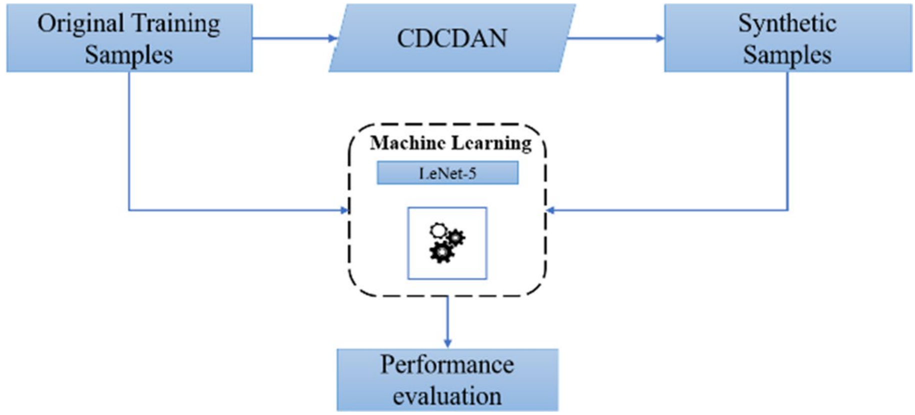

Fig. 4 Machine Learning Training and Evaluation of CDCGAN

normalization in all its layers and dropout regularization on the fully connected layers is trained for handwriting recognition task in line with the evaluation procedure shown in Fig. 4. Further details of the modified Lenet-5 model are presented in Table c of the supplementary material.

4.5 Dataset

The CDCGAN was trained on the AHCD database of 16,800 isolated handwritten Arabic characters collected by [1]. All 28 Arabic characters from "Alif" to "Yā'" (see Table 1) were written by 60 different participants, majority of whom are right-handed, ten times after which they were scanned and segmented. Although the dataset is divided into training and test sets of 13,440 and 3,360 images of characters, respectively, the combined dataset was used in the adversarial training. The dataset has 28 classes representing each distinct character. More details on the dataset can be found in [1].

In addition to Arabic handwriting character generation task, the performance of CDCGAN on other domain of handwritten characters is also explored to ascertain its effectiveness. Precisely, handwritten English letters from the EMNIST [35] letters benchmark were used. Only 26 classes of handwritten English capital letters from the "ByClass" split were selected. The selected split has a total of 220,304 images with the training and test sample sizes being 188,958 images and 31,346 images, respectively. A -1 to 1 scaling of all input images was carried out on each input data for CDCGAN training.

4.6 Experimental Environment

Implementation and evaluation of the CDCGAN were carried out in Keras 2.3.1 deep learning environment [36] with TensorFlow backend on a Dell Inspiron 7559 machine with Nvidia GeForce GTX 960 M graphic card.

5 Results

5.1 Qualitative Results

The qualitative results of the generated and real training samples for both isolated handwritten Arabic and English characters are presented here. For each class of the generated and training (real) samples, a sample is randomly generated (selected) for display as shown in Fig. 5. The generated samples were not cherry-picked, and as can be seen, they are as perceptually discriminable across classes as the training samples. In addition, the visual quality of the generated images for both datasets is comparable to their respective training samples, revealing how well the proposed CDCGAN learned the underlying data distribution and generated samples accordingly. However, the CDCGAN model learned the English letters faster and better compared to the Arabic characters. The reason for this is nor far-fetched as English characters are by their nature fully connected unlike Arabic characters with detached variable number dots above or below similar looking master strokes. Moreover, the CDCGAN was trained on a much larger samples of boldly written English characters than the Arabic characters.

5.2 Quantitative Results

While the qualitative results of the generated samples have been shown to be impressive and comparable to the training samples, it is not an all-encompassing indicator of the performance of the proposed CDCGAN, especially as it regards capturing the natural diversity and variability that comes with individual handwriting across different characters. One of the known problems with GANs is mode collapse [37], which occurs when the generator collapses to a specific state in which it masters a few samples which it generates every time to ultimately deceive the discriminator during training. The proposed CDCGAN will be of little use if it is only able to generate some set of samples per class of the Arabic

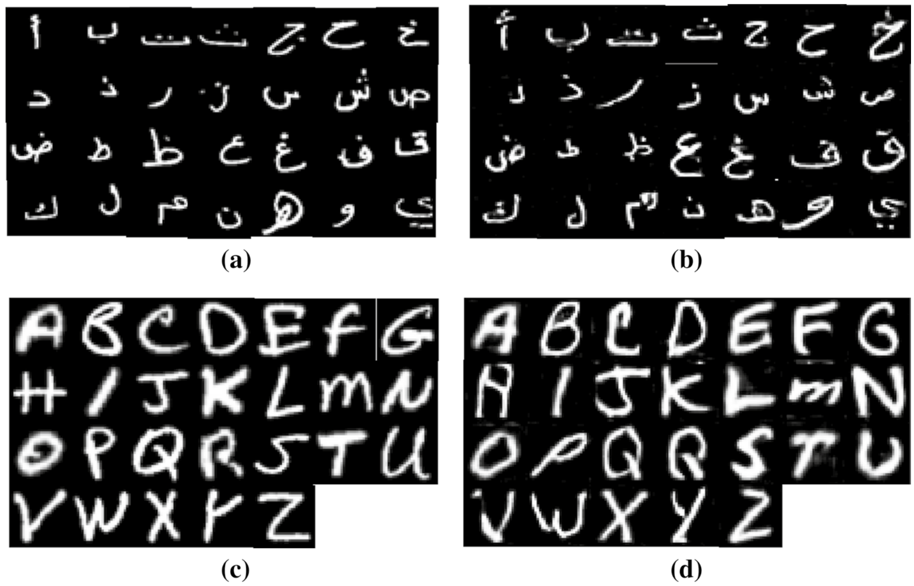

Fig. 5 **a** Real Handwritten Arabic Characters; **b** Generated Handwritten Arabic Characters; **c** Real Handwritten English Alphabets; **d** Generated Handwritten English Alphabets

handwritten characters. Hence, multiscale structural similarity (MS-SSIM) [38] metric is employed to measure the intra-class diversity of the generated samples with respect to the training ones.

MS-SSIM has been successfully applied in previous GANs research to measure the perceptual diversity of image samples [39, 40]. The score ranges from 0.0 to 1.0 with higher values indicating more structural similarity between a pair of images. In the context of this research, it means that images with high diversity should have lower MS-SSIM score and vice versa. For each of the training and generated samples, the MS-SSIM scores for 100 randomly selected pairs of images per class are calculated and their mean scores reported. Samples with lower MS-SSIM scores exhibit higher degree of diversity in comparison with those with lower MS-SSIM scores.

As shown in Fig. 6a, the mean MS-SSIM scores for the real and generated samples, respectively, indicated by blue- and red-colored bars exhibit high variability in diversity of the samples across the 28 classes of the Arabic handwritten characters. The least and most diverse classes in the real and generated samples, as modeled by the CDCGAN, are "dāl" (د) and "ghayn" (غ) with mean MS-SSIM scores of 0.75 and 0.51, respectively. Another character of comparably low diversity, similar to that of "ghayn" (غ), is "sha" (ش). The reason behind this is largely due to the high variability in the writing of these letters in the original training set. As it can be observed, the difference between the MS-SSIM scores of the original and generated samples for these two letters is minimal, implying that the CDCGAN model learns the data fairly well. However, the intrinsic variability in the writing of the master strokes coupled with the variation in number and position of the dots on these letters contributes to the low MS-SSIM scores. Generally, similar looking characters with higher number of dots (like "tāʾ," "thā," "shīn," "qāf") exhibit lower MS-SSIM scores than their single-dotted counterpart (like "bāʾ," "sīn," "fāʾ").

Similarly, Fig. 6b shows the bar charts of the mean MS-SSIM scores for the real and generated handwritten English letters. These characters exhibit higher diversity across the 26 alphabets given the low mean MS-SSIM scores (mostly below 0.5) across the samples, the least and highest being letters "Q" (0.358) and "I" (0.696), respectively. Besides, CDCGAN learns the English characters well as indicated by the marginal difference between the MS-SSIM scores of the real and generated data.

Overall, the generated data are modestly less structurally diverse than the training/real data as shown in Table 3 where the average MS-SSIM scores of all of the real and generated Arabic and English Handwritten characters are, respectively, displayed. The handwritten English characters show higher structural diversity than the Arabic ones. Nonetheless, the CDCGAN model learns the handwritten English characters better, given a difference of 0.0048 between the mean MS-SSIM scores of the generated and real data compared 0.0108 in case of handwritten Arabic characters.

5.3 Machine Learning Efficacy on Handwriting Recognition Task

A deep learning model based on the modified Lenet-5 described in Sect. 4.4 is, respectively, built for real, CDCGAN-generated and augmented data (augmentation of real training with CDCGAN-generated data). The average performance accuracy of the model on the real test data after ten repetitions of the experiment is presented in Table 4. No data augmentation is performed for the handwritten English letters since the available training data are enough to achieve

Fig. 6 Bar Chart MS-SSIM Scores for Real and Generated Samples of Handwritten **a** Arabic and **b** English Characters

Table 3 Mean and Standard Deviation of MS-SSIM Scores Across All Classes

	Mean	Standard deviation
Arabic		
Generated	0.6350	0.0533
Real	0.6142	0.0531
Latin		
Generated	0.4658	0.0684
Real	0.4610	0.0690

Table 4 Accuracy of LeNet5 on real, generated and augmented data

Model	Data	Real	Generated	Augmented
Modified Lenet-5	Arabic	94.99	84.77	95.08
	English	99.91	94.30	–

an insurmountably high performance on the test data. With performance gaps of 10.22% and 5.61% between generated and real handwritten Arabic and English letters, respectively, CDCGAN performs reasonably well compared to findings in similar works on Tabular data [6].

6 Discussion

Both the qualitative and quantitative results collectively reveal consistent patterns in the generated Arabic characters where the two most diverse characters (based on quantitative results), “ghayn” (غ) and “sha” (ش), are also visually distorted as shown in Figs. 5a and 7, revealing the higher difficulty in the placement of dots on these letters compared to undotted ones. Interestingly, a trend that can be observed in quantitative results (Fig. 6a) is that characters with dots

Fig. 7 (Left) Real and (left) Generated Handwritten Arabic Characters

have lower MS-SSIM scores compared to characters having the same master stroke but with lesser number of dots or none at all. This pattern is the same across both generated and real samples, revealing that characters with dots are more perceptually and structurally diverse than their undotted or less dotted counterpart. For instance, characters such as "khāʾ," "dhāl, "zayn," "shīn," "ḍād," "ẓāʾ," "ghayn" and "qāf" have lower MS-SSIM scores than "ḥāʾ," "dāl," "rāʾ," "sīn," "ṣād," "ṭāʾ," "ʿayn" and "fāʾ," respectively. Figure 7 shows a side-by-side comparison of the variation in the dot positions for some of dotted characters between the real and generated samples.

It was observed during the training of CDCGAN that unique characters (like "mim," "waw," "alif") were learnt earlier than dotted ones with similar master strokes (like "ba," "ta," "tha"), especially as it relates to correct placement of dots. Inconsistencies in the positions/placement of dots have been identified in pertinent handwritten Arabic character recognition studies as a reason for misclassification of some characters [1, 41]. Hence, in addition to the choice of CNN architecture for both the D and G networks of CDCGAN to take advantage of its translational invariance capability, D has also been uniquely designed as a 56-class multiclass classifier (with 28 real and 28 generated classes) to allow fine-grained identification of various classes of generated and real characters. Besides, other factors like illegibility of the dots and lack of clear separation of dots from master strokes still remain persistent in handwriting datasets. Nonetheless, CDCGAN produced impressive synthetic prototypes of the Arabic handwritten characters that can be used in training or augmenting training of recognizers for reasonably good performance. Thus, CDCGAN can be alternatively leveraged to alleviate the challenges related to limited training samples and bureaucratic hurdles in the way of data access.

On the other hand, despite the higher structural diversity shown by the English alphabets, the qualitative and machine learning results indicate CDCGAN performs better on the English handwriting compared the Arabic ones. Moreover, CDCGAN learnt to generate English alphabets in a shorter number of epochs. This suggests that Arabic handwriting recognition and synthesis are a more challenging task. Representative works on GANs have also indicated the nature of dataset as a major factor in model performance [42].

A common limitation of earlier GANs architectures that CDCGAN is not free from is the instability of training while synthesizing images of high perceptual quality [8, 37], especially for Arabic handwriting characters. However, using a non-saturating loss and a multiclass discriminator with batch size of 32 brought stability to the training. Other potential limitations of this study lie in the size and nature of the AHCD dataset. GAN-based studies on handwritten Chinese character style transfer have reported better performance with increasing training sample size [11, 19]. Besides, the reported performance in this study is only in relation to isolated handwritten characters. Arabic handwritten characters in their connected forms come with additional constraints that are not considered here. Therefore, applying the findings to Arabic words or characters in their connected forms should be done with caution as factors such as positional variation in the forms of each character within words have not been considered in this study.

7 Conclusion and Future Work

In this paper, a conditional deep generative adversarial network (CDCGAN) is proposed for isolated handwritten Arabic character generation. Beyond classical binary discrimination of real and fake samples, a fine-grained multiclass classification is employed where for every handwritten character, there is a corresponding class for real and generated samples. Experimental findings in terms of qualitative, quantitative and machine learning efficacy show that the generated samples are reasonably comparable to the real ones, both in diversity and quality. The proposed approach offers an easy alternative to the limited database of handwritten Arabic characters. An interesting future direction is the integration of multiple auxiliary classifiers to CDC-GAN to specifically learn the number and position of dots. Besides, interpolation of the proposed scheme for automatic generation and classification of concatenated Arabic sentences/phrases instead of isolated characters is another intriguing future direction.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s13369-021-05796-0.

Declarations

Conflict of Interest The author declares no conflict of interest.

References

1. El-Sawy, A.; Loey, M.; El-Bakry, H.: Arabic handwritten characters recognition using convolutional neural network. WSEAS Trans. Comput. Res. **5**, 11–19 (2017)
2. Ghadhban, H.Q., et al.: Survey of offline Arabic handwriting word recognition. In: International Conference on Soft Computing and Data Mining. Springer (2020)
3. Maalej, R.; Kherallah, M.: Convolutional neural network and BLSTM for offline Arabic handwriting recognition. In: 2018 International Arab conference on information technology (ACIT). IEEE. (2018)
4. Altwaijry, N.; Al-Turaiki, I.: Arabic handwriting recognition system using convolutional neural network. Neural Comput. Appl. (2020). https://doi.org/10.1007/s00521-020-05070-8
5. Mars, A.; Antoniadis, G.: Arabic online handwriting recognition using neural network. Int. J. Artif. Intell. Appl. **7**(5), 51–59 (2016)
6. Xu, L.: Veeramachaneni, K.: Synthesizing tabular data using generative adversarial networks. arXiv preprint http://arxiv.org/abs/1811.11264. (2018)
7. Che, Z., et al.: Boosting deep learning risk prediction with generative adversarial networks for electronic health records. In: 2017 IEEE international conference on data mining (ICDM). IEEE. (2017)
8. Goodfellow, I.J., et al.: Generative adversarial networks. arXiv preprint http://arxiv.org/abs/1406.2661. (2014)
9. Mirza, M.; Osindero, S.: Conditional generative adversarial nets. arXiv preprint http://arxiv.org/abs/1411.1784, (2014)
10. Karras, T.; Laine, S.; Aila, T.: A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. (2019)
11. Chang, B., et al.: Generating handwritten Chinese characters using cyclegan. In: 2018 IEEE winter conference on applications of computer vision (WACV). IEEE. (2018)
12. Alonso, E.; Moysset, B.; Messina, R.: Adversarial generation of handwritten text images conditioned on sequences. arXiv preprint http://arxiv.org/abs/1903.00277. (2019)
13. Elzobi, M., et al.: IESK-ArDB: a database for handwritten Arabic and an optimized topological segmentation approach. Int J Doc. Anal. Recognit. **16**(3), 295–308 (2013)
14. LeCun, Y., et al.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
15. Torki, M., et al.: Window-based descriptors for Arabic handwritten alphabet recognition: a comparative study on a novel dataset. arXiv preprint http://arxiv.org/abs/1411.3519. (2014)
16. Graves, A.: Generating sequences with recurrent neural networks. arXiv preprint http://arxiv.org/abs/1308.0850, (2013)
17. Kong, W.; Xu, B.: Handwritten Chinese character generation via conditional neural generative models. In: Proceedings of the 31st conference on neural information processing systems (NIPS 2017), Long Beach, CA, USA. (2017)
18. Zhu, J.-Y., et al.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE international conference on computer vision. (2017)
19. Chang, J., et al.: Chinese handwriting imitation with hierarchical generative adversarial network. BMVC. (2018)
20. Liu, J., et al.: Multi-scale multi-class conditional generative adversarial network for handwritten character generation. J. Supercomput. **75**(4), 1922–1940 (2019)
21. Ji, B.; Chen, T.: Generative adversarial network for handwritten text. arXiv preprint http://arxiv.org/abs/1907.11845, (2019)
22. Fogel, S., et al.: ScrabbleGAN: semi-supervised varying length handwritten text generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2020)
23. Davis, B., et al.: Text and style conditioned GAN for generation of offline handwriting lines. arXiv preprint http://arxiv.org/abs/2009.00678, (2020)
24. Kang, L., et al.: GANwriting: content-conditioned generation of styled handwritten word images. In: European conference on computer vision. Springer. (2020)
25. Gan, J.; Wang, W.: HiGAN: handwriting imitation conditioned on arbitrary-length texts and disentangled styles. (2021)
26. Haque, S., et al.: Onkogan: Bangla handwritten digit generation with deep convolutional generative adversarial networks. In International conference on recent trends in image processing and pattern recognition. Springer. (2018)
27. Nishat, Z.K.; Shopon, M.: Synthetic class specific Bangla handwritten character generation using conditional generative adversarial networks. In: 2019 International conference on Bangla speech and language processing (ICBSLP). IEEE. (2019)
28. Denton, E., et al.: Deep generative image models using a laplacian pyramid of adversarial networks. arXiv preprint http://arxiv.org/abs/1506.05751, (2015)
29. Radford, A.; Metz, L.; Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint http://arxiv.org/abs/1511.06434, (2015)
30. Gauthier, J.: Conditional generative adversarial nets for convolutional face generation: class Project for Stanford CS231N: convolutional neural networks for visual recognition. Winter Sem. **2014**(5), 2 (2014)
31. Maas, A.L.; Hannun, A.Y.; Ng, A.Y.: Rectifier nonlinearities improve neural network acoustic models. In: Proc. icml. Citeseer (2013)
32. Dumoulin, V.; Visin, F.: A guide to convolution arithmetic for deep learning. arXiv preprint http://arxiv.org/abs/1603.07285, (2016)
33. Da, K.,: A method for stochastic optimization. arXiv preprint http://arxiv.org/abs/1412.6980, (2014)
34. LeCun, Y.: LeNet-5, convolutional neural networks. URL: http://yann.lecuncom/exdb/lenet. **20**(5), 14 (2015)

35. Cohen, G., et al.: EMNIST: Extending MNIST to handwritten letters. In: 2017 International joint conference on neural networks (IJCNN). IEEE. (2017)
36. Chollet, F.: keras. (2015)
37. Salimans, T., et al.: Improved techniques for training gans. In: Advances in neural information processing systems. (2016)
38. Wang, Z.; Simoncelli, E.P.; Bovik, A.C.: Multiscale structural similarity for image quality assessment. In: The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers. (2003)
39. Odena, A.; Olah, C.; Shlens. J.: Conditional image synthesis with auxiliary classifier gans. In: Proceedings of the 34th international conference on machine learning-volume 70. (2017)
40. Borji, A.: Pros and cons of gan evaluation measures. Comput. Vis. Image Underst. **179**, 41–65 (2019)
41. Younis, K.S.: Arabic handwritten character recognition based on deep convolutional neural networks. Jordan J. Comput. Inf. Technol. **3**(3), 186–200 (2017)
42. Lucic, M., et al.: Are gans created equal? A large-scale study. arXiv preprint http://arxiv.org/abs/1711.10337, (2017)
43. Pechwitz, M., et al.: IFN/ENIT-database of handwritten Arabic words. In Proc. of CIFED. Citeseer (2002)
44. El-Sherif, E.A.; Abdelazeem, S.: A two-stage system for Arabic handwritten digit recognition tested on a new large database. Artif. Intell. Pattern Recognit. (2007)
45. Das, N., et al.: Handwritten Arabic numeral recognition using a multilayer perceptron. arXiv preprint http://arxiv.org/abs/1003.1891. (2010)
46. CMATERDB. Google code archieve—long-term storage for google code project hosting. [cited 2021 2/2]; Available from: https://code.google.com/archive/p/cmaterdb/downloads.
47. Lawgali, A.; Angelova, M.; Bouridane, A.: HACDB: Handwritten Arabic characters database for automatic character recognition. In: European workshop on visual information processing (EUVIP). IEEE. (2013)