IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 29 July 2022, accepted 9 August 2022, date of publication 25 August 2022, date of current version 31 August 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3201560

RESEARCH ARTICLE

DeblurGAN-CNN: Effective Image Denoising and Recognition for Noisy Handwritten Characters

SARAYUT GONWIRAT, AND OLARIK SURINTA
Multi-Agent Intelligent Simulation Laboratory (MISL), Department of Information Technology, Faculty of Informatics, Mahasarakham University, Mahasarakham 44150, Thailand

Corresponding author: Olarik Surinta (olarik.s@msu.ac.th)

This work was supported by the Royal Golden Jubilee Ph.D. Program by the Thailand Research Fund under Grant PHD/0210/2561.

ABSTRACT Many problems can reduce handwritten character recognition performance, such as image degradation, light conditions, low-resolution images, and even the quality of the capture devices. However, in this research, we have focused on the noise in the character images that could decrease the accuracy of handwritten character recognition. Many types of noise penalties influence the recognition performance, for example, low resolution, Gaussian noise, low contrast, and blur. First, this research proposes a method that learns from the noisy handwritten character images and synthesizes clean character images using the robust deblur generative adversarial network (DeblurGAN). Second, we combine the DeblurGAN architecture with a convolutional neural network (CNN), called DeblurGAN-CNN. Subsequently, two state-of-the-art CNN architectures are combined with DeblurGAN, namely DeblurGAN-DenseNet121 and DeblurGAN-MobileNetV2, to address many noise problems and enhance the recognition performance of the handwritten character images. Finally, the DeblurGAN-CNN could transform the noisy characters to the new clean characters and recognize clean characters simultaneously. We have evaluated and compared the experimental results of the proposed DeblurGAN-CNN architectures with the existing methods on four handwritten character datasets: n-THI-C68, n-MNIST, THI-C68, and THCC-67. For the n-THI-C68 dataset, the DeblurGAN-CNN achieved above 98% and outperformed the other existing methods. For the n-MNIST, the proposed DeblurGAN-CNN achieved an accuracy of 97.59% when the AWGN+Contrast noise method was applied to the handwritten digits. We have evaluated the DeblurGAN-CNN on the THCC-67 dataset. The result showed that the proposed DeblurGAN-CNN achieved an accuracy of 80.68%, which is significantly higher than the existing method, approximately 10%.

INDEX TERMS Handwritten character recognition, denoising image, generative adversarial network, DeblurGAN, convolutional neural network.

I. INTRODUCTION

Character recognition is a sub-process of text recognition systems used to recognize handwritten and printed texts within document images, such as historical documents, memoranda, and archival material. Therefore, when the main objective is to focus on the effects of handwritten character recognition, the factors that affect are as follows. 1) Writing styles; the distinctions of writing in each era, the diversity of individual writing styles, and even writing types of equipment [1], [2]. 2) Degradation of historical documents; this maybe due to a lack of expert staff and the humidity of a storage location. 3) Digital transformation; blurred and noisy document images were created when using low-quality equipment and taking the picture with a camera without adequate lighting. 4) Limitations of data; an insufficient and uncovered dataset of handwritten character images in the training process. These factors need to be considered when recognizing handwritten text images.

The factors mentioned above directly affect machine learning, leading to decreased recognition performance. In the case of noise when digitizing ancient documents, Su *et al.* [3] experimented with noise generation using the differential evolution method to determine the optimal position for

The associate editor coordinating the review of this manuscript and approving it for publication was Donato Impedovo.

digitization. Adding one pixel to the original image (called a one-pixel attack) logically is the trick that causes the convolutional neural network (CNN) models to be misrecognized. Their experiments showed that adding one pixel could harm the CNN model by increasing the recognition errors. Mei *et al.* [4] demonstrated that blurred images affect the recognition rate. Subsequently, the DeepDeblur algorithm was invented to transform blurred into sharp images before sending the sharp images for recognition. Also, the sharp images caused the model to increase its recognition performance.

Recently, CNN has replaced traditional machine learning [5] and is widely used in handwritten character recognition. Since the CNN method is an automatic algorithm that consists of feature extraction techniques and image recognition, it is currently used in character recognition in many languages, such as Latin, Arabic, Bangla, Korean, Chinese, and Thai [1], [2], [6], [7], resulting in increased character recognition efficiency. However, if the training images are low quality and noisy, they will significantly reduce recognition efficiency [3], [4].

Furthermore, deep learning techniques, including CNN, auto-encoder, and generative adversarial network (GAN), have also been proposed to improve image restoration and denoising. Dong *et al.* [8] proposed the image restoration technique using the CNN technique. The objective of their study was to transform the low-resolution images into high-resolution images. They proposed the super-resolution CNN method, which is a lightweight deep learning architecture that quickly restores and reconstructs quality images. Zhang *et al.* [9] presented feed-forward denoising CNNs, which integrate single residual learning into the CNN architecture for denoising images and to manipulate blind Gaussian noise without unknown noise levels. Further, Gondara [10] proposed a convolutional denoising autoencoder to denoise the signal from the medical images and Souibgui *et al.* [11] proposed an encoder-decoder architecture based on vision transformers, called DocEnTr, to enhance degraded document images.

The GAN architecture is widely used in many domains, especially for image restoration and deblur [12], [13], [14]. The GAN architecture is designed as a generator that is capable of learning from many images and recreating a new image. The adversarial loss function in the GAN architecture is used to create a robust model that aims to create high-quality images during regeneration. DeblurGAN [14] was first employed by using the learning process of the WGAN-GP [15] and used perceptual loss [16], allowing the model to deblur images in the form of blind motion blur that can be caused by camera movement during a photograph. Consequently, GAN is designed to solve the problems of document images, such as cleaning noisy backgrounds, deblurring text in the documents, and regeneration of damaged characters into the complete characters [17], [18], [19], [20].

A. CONTRIBUTION

This research presents the DeblurGAN-CNN architecture that aims to solve the recognition problems of noisy handwritten character images. The proposed DeblurGAN-CNN architecture improved the image quality and resulted in higher performance of handwritten character recognition on various handwritten character and noisy character datasets. The contributions of our research are the following.

1) This paper proposes a new standard noisy Thai handwritten character dataset, called the n-THI-C68 dataset, to challenge other researchers to reconstruct sharp and clean handwritten characters. The noisy handwritten character images were synthesized by adding five noisy methods: low resolution, low contrast, additive white Gaussian noise, motion blur, and mixed noise. The n-THI-C68 dataset includes 68 classes and contains 11,592 character images in the training set and 14,290 character images in the test set.
2) We propose the deblur generative adversarial networks (GANs) combined with the convolutional neural network (CNN) architectures, called the DeblurGAN-CNN architecture, to reconstruct high-quality handwritten characters from noisy handwritten characters and simultaneously enhance the accuracy of the handwritten character recognition systems. In the DeblurGAN-CNN architecture, DeblurGAN is proposed to learn from the noisy images and regenerate the new sharp and clean handwritten character images. Hence, the reconstructed handwritten character images are assigned to the CNN architecture for recognition.

B. PAPER OUTLINE

This paper is organized as follows. Section II presents handwritten character recognition, convolutional neural network, and generative adversarial network. The proposed DeblurGAN-CNN architectures are described in detail in section III. The handwritten character (THI-C68 and THCC-67) and noisy handwritten character (n-THI-C68 and n-MNIST) datasets that were used in the experiments are described in Section IV. Section V reports the experimental results. The performance of the proposed method is discussed in Section VII. Finally, conclusions and future work are addressed in section VI.

II. RELATED WORK

A. HANDWRITTEN CHARACTER RECOGNITION

In the last two decades, handwritten character recognition (HCR) has been well-studied and has become fundamental to research in image recognition. Many handwritten datasets have been collected from real-world data that aim to improve the quality of the characters and enhance the recognition performance. The most well-known dataset is the MNIST dataset [21], which collected many digits written on envelopes and has 70,000 handwritten digits in total. To recognize the digit images from the MNIST dataset,

LeCun *et al.* [21] proposed the first convolutional neural network that included five convolutional layers, called LeNet-5, to address problems of the MNIST dataset. Their method achieved an accuracy of 99.20%. Belongie *et al.* [22] proposed shape context to discover the correspondence points on the digit images and then match two shapes using the bipartite graph method. Hence, the minimum cost between the shape of the query image and training images was the best matching. As a result, an accuracy of 99.37% was achieved from their method. Surinta *et al*. [23] proposed the histograms of oriented gradients (HOG) and bag of visual words (BOW), called HOG-BOW, to first extract the local features from the sub-images. Second, local features were sent to the K-means clustering algorithm to construct the codebook and used as the BOW features. Finally, the L2-regularized support vector machine (L2-SVM) was proposed as the classifier. The HOG-BOW combined with L2-SVM achieved an accuracy of 99.43%. Maas *et al.* [24] proposed dual codebooks that were constructed from the features extracted using pixel intensity and HOG method, called dual-BOW. The dual-BOW method achieved an accuracy of 99.17%. Furthermore, Abdulhussain *et al.* [25] used orthogonal polynomials and moments to extract the gradient and smooth from the digit images. These features were sent to the SVM to classify the digit images of three datasets: Roman, Arabic, and Devanagari, achieving an accuracy of 100%, 99.32%, and 99.28%, respectively.

For the Thai character dataset, Surinta *et al.* [1] collected isolated Thai handwritten characters that contained 68 classes and consisted of consonants, vowels, tones, and special symbols. They also proposed two local descriptors: scale-invariant feature transform descriptor (siftD) and HOG, to extract the robust features from the Thai character images. The robust features were sent to classify using SVM and K-nearest neighbor (KNN) methods. The Thai handwritten character dataset was divided into training and test sets for evaluation. The best method was the siftD method which combined SVM with the radial basis function (RBF) kernel. The siftD+SVM method achieved 98.93% with 10-fold cross-validation and 94.34% on the test set. Furthermore, Inkeaw *et al.* [26] proposed the gradient features of discriminative regions (HOGfoDRs) and SVM to recognize Thai characters. The HOGfoDRs+SVM method achieved 98.76% with 5-fold cross-validation. For the updated Thai handwritten dataset, Onuean *et al.* [27] collected Thai handwritten characters, called Burapha-TH, that consisted of 10 digits, 68 characters, and 320 syllable classes. They also created a CNN model using a VGG architecture with a batch normalization layer containing 13 layers, called VGG-13, evaluated on the Burapha-TH dataset, and which achieved 92.29%, 95.00%, and 96.16% accuracy on the digit, character, and syllable classes, respectively.

In this section, we focused on various approaches which used the traditional methods, including feature extraction methods and machine learning techniques for handwritten character recognition. For the feature extraction method, many state-of-the-art methods were investigated, such as siftD, HOG, HOG-BOW, dual-BOW, HOGfoDRs, orthogonal polynomials, moments, and shape context. Some state-of-the-art methods, including siftD and HOG, focus on extracting the feature from the invariant key points when the image is resized and rotated. Other methods, such as HOG-BOW and dual-BOW, cluster the robust feature that is extracted by the feature extraction methods into a codebook using clustering algorithms. Then, encoding the codebook from the input images and using them as robust features. For the machine learning techniques, two techniques: SVM and KNN, are proposed to create a robust model using the robust features. We have seen that simple machine learning techniques, such as the KNN algorithm, could obtain a high recognition rate when the robust features are extracted. However, complex computation processes are required when extracting the robust features.

B. GENERATIVE ADVERSARIAL NETWORK

The generative adversarial network (GAN) was first presented by Goodfellow *et al.* [12]. GAN is an unsupervised learning model that automatically learns from the regularities of input images and is then capable of creating a new image that is similar to the original image. Therefore, GAN has been applied in a wide range of applications, such as natural transfer style, image super-resolution, face generation, image restoration, and even image deblurring [14], [28], [29], [30].

Since GANs have generative ability and style transformation, they were applied in the data augmentation technique [7], [29] to improve recognition performance for document images. Fogel *et al.* [31] proposed ScrabbleGAN, which is semi-supervised learning by using unlabeled and labeled samples during the training process, to synthesize different Latin and French handwritten text styles. In addition, Eltay *et al.* [7] proposed adaptive data augmentation based on the ScrabbleGAN architecture to recognize Arabic handwritten text. The adaptive method generated more balanced characters in training samples.

Moreover, many issues in documents, such as blurred image, noisy background, salt-and-pepper, and faded text, lead to the document being unreadable to humans, significantly decreasing the recognition performance of the text algorithms [17], [18], [19], [20]. To solve these problems, Bhunia *et al.* [18] proposed two networks, including texture augmentation and binarization networks, to binarize the degraded document images. First, the texture augmentation network was designed to create multiple textual contents with diverse noisy textures to increase the size of the document binarization dataset. Second, the binarization network generated new images, which are the clean binary document images. Sharma *et al.* [17] used CycleGAN to remove the noise from the documents resulting in cleaned documents. The CycleGAN model was employed to map noise to clean documents and clean to noisy documents using the cycle consistency loss function. Their experiment showed that the CycleGAN provided acceptable results. In terms of document

enhancement, Souibgui and Kessentini [20] applied conditional GAN, which is a single GAN network, to restore various problems of mixed document degradations, including tasks of document clean up, binarization, deblurring, and watermark removal.

Furthermore, Wu *et al.* [32] applied Wasserstein loss to the CycleGAN that improved the CycleGAN algorithm to deblur text images into clear text images. Also, Zhao *et al.* [33] used the GAN model to optimize the distortion of input images before feeding the rectified images to the text recognizer.

Since the first GAN architecture was presented in 2014 [12] to regenerate a new image similar to the original image, many GAN architectures have been proposed to solve the problems, for example, noisy images, degraded documents, and blur text, in the domain of document images. We then have the concept of using the GAN architecture to denoise the handwritten character images before recognizing them using the CNN architecture.

C. CONVOLUTIONAL NEURAL NETWORK

The convolutional neural network (CNN) architecture method was first proposed in 1998 by LeCun *et al*. [21] to recognize handwritten digit images. In 2012, CNN began to gain attention and significant influence on image recognition research when Krizhevsky *et al.* [34] proposed a new CNN architecture that contains eight weighted layers, five convolutional layers and three fully-connected layers, to train on one million images from the ImageNet dataset with 1,000 classes [35] and win the LSVRC competition. In 2014, Simonyan and Zisserman [36] presented very deep CNN architecture to the depth of 16-19 weight layers, called VGGNets. These CNN architectures are called plain networks.

Consequently, we have seen that the design of the CNN architectures has very deep architecture, such as GoogLeNet which had 22 layers and ResNet which had more than 100 layers. However, when designed with deep weight layers, the weight parameters also decrease and require high computation. However, the new architectures were also proposed with new convolution techniques. For example, GoogLeNet proposed an inception architecture [36] which calculated with the small filter size of 1×1, 3×3, and 5×5. The dimension reduction technique was employed in the inception modules to reduce the weight parameters and then, the inception module was stacked on top of each module. A Residual connection [37] and InceptionResNet [38] architectures were proposed according to the time-consuming in the plain network like VGGNets. The residual connections were added to the plain network, which can be operated when input and output are the exact sizes. ResNets also has only one fully-connected layer with 1,000 nodes, While the VGGNets have three fully-connected layers with 4,096, 4,096, and 1,000 nodes, respectively.

Furthermore, the concept of connecting the current layer to other layers in a feed-forward direction was proposed and called DenseNet [39]. In the DenseNet architecture, the weight layers were then reused entirely in the network to make the model more compact. The DenseNet architecture could define the network with more than 200 weight layers. For MobileNetV1 [40], the lightweight architecture was proposed by using depthwise separable convolution, which is the operation of depthwise and pointwise convolutions proposed to reduce the dimension of the feature map. Subsequently, the inverted residual modules were proposed in MobileNetV2 [41]. The concept of the automatic discovery of the CNN architecture using reinforcement learning and recurrent neural networks (RNN) was invented and was called neural architecture search (NASNet) [42], which is a scalable architecture. Many convolution operations were selected using the controller RNN and recursively constructed convolutional cell blocks.

For the use of CNN in digit handwritten character recognition, Cireşan *et al.* [43] proposed multi-column deep neural networks (MCDNNs) in which the input image was first trained by different DNN blocks. Hence, the outputs of each DNN block were classified by averaging individual predictions. The MCDNN yields high performance with 99.77% accuracy on the MNIST dataset. Furthermore, Savita *et al.* [44] discovered the best hyperparameters of the CNN architecture, including the number of layers, kernel size, padding, stride, and receptive field. They also trained the CNN architecture with various optimization algorithms (SGDM, Adam, Adagrad, and Adadelta). The results showed that the CNN architecture with the Adam optimizer achieved an accuracy of 99.89% on the MNIST dataset.

Tang *et al.* [45] proposed two CNN architectures that included 6 layers (4 convolutional layers and 2 fully-connected layers) and 8 layers (5 convolutional layers and 3 fully-connected layers). The first CNN architecture was trained on printed Chinese characters. Hence, the pre-trained model of the first CNN architecture was used as a transfer learning to the second CNN architecture. The second CNN model was trained on historical Chinese characters. The accuracy was increased from 79.2% to 88.56% when using the transfer learning technique. Alom *et al.* [2] used various state-of-the-art CNN architectures: VGGNet, Network in Network, ResNet, and DenseNet, to recognize handwritten Bangla characters. The result showed that the DenseNet achieved the best accuracy with 98.31%. Gonwirat and Surinta [6] used the pre-trained model of the VGGNet instead of training from scratch. The result showed that the transfer learning of the VGGNet achieved 99.20% on the Thai handwritten character dataset.

Although the CNN architectures achieved high efficiency on image classification problems, Su *et al.* [3] demonstrated an image generation technique that only added one pixel into the target image based on the differential evolution technique. With only one attack pixel, the accuracy performance significantly decreased. For handwritten character recognition, many noisy methods were applied to the character images, such as motion blur, low contrast, and additive Gaussian white noise (AGWN) [46], to demonstrate

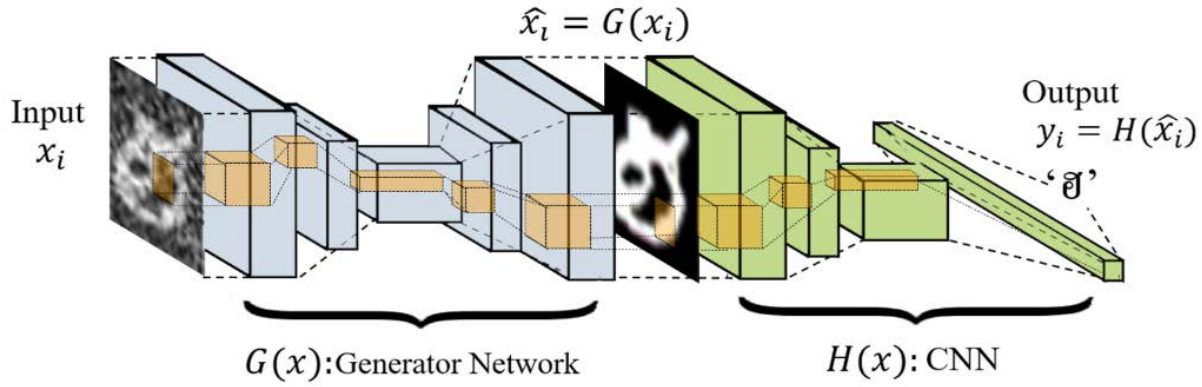

FIGURE 1. Illustration of the DeblurGAN-CNN architecture.

that the noise images could significantly reduce the recognition performance. Consequently, to increase the recognition efficiency, a synthesized image technique was introduced to remove noise before sending images to recognition.

CNN architectures have been proposed for image classification purposes. The CNN architectures combine two main tasks (feature extraction and machine learning) into one architecture to specifically reduce the complex feature extraction processes. Many state-of-the-art CNN architectures have been proposed and have become successful in many domains. For example, AlexNet, GoogLeNet, VGGNets, MobileNets, ResNet, DenseNet, NASNet, and EfficientNet. However, the latest CNN architectures operate with more deep layers, convolution operations (i.e., 1D, 2D, 3D convolution [47], [48], and depthwise separable convolution), and extra layers (i.e., global average pooling, inception module, reduction cell) [42] to compute the robust spatial features from the image. Therefore, the researcher could propose new CNN architecture, invent new operations, and combine them with the existing CNN architectures.

From related work above, we found that the GAN architecture could be used to solve the problems of noisy images, while various CNN architectures could propose to recognize the noisy handwritten character images. The proposed denoising and recognition framework is described in-depth in the following section.

III. THE PROPOSED DENOISING AND RECOGNITION FRAMEWORK

The performance of the handwritten character recognition is always affected by noise. Consequently, we proposed the DeblurGAN-CNN architecture to address the noise problems. Although, many robust CNN architectures achieved high accuracy in every domain, even on handwritten character images. However, the accuracy decreases when affected by many types of noise, such as blur, low resolution, and low contrast. In this research, we first studied the effect of the noisy character images that harm the performance of handwritten character recognition. Second, data augmentation techniques were applied while training the CNN model to increase new patterns of the handwritten character images. The data augmentation methods could generalize the CNN model when the noise was not adequately high. Hence, the performance decreased after adding a high noise level. Third, we discovered that the DeblurGAN could transform the noise into new clean handwritten characters. Finally, DeblurGAN architecture and the robust CNN architecture were combined to enhance the recognition performance of the handwritten character images, called DeblurGAN-CNN.

There are several methods for improving image quality, for example, super-resolution, image restoration, and deblurring images. However, some noise appears in the handwritten character images while transforming the document papers into digital format. Consequently, we considered two GAN architectures (DeblurGAN and CycleGAN) to address our problems because these two GAN architectures are designed for deblurring images. However, the CycleGAN is mainly used for a style transfer that transforms from one style to another style. In comparison, many forms of noise occur in handwritten character images, which means CycleGAN is not appropriate for these problems. Furthermore, we used the DeblurGAN architecture that could deal with many-to-one style transfer.

In this paper, we proposed the DeblurGAN-CNN framework that combines two state-of-the-art deep learning architectures to denoise and recognizes the noisy handwritten characters into one architecture. The proposed framework contains a generator of generative adversarial network (GAN) and convolutional neural network (CNN) architectures, as shown in Figure 1.

In the following subsections, the details of the DeblurGAN-CNN framework are described. 1) DeblurGAN is employed as a denoising network. 2) DenseNet121 is the convolutional neural network architecture performed as a recognition

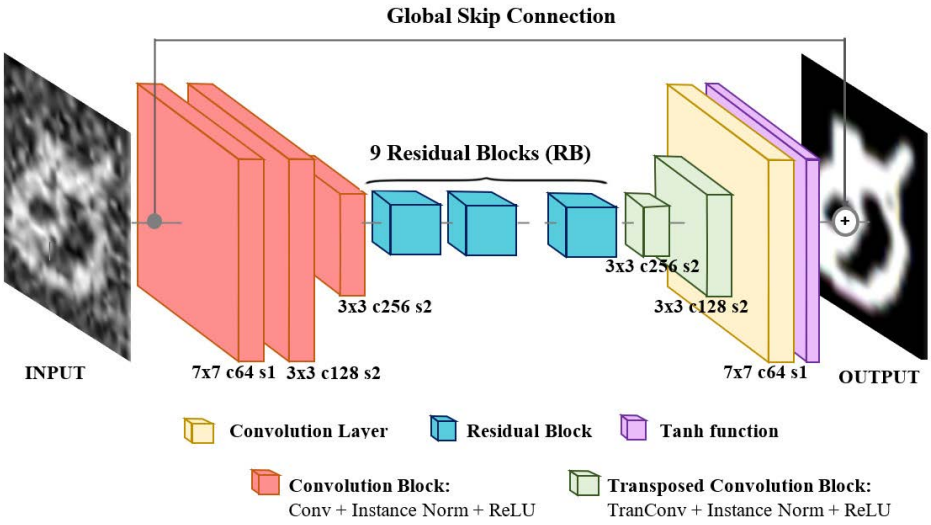

FIGURE 2. Illustration of the DeblurGAN generator architecture.

network. 3) We describe the DeblurGAN-CNN architecture and training strategy that is used for training the proposed framework.

A. DEBLURGAN

Kupyn *et al.* [14] proposed the GAN architecture to automatically deblur blurred images from any unknown blur function, called DeblurGAN, which can synthesize sharp images (I_S) from blurred images (I_B). The DeblurGAN uses the generator (G_{θ_G}) and the discriminator (D_{θ_D}) to distinguish between real and generated images.

The generator architecture of the DeblurGAN is shown in Figure 2. The beginning part of the network consists of three convolutional blocks that are designed to downsample the feature maps. The middle of the network is a sequence of nine residual blocks. In the last part of the network, the transposed convolution blocks are constructed to upsample feature maps to the original size as an input image. Moreover, the global skip connection is also proposed for this architecture by adding input to the output image. The global skip connection makes the network converge faster and yields better output results.

In the DeblurGAN, the PatchGAN architecture [13] is used as the discriminator. The PatchGAN architecture has downscale convolutional layers followed by instance normalization and leaky rectified linear unit (LeakyReLU) with $\alpha = 0.2$.

Consequently, as shown in Equation (1), the loss function is presented in the DeblurGAN that includes adversarial ($\mathcal{L}_{GAN}$) and content loss ($\mathcal{L}_X$) that is weighted by λ, where λ is a parameter that controls the relative of two objectives: adversarial and content loss. The WGAN-GP [15], which is the critic function to determine the completeness of the generator result, is used as the adversarial loss, as shown in Equation (2). Also, the content loss is the perceptual loss [16] to compare the style-transfer, called reconstructed image, with the original image using the L2 loss function.

$$\mathcal{L} = \underbrace{\underbrace{\mathcal{L}_{GAN}}_{adversarial\ loss} + \underbrace{\lambda \mathcal{L}_X}_{content\ loss}}_{total\ loss} \tag{1}$$

$$\underset{GAN}{\mathcal{L}} = \sum_{n=1}^{N} -D_{\theta_D}\left(G_{\theta_G}\left(I^B\right)\right) \tag{2}$$

B. DENSENET

In the early architecture, a residual connection using element-wise input (x) with an output building block ($F(x)$) [37] was proposed, called ResNet. The benefit of the ResNet architecture was that the network could construct with deep convolutional layers and still obtain better results in terms of speed and performance. However, the DenseNet architecture [39] was designed to include the maximum information flow by concatenating all feature maps (x_n^p) from the previous convolutional layers, called a dense block. DenseNet was proposed to deal with the reuse of the features, reduce the architecture parameters, and eliminate gradient problems. The equation of the DenseNet is shown in Equation (3).

$$\boldsymbol{x_n^p} = \boldsymbol{H^p}\left(\left[\boldsymbol{x_n^0}, \boldsymbol{x_n^1}, \boldsymbol{x_n^2}, \ldots, \boldsymbol{x_n^{p-1}}\right]\right) \tag{3}$$

where $\boldsymbol{H^p}(\cdot)$ is the composite function of a layer (p), including batch normalization (BN), rectified linear unit (ReLU), and convolutional (Conv) layer. The function parameter $\left[x_n^0, x_n^1, x_n^2, \ldots, x_n^{p-1}\right]$ is a concatenation of previous layers from the first layer (x_n^0) to last layer (x_n^{p-1}).

An overview of the DenseNet is shown in Figure 3(a). The DenseNet architecture consists of three main parts.: 1) A convolutional layer with a kernel size of 7×7. The convolutional block includes BN, ReLU, and Conv layers, with a stride of 2 and followed by a 3×3 max pooling layer

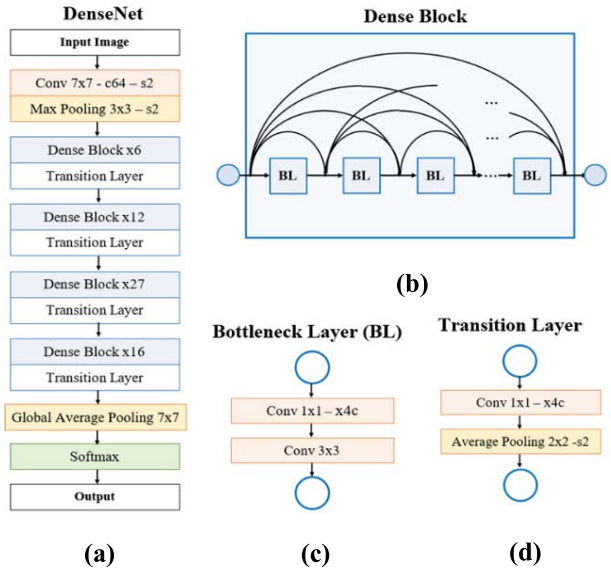

FIGURE 3. Illustration of the DenseNet121 architecture, including (a) core block, (b) dense block, (c) bottleneck layer, and (d) transition layer.

with a stride of 2. 2) Four dense blocks and transition layers. 3) The global average pooling (GAP) and classification layers with a softmax function.

Details of the DenseNet architecture, are as shown in Figure 3(b). The dense block is concatenated with the output of bottleneck layers, which is expanded N times, proposed to decrease the parameters of the architecture. Each bottleneck layer consists of 1×1 Conv and 3×3 Conv layers, as shown in Figure 3(c). The transition layer (see Figure 3(d)) is proposed to reduce the feature map width and height by 2×2 average pooling with a stride of 2 and θ parameter applies to compress the network where a range of a parameter is $0 < \theta \leq 1$.

In this paper, we proposed to use DenseNet121 since it is the smallest size appropriate for handwritten character recognition.

C. DEBLURGAN-CNN SETTING AND TRAINING SCHEME

In this section, we provide the construction and training strategy of the proposed framework, as shown in Algorithm 1. Also, the details of the setting and training strategy of the DeblurGAN-CNN framework are described in the following.

1) DEBLURGAN TRANING

DeblurGAN was designed for deblurring images. However, in our problems, DeblurGAN was applied to reconstruct the sharp handwritten character images from the various noisy styles, such as low contrast, motion blur, and white Gaussian noise. To train the DeblurGAN architecture, the dataset then includes the pairs of noisy and sharp handwritten character images, $(x_i^{noisy}, x_i^{sharp})$, where $i = 1, 2, \ldots, n$. In the DeblurGAN training process, the generator network receives a noisy image (x_i^{noisy}) as input and adjusts the weights to reconstruct the output as a sharp image (x_i^{sharp}). We evaluate the quality of the reconstructed handwritten character images using the discriminator and the loss function as shown in Equation (1).

Algorithm 1 Construction and training of the DeblurGAN-CNN framework

Input: Training set including pair of sharp and noisy character images $(x_i^{sharp}, x_i^{noisy})$ and label y_i, where $i = 1, 2, \ldots, n$, training epochs of DeblurGAN: M, training epochs of CNN: P.

Define: (X^D, Y^D) is training set of data augmentation technique $\{(x_i^{sharp}, y_i)\} \cup \{(x_i^{noisy}, y_i)\}$.

Step 1) Create DeblurGAN network including generator network $G(x)$ and discriminator $D(x)$.

Step 2) Train DeblurGAN using M epochs with dataset of pair set $\{(x_i^{sharp}, x_i^{noisy})\}$ and save the best model based on the loss function in Equation (1).

Step 3) Create CNN of pretrained weight from the ImageNet dataset.

Step 4) Train CNN using P epochs with the dataset (X^D, Y^D) and save the best model based on the loss function in Equation (4).

Step 5) Construct a DeblurGAN-CNN network as the following:
- Load the $G(x)$ network in the step 2).
- Load the CNN network in the step 4).
- Combine $G(x)$ and CNN with the intermediate layer.

Step 6) Fine tune the DeblurGAN-CNN network training using P epochs with the dataset(X^D, Y^D) and the loss function in Equation (4). The training steps consist of two steps as the following:
-Freeze the part of $G(x)$ in the network and train using $P/2$ epochs.
-Unfreeze and train all layers in the network using $P/2$ epochs.

Output: The DeblurGAN-CNN network

2) CNN TRANING

We employed the CNN architectures to train on a handwritten character dataset that consisted of the pairs (x_i, y_i), where $i = 1, 2, \ldots, n$, x_i is handwritten character of character i and y_i is label of character i. To improve the efficiency performance, we proposed the transfer learning method [6] with convolutional kernels of prior knowledge for faster convergence in a few epochs. The pre-trained CNN model was modified in the classification layer and then fine-tuned in the network. Furthermore, we trained the CNN models with the data augmentation techniques with noisy handwritten character images (x_i^{noisy}), which is synthesized from the original sharp images (x_i^{sharp}), where $x_i^{noisy} = f^{noisy}(x_i^{sharp})$ and $f^{noisy}(x)$ is the generator function of a synthesized noisy image. We trained the CNN model to classify images using categorical cross-entropy loss function as shown in Equation (4).

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^{N} \boldsymbol{log}\left(\boldsymbol{p_{CNN}}\left(\boldsymbol{x_i}; \boldsymbol{\theta}\right)\right) \tag{4}$$

where N is the number of training images and $\boldsymbol{p_{CNN}}(x_i; \theta)$ is the probability distribution of CNN output, where x_i is an input image and θ is weight parameters.

3) DEBLURGAN-CNN CONSTRUCTION

The DeblurGAN-CNN network connects the DeblurGAN generator and CNN, as shown in Figure 1. This proposed network benefits from the generator producing a sharp output image from various noisy images before recognizing it by the CNN model. The output of the generator ($G(x)$) is called the intermediate output ($\hat{x}_i$), where $\hat{x}_i = G(x_i)$ is computed using the tanh function. Hence, the output image values are in the range of -1 and 1. Subsequently, we add the intermediate layer to convert the value to the range of 0 and 1, the same as the input of the CNN which the adjusting function is $f(x) = (x + 1)/2$ where x is intermediate output.

4) DEBLURGAN-CNN FINE-TUNING

The DeblurGAN-CNN network is still an incomplete merge network since a part of CNN has inexperienced generator output. Thus, fine-tuning the DeblurGAN-CNN network is an approach to improvement. In the first step, we only trained the CNN by freezing the DeblurGAN generator for stable network training and retraining the output as sharp images. In the second step, we trained the DeblurGAN-CNN network with unfrozen whole layers. The proposed DeblurGAN-CNN network was trained with a few training epochs. We trained only ten epochs in each frozen step and each unfrozen step.

IV. HANDWRITTEN CHARACTER DATASETS

In this section, we briefly describe the handwritten character datasets used in the experiments, including two Thai handwritten character datasets: THCC-67 [49] and THI-C68 [1], and two noisy handwritten character datasets: n-MNIST [46] and n-THI-C68. An overview of the handwritten character datasets is shown in Table 1.

A. THE NECTEC THAI HANDWRITTEN CHARACTER CORPUS (THCC-67)

The National Electronic and Computer Technology Center (NECTEC) presented a Thai handwritten character corpus (THCC) of consonants, vowels, and tones that contains 67 classes, called THCC-67. The THCC-67 dataset has 9,012 characters that were rescaled to 32 × 32 pixels. In this research, we used it as an independent test. The THCC-67 dataset is shown in Figure 4(a).

TABLE 1. Overview of the handwritten character datasets.

Datasets	Types and Languages	Number of Classes	Training Sets	Test Sets
THCC-67 [49]	Char, Thai	67	-	9,012
THI-68 [1]	Char, Thai	68	11,592	2,898
n-THI-68	Char, Thai	68	11,592	14,290
n-NMINST [46]	Digit, Arabic	10	180,000	30,000

B. THE ALICE OFFLINE THAI HANDWRITTEN CHARACTER DATASET (THI-68)

The THI-C68 dataset containing 28 classes was proposed by Surinta *et al.* [1]. The THI-C68 dataset was collected from 150 university students aged 20-23 years old. Students wrote the Thai characters on a form with a white background that was scanned with a resolution of 200 dpi. Image transformation was used to rescale the aspect ratio to avoid distortion and images were stored in grayscale format. The THI-C68 dataset has 14,490 character images containing consonants, vowels, and tones. An example of the THI-C68 is shown in Figure 4(b).

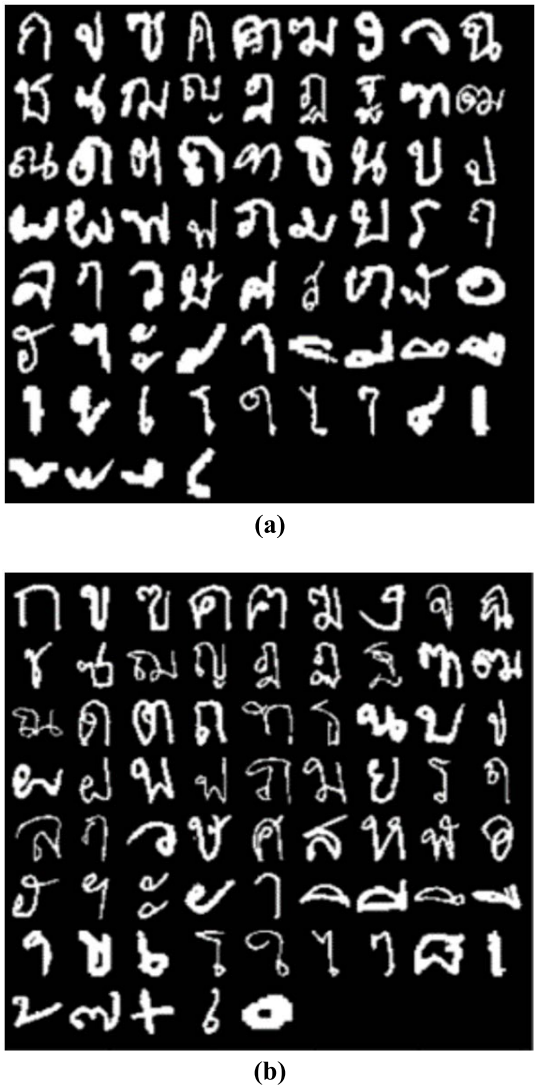

FIGURE 4. Examples of Thai handwritten character datasets: (a) THCC-67 and (b) THI-C68.

C. NOISY THI-C68 (N-THI-68)

In this research, we propose a new noisy Thai handwritten character dataset, called noisy THI-C68 (n-THI-C68). We synthesized new noisy character images using five different noisy techniques: low resolution, additive white Gaussian noise (AWGN), low contrast, motion blur, and mixed noise.

We randomly selected one noisy technique to synthesize each character image according to the THI-C68 dataset with

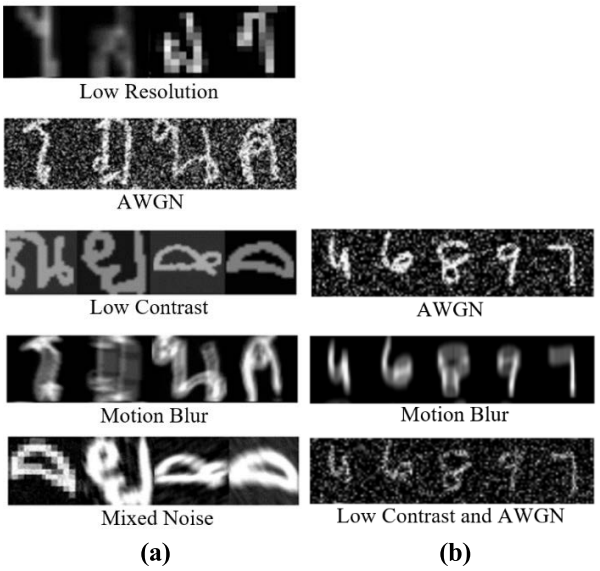

FIGURE 5. Examples of noisy handwritten character datasets: (a) n-THI-C68 that applied 1) low resolution, 2) AWGN, 3) low contrast, 4) motion blur, and 5) mixed noise and (b) n-MNIST that applied 1) AWGN, 2) Motion blur, and 3) low contrast and AWGN.

11,592 training images and 2,898 test images. We obtained 11,592 noisy character images for the training set that were randomly applied with noisy techniques with various adjustment values. For the test set, we increased the size from 2,898 character images up to 14,290 noisy character images by randomly applying five noisy techniques to the original character images.

As shown in Figure 5(a), noisy Thai handwritten character images were synthesized as follows. 1) Low resolution with a low level at 8-12 pixels. 2) AWGN with increasing noise with a peak signal to noise ratio (PSNR) of 9.5. 3) Low contrast with reduced color gradient in range of 0.15-0.5 based on the original images. 4) Motion blur with two blur methods: directional motion blur [46], [50] and random motion blur [51]. 5) Mixed noise between four noisy methods.

D. NOISY MNIST (N-MNIST)

Basu *et al.* [50] proposed the noisy MNIST (n-MNIST), which is the extended version of the MNIST dataset [21] that applied three noisy methods: AWGN, motion blur, and combinations between reduced contrast and AWGN. The n-MNIST dataset contains 10 classes (0-9) and has 180,000 training samples and 30,000 test samples due to applying three noisy techniques to the original images.

Figure 5(b) shows noisy digits were applied as follows. 1) AWGN using increase noise with RSNR of 9.5. 2) Motion blur using linear motion filter with a size of 5 pixels and rotation with 15 degrees, and a combination between reduced contrast and AWGN with a PSNR of 12.

V. EXPERIMENT RESULTS

In this section, we evaluated the performance of the proposed DeblurGAN-CNN architecture on the handwritten character datasets and noisy handwritten character datasets. We then investigated the effective recognition of CNNs and the quality of image restoration by the generative adversarial networks (GAN). In this study, we trained the CNN and GAN models on Linux operating systems with Nvidia GeForce GTX1080ti 8G GPU, Intel(R) Core i5-7400 Processor 3.00GHz CPU, 32GB DDR4 RAM.

A. EVALUATION OF THE CNN ARCHITECTURES ON THI-C68 DATASET

1) COMPARISON OF STATE-OF-THE-ART CNNS

We evaluated four CNN architectures: VGG19, Inception-ResNet, MobileNetV2, and DenseNet121 on the Thai handwritten character dataset to find the best CNN architecture. We divided the THI-C68 dataset into a training set and test set with 80% and 20% ratios, with 13,041 training images and 1,449 test images. Hence, the training set was evaluated using 5-fold cross-validation. The test set was an independent holdout set for final evaluation.

Furthermore, we focused on three training methods: 1) scratch learning (SL), 2) transfer learning (TL), and transfer learning with noisy data augmentation techniques (TL-nDA).

TABLE 2. Recognition performances (mean validation accuracy: 5-cv, standard deviation, and test accuracy) of four CNN models: VGG19, InceptionResNet, MobileNetV2, and DenseNet121, using different learning methods (SL, TL, and TL-nDA) on the THI-C68 dataset.

CNN Models	Learning Methods					
	SL		TL		TL-nDA	
	5-cv	Test	5-cv	Test	5-cv	Test
VGG19	96.51±0.76	96.93	**99.34±0.23**	98.81	92.72±7.39	98.45
InceptionResNet	**98.63±0.31**	98.15	99.05±0.19	98.61	92.79±7.40	98.38
MobileNetV2	97.10±0.78	97.10	99.13±0.21	98.96	93.97±6.51	98.93
DenseNet121	98.61±0.32	**98.41**	99.27±0.11	**99.48**	**95.41±5.06**	**99.28**

TABLE 3. The performance (mean validation accuracy: 5-cv and 10-cv, standard deviation, and test accuracy) comparison of the CNN models using different learning methods with other studies on the THI-C68 dataset.

Methods	Set-I		Set-II	
	5-cv	Test	10-cv	Test
SiftD-SVM [1]	-	-	98.93 ± 0.03	94.34
HOGFoDRs-SVM [26]	98.76	-	-	-
MobileNetV2-TL	99.13 ± 0.21	98.96	99.16 ± 0.31	**99.31**
DenseNet-TL	**99.27 ± 0.11**	**99.48**	**99.30 ± 0.36**	99.03
MobileNetV2-TL-nDA	93.97 ± 6.51	98.93	94.69 ± 7.62	99.10
DenseNet-TL-nDA	95.41 ± 5.06	99.28	95.44 ± 6.88	99.17

We proposed four noisy data augmentations: low resolution, AWGN, motion blur, and mixed noise, which were generated as a training set of the n-THI-C68 dataset.

The hyperparameters in CNN models were defined as follows: training epochs = 100 epochs, batch size = 32,

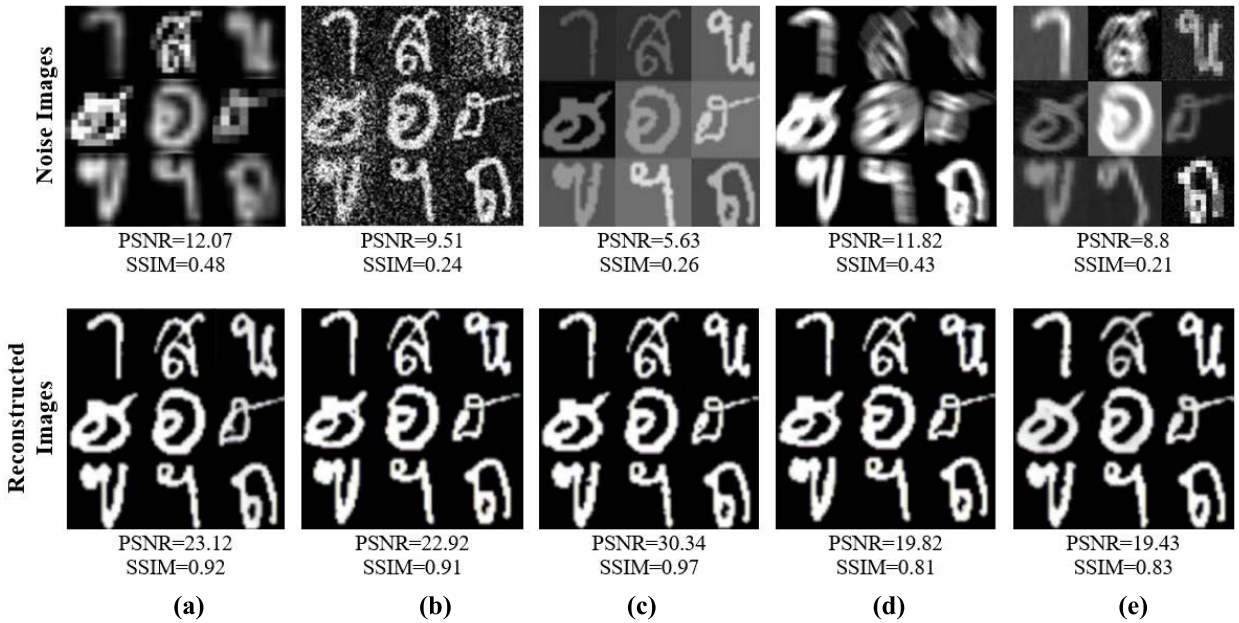

FIGURE 6. Illustration of the noisy images of (a) low resolution, (b) AWGN, (c) low contrast, (d) motion blur, and (e) mixed noise, as shown in the first row and reconstructed images using DeblurGAN architecture, as shown in the second row. Note that the high PSNR value presents better performance accuracy, and the high SSIM value presents the most similar character images between the reconstructed and original images.

TABLE 4. The performance of the CNN architectures and DeblurGAN-CNN architectures on the n-THI-C68 dataset.

Noise Methods	CNN Architectures				DeblurGAN-CNN Architectures	
	MobileNetV2-TL	DenseNet121-TL	MobileNetV2-TL-nDA	DenseNet121-TL-nDA	DeblurGAN-MobileNetV2	DeblurGAN-DenseNet121
Low Resolution	77.24	49.90	93.02	93.96	98.48	**98.52**
AWGN	27.92	16.63	96.72	98.21	98.72	**99.03**
Low Contrast	13.80	31.30	95.62	93.51	99.28	**99.41**
Motion Blur	45.89	49.59	91.75	93.06	**97.96**	97.69
Mixed Noise	30.78	25.02	93.93	92.89	97.90	**98.00**
Overall	39.13	34.49	94.21	94.33	98.47	**98.53**

stochastic gradient descent (SGD) optimizer, learning rate = 0.001, decay rate = 0.0001, momentum = 0.9, and image size = 128 × 128 pixels which is the smallest input of the InceptionResNet architecture. In transfer learning, we also used the pre-trained CNN model that learned on the ImageNet Dataset [35].

The accuracy results of CNN architectures are shown in Table 2. The accuracy performance of the CNNs was above 97% accuracy. The VGG19 architecture achieved the lowest performance on the THI-C68 dataset with an accuracy of 96.93% when training from scratch. On the other hand, the DenseNet121 architecture achieved the best performance in all learning methods with an accuracy of 99.48% when using transfer learning.

Furthermore, we demonstrate that noisy data can decrease the recognition performance of the CNN architectures. This experiment then applied four noisy data augmentation techniques while training the CNN model using the transfer learning method. It clearly showed that the accuracy of DenseNet121 was slightly decreased from 99.48% to 99.28% when training with noisy images. Subsequently, we proposed the DeblurGAN-CNN architecture to address the problems of noisy images. The result of the DeblurGANs is shown in the Section B.

2) COMPARISON OF THE CNNS AND OTHER STUDIES

According to previous experiments, we selected two CNN architectures, DenseNet121 and MobileNetV2. In this study, two CNN architectures were used and hand-crafted feature extraction combined with machine learning, namely SiftD-SVM [1] and HOGFoDRs-SVM [26], were evaluated and compared on the THI-C68 dataset.

To consider a fair comparison between CNN architectures and previous studies, we provided two shuffled random subsets of the THI-C68 dataset according to the experiments of Surinta *et al.* [1] and Inkeaw *et al.* [26]. The first subset (Set-I) had 11,592 training samples and 2,898 test samples. The second subset (Set-II) had 13,041 training samples

and 1,449 test samples. Note that, Set-I and Set-II were compared with the HOGFoDRS-SVM and the SiftD-SVM methods.

The results reported in Table 3 show that the DenseNet121 architecture with transfer learning (DenseNet121-TL) outperformed every CNN architecture on both sets with 5-fold cross-validation. Consequently, DenseNet121-TL outperformed the HOGFoDRs-SVM method by 0.51% on Set-I and outperformed the SiftD-SVM method by 0.37%. Also, MobileNetV2 with transfer learning (MobileNetV2-TL) achieved the highest performance on the independent test set of Set-II with 99.31% accuracy. MobileNetV2-TL significantly outperformed the SiftD-SVM method by 4.97%.

From the results above, the CNN architectures with transfer learning impact improving the performance of handwritten character recognition. Consequently, the CNN models achieved better accuracy than the hand-crafted features [1], [26] on the THI-C68 dataset.

B. DENOISING PERFORMANCE OF DEBLURGAN ON THE N-THI-C68 DATASEST

In this experiment, the input images were the noisy images of the n-THI-C68 dataset with 128×128 pixels. We first reconstructed the denoise character images with 128×128 pixels using Wasserstein and content loss functions. The hyperparameters of DeblurGAN were applied as follows: the optimization algorithm is Adam, learning rate $= 0.0001$, momentum $= 0.9$ and 0.999, training epochs $= 200$, and batch size $= 32$.

To study the reconstruction quality of the denoise images, we evaluated the DeblurGAN architecture with two well-known image quality metrics called the peak signal to noise ratio (PSNR) and the structural similarity index (SSIM) on the n-THI-C68 dataset. The noise images with different noise methods and reconstructed images are shown in Figure 6. We reported the PSNR and SSIM values obtained when evaluating the different noise methods. High PSNR and SSIM values represent better accuracy and reconstruction of the image, respectively. We achieved the best PSNR and SSIM when using DeblurGAN to reconstruct the character images from noisy images of the low contrast, low resolution, and AWGN, respectively. However, motion blur and mixed noise were the most difficult to reconstruct.

The DeblurGAN architecture adds the residual blocks and global skip connection in the generator, making the DeblurGAN only learn a residual correction to transform the noisy images. The DeblurGAN could be more generalized in reconstructing the denoise images generated by multiple generations or from the unknown kernel. Importantly, the DeblurGAN [14] uses the WGAN-GP and perceptual loss when reconstructing denoise images, while the traditional neural networks use L1 and L2 optimization algorithms when reconstructing denoise images.

C. DENOISING PERFORMANCE OF DEBLURGAN ON THE N-THI-C68 DATASEST

This section presents the DeblurGAN-CNN architectures to perform on the n-THI-C68 dataset. In response to the experimental results, as shown in Section A, we selected two CNN architectures, DenseNet121 and MobileNetV2, as the CNN models. Hence, we connected DeblurGAN with CNN architecture, called DeblurGAN-DenseNet121 and DeblurGAN-MobileNetV2. Consequently, we compared the DeblurGAN-CNN architectures with the traditional CNN architectures to recognize the noisy character images, as shown in Table 4.

Table 4 shows that the CNN architecture achieved low accuracy when using MobileNetV2-TL. It attained 77.24% accuracy when recognizing the noisy images with low resolution. The worst performance of only 13.80% accuracy was achieved when recognizing low-contrast images. However, we found that when training the CNN model using transfer learning with noisy data augmentation techniques (TL-nDA), the accuracy increased from only 13.80% to 95.62% when using MobileNetV2-TL-nDA. The overall performance accuracy of MobileNetV2-TL-nDA and DenseNet121-TL-nDA was 94.21% and 94.33% respectively.

The results show that the DeblurGAN-CNN architectures could address the problems of the noisy character images by achieving higher performance above 97% accuracy on all noise methods. Subsequently, the DeblurGAN-DenseNet121 achieved 98.53% accuracy and slightly outperformed the DeblurGAN-MobileNetV2 that achieved an accuracy of 98.47%. Moreover, the DeblurGAN-CNN architectures significantly outperformed the DenseNet121-TL-nDA and MobileNetV2-TL-nDA (The result was significant at $p < .05$). The misclassified characters are shown in Figure 7.

We concluded that only training the CNN models using the transfer learning with noisy data augmentation techniques could achieve accuracy above 90% on the n-THI-C68 dataset, although, very high accuracy is required in the handwritten character tasks to reduce the error while using the output data. Importantly, we recommend using the DeblurGAN-CNN architectures as this study yielded promising and outstanding results.

D. COMPARISON OF THE DEBLURGAN-CNN ARCHITECTURE AND OTHER APPROACHES

We selected two DeblurGAN-CNN architectures: DeblurGAN-MobileNetV2 and DeblurGAN-DenseNet121, to evaluate generalization ability on other noisy datasets n-MNIST and THCC-67. Comparisons of results on the n-MNIST and THCC-67 datasets with the GAN-CNNs and other approaches are presented in Table 5 and Table 6.

Table 5 presents the comparison results between the proposed DeblurGAN-CNN architectures and other approaches on the n-MNIST dataset. As a result, the accuracy of the DeblurGAN-MobileNetV2 slightly outperformed the DeblurGAN-DenseNet121. The DeblurGAN-MobileNetV2

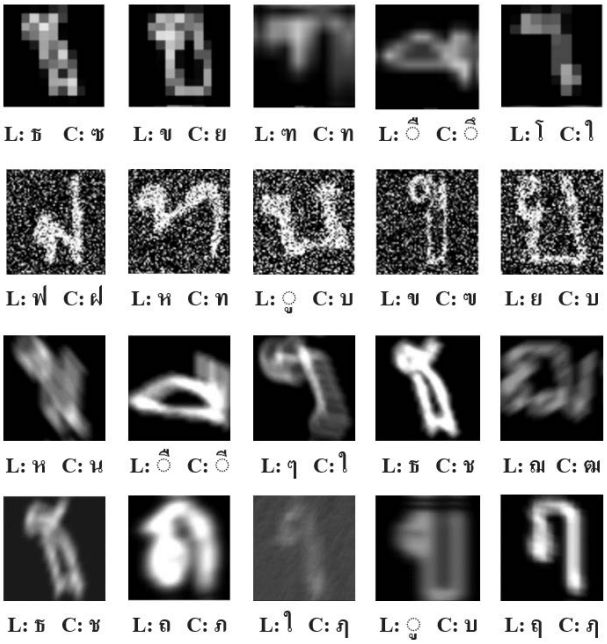

FIGURE 7. Illustration of misclassified characters on the test set of the n-THI-C68 dataset using DeblurGAN-CNN.

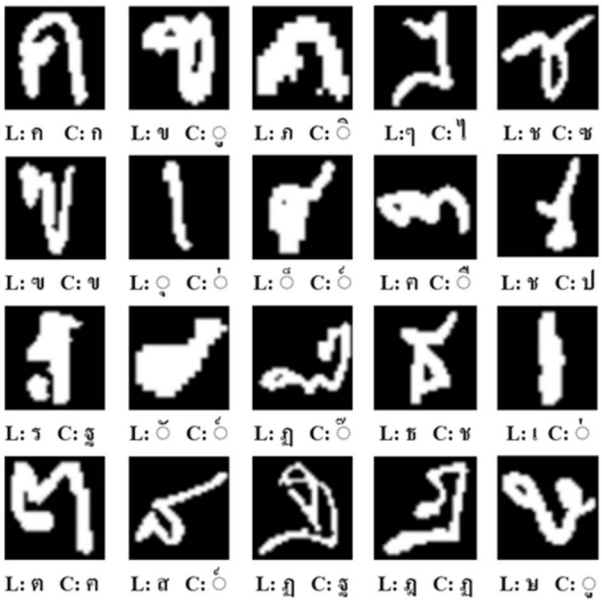

FIGURE 8. Illustration of the misclassified characters on the THCC-67 dataset using DeblurGAN-Densent121.

achieved the best accuracy on the n-MNIST dataset using AWGN and AWGN+Contrast noise methods.

The experimental results on the n-MNIST dataset showed that the optimal CNN-Hopfield network achieved an accuracy of 99.18%, 99.74%, and 97.53% when the AWGN, motion blur, and AWGN+Contrast noises were applied, respectively.

E. COMPARISON OF THE DEBLURGAN-CNN ARCHITECTURE AND OTHER APPROACHES

We selected two DeblurGAN-CNN architectures: DeblurGAN-MobileNetV2 and DeblurGAN-DenseNet121, to evaluate generalization ability on the other noisy datasets n-MNIST and THCC-67. Comparisons of results on the n-MNIST and THCC-67 datasets with the GAN-CNNs and other approaches are presented in Table 5 and Table 6.

Table 5 compares the results between the proposed DeblurGAN-CNN architectures and other approaches on the n-MNIST dataset. As a result, the accuracy of the DeblurGAN-MobileNetV2 slightly outperformed the DeblurGAN-DenseNet121. The DeblurGAN-MobileNetV2 achieved the best accuracy on the n-MNIST dataset using AWGN and AWGN+Contrast noise methods.

The experimental results on the n-MNIST dataset showed that the optimal CNN-Hopfield network achieved an accuracy of 99.18%, 99.74%, and 97.53% when the AWGN, motion blur, and AWGN+Contrast noises were applied, respectively.

On the other hand, the DeblurGAN-MobileNetV2 achieved 98.93%, 99.36%, and 97.59% accuracies when applying the AWGN, motion blur, and AWGN+Contrast noises, respectively. Further, the DeblurGAN-MobileNetV2 architecture outperformed the optimal CNN–Hopfield network on the n-MNIST dataset when applying AWGN+Contrast noise.

Undoubtedly, the DeblurGAN-CNN architectures demonstrated the highest accuracy performance compared with other methods on the n-MNIST dataset when AWGN+Contrast noise was applied.

Table 6 evaluated the DeblurGAN-CNN architectures on the THCC-67 dataset and compared them

TABLE 5. The performance comparison of DeblurGAN-CNN architectures with other approaches on the n-MNIST dataset.

Methods	Noise Methods		
	AWGN	Motion Blur	AWGN+ Contrast
PQ-DBN [46]	90.07	97.40	92.16
Dropconnect DBN [50]	97.57	97.20	96.93
PixelCNN PQ-DBN [50]	97.62	97.20	95.04
PCGAN-CHAR [52]	98.43	99.20	97.25
Optimal CNN-Hopfield Network [53]	**99.18**	**99.74**	97.53
DeblurGAN-MobileNetV2 (Proposed method)	98.93	99.36	**97.59**
DeblurGAN-DenseNet (Proposed method)	98.89	99.40	97.51

TABLE 6. The performance comparison of DeblurGAN-CNN architectures with the HOGFoDRs-SVM method on the THCC-67 dataset.

Methods	Accuracy
HOGFoDRs-SVM [26]	70.74
DeblurGAN-MobileNetV2 (Proposed method)	80.63
DeblurGAN- DenseNet121 (Proposed method)	**80.68**

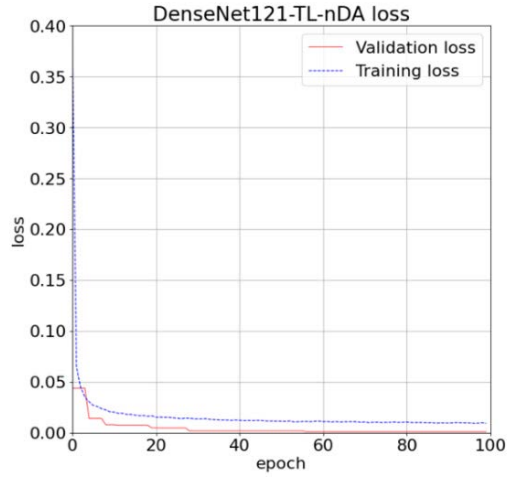

FIGURE 9. Illustration of the validation and training loss (a) DenseNet121-TL-nDA (b) DeblurGAN-DenseNet121 and (c) comparison of improving in validation loss.

with the HOGFoDRs-SVM method. We showed that the proposed DeblurGAN-CNN architectures significantly outperformed the existing method by more than 10%. Consequently, we achieved only 80.68% accuracy with the DeblurGAN-DenseNet121.

We illustrated the misclassified characters recognized using the DeblurGAN-DenseNet121, as shown in Figure 8.

Also, there is still scope to increase the performance of this dataset. Indeed, the proposed DeblurGAN-CNN architectures could be applied to classify the noisy image datasets, even with the THCC-67, the unseen noisy dataset.

VI. DISCUSSION

We observed the training loss between the DenseNet121-TL-nDA and DeblurGAN-DenseNet121, as shown in Figures 9(a) and 9(b). The improvement of validation loss is shown in Figure 9(c). It can be seen that the training loss of the DeblurGAN-DenseNet121 is relatively low in the early epochs due to the transferring of pre-trained weights. The training loss of the DeblurGAN-DenseNet121 is always lower than the DenseNet121-TL-nDA.

As shown in Figure 10, we found that the DenseNet121 model with TL (DenseNet121-TL) achieved unsatisfactory performance when evaluated on the noisy images. The accuracy of DenseNet121-TL quickly dropped when the PSNR value was increased. The result shows that DenseNet121-TL-nDA obtained much better performance than DenseNet121-TL. However, the accuracy of DenseNet121-TL-nDA was quickly decreased when the PSNR value was higher than 20. Furthermore, the DeblurGAN-DenseNet121, when training using TL-nDA methods, achieved high accuracy even when the PSNR value was increased more to than 26, with an accuracy above 90%.

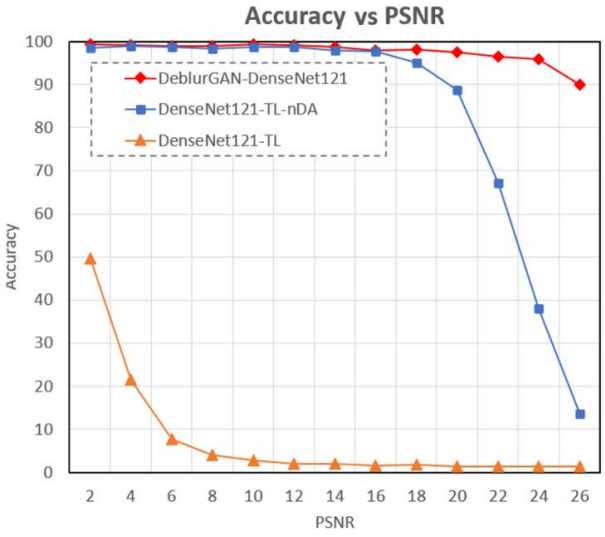

FIGURE 10. The effectiveness of different denoise architectures proposed to recognize the noisy character images on the n-THI-C68 dataset.

We also discussed in-depth the proposed DeblurGAN-CNN architecture and the optimal CNN-Hopfield network on the n-MNIST dataset in terms of accuracy. Therefore,

the optimal CNN-Hopfield network [53] outperformed our proposed architecture because the optimal CNN-Hopfield network is an ensemble method that combines many CNN outputs to achieve better recognition. The ensemble method has been reported to guarantees better accuracy in much published research [54], [55], [56]. On the other hand, the DeblurGAN-CNN architecture is a deep learning architecture that combines GAN and CNN architectures. So, only one output is recognized from the proposed architecture. Consequently, the optimal CNN-Hopfield network achieved an accuracy of 62%, 92%, and 97.52% when recognized using one, two, and three CNN models. In comparison, our proposed method achieved an accuracy of 98.93% using only one model and given an accuracy higher than 6% compared to the optimal CNN-Hopfield network that uses three CNN models.

Furthermore, finding texts that appear in natural scene images is challenging. To solve this challenge, object and scene text detection in the wild should be first applied to obtain the region of interest, which is the area of texts. Second, we could employ the DeblurGAN-CNN method to denoise and recognize the text in the natural scene images. This solution could enhance the recognition performance. In future work, we will concentrate on finding and recognizing text that appears in natural scene images.

VII. CONCLUSION

The performance of the handwritten character recognition systems decreases in consequence of many problems, such as handwriting styles, degradation of the documents, and noise appearance while transforming documents into a digital format. This research mainly focused on the denoise and recognition of noisy handwritten character images. Consequently, the robust generative adversarial network (GAN) combined with the convolutional neural network (CNN) architecture, called DeblurGAN-CNN, was proposed to synthesize new clean handwritten characters from noisy handwritten characters and recognition with improved handwritten character performance. For the CNN architecture, we combined two state-of-the-art CNNs: MobileNetV2 and DenseNet121, with the DeblurGAN, called DeblurGAN-MobileNetV2 and DeblurGAN-DenseNet121. The DeblurGAN-CNN architectures were trained using the transfer learning technique and applying the noisy data augmentation techniques to create a robust model. The most beneficial aspect of the DeblurGAN-CNN models was that they could learn and generalize from many noisy methods, including low resolution, additive white Gaussian noise (AWGN), low contrast, motion blur, and mixed noise.

To evaluate the denoise model, the DeblurGAN produced significant output that achieved a high peak signal to noise ratio (PSNR) and structural similarity index (SSIM) values. As a result, the DeblurGAN architecture could remove various noises from the noisy handwritten character images. For the accuracy performance, the results show that the DeblurGAN-CNN architectures generated strong handwritten character images and achieved the highest performance on the n-MNIST and n-THI-C68 datasets when compared with other existing methods. Also, both DeblurGAN-DenseNet121 and DeblurGAN-MobileNetV2 presented significant performance and outperformed the HOGFoDRs-SVM on the THI-C68 and THCC-67 datasets. The DeblurGAN-CNN architectures achieved an accuracy above 98%, 97.59%, and 80.68% on the n-THI-C68, n-MNIST, and THCC-67 datasets. Subsequently, the DeblurGAN-CNN architectures, which used the DenseNet121 and MobileNetV2 as the CNN architectures, achieved high handwritten character recognition performance with and without noisy handwritten characters.

In the future, we plan to work on the ensemble CNNs technique and combine the DeblurGAN-CNN architecture as a part of the ensemble CNNs technique [54], [55] to achieve much higher accuracy. Another direction for future work is creating new DeblurGAN-CNN architecture by searching for efficient CNN architectures with lightweight models. We will embed DeblurGAN-CNN with the recurrent neural networks (RNNs) [57] or vision transformers [11], [58] to recognize word and sentence images. Finally, finding the text from the natural scene images using the object detection methods [59], [60] and recognition by our DeblurGAN-CNN is also another direction we wish to pursue.

REFERENCES

[1] O. Surinta, M. F. Karaaba, L. R. B. Schomaker, and M. A. Wiering, "Recognition of handwritten characters using local gradient feature descriptors," *Eng. Appl. Artif. Intell.*, vol. 45, pp. 405–414, Oct. 2015, doi: 10.1016/j.engappai.2015.07.017.

[2] M. Z. Alom, P. Sidike, M. Hasan, T. M. Taha, and V. K. Asari, "Handwritten Bangla character recognition using the state-of-the-art deep convolutional neural networks," *Comput. Intell. Neurosci.*, vol. 2018, pp. 1–13, Aug. 2018, doi: 10.1155/2018/6747098.

[3] J. Su, D. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *IEEE Trans. Evol. Comput.*, vol. 23, no. 5, pp. 828–841, Oct. 2019, doi: 10.1109/TEVC.2019.2890858.

[4] J. Mei, Z. Wu, X. Chen, Y. Qiao, H. Ding, and X. Jiang, "DeepDeblur: Text image recovery from blur to sharp," *Multimedia Tools Appl.*, vol. 78, no. 13, pp. 18869–18885, 2019, doi: 10.1007/s11042-019-7251-y.

[5] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.

[6] S. Gonwirat and O. Surinta, "Improving recognition of Thai handwritten characters with deep convolutional neural networks," in *Proc. 3rd Int. Conf. Inf. Sci. Syst.*, Mar. 2020, pp. 82–87, doi: 10.1145/3388176.3388181.

[7] M. Eltay, A. Zidouri, I. Ahmad, and Y. Elarian, "Generative adversarial network based adaptive data augmentation for handwritten Arabic text recognition," *Peer J. Comput. Sci.*, vol. 8, pp. 1–22, Jan. 2022, doi: 10.7717/PEERJ-CS.861.

[8] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 295–307, Feb. 2015, doi: 10.1109/TPAMI.2015.2439281.

[9] K. Zhang, W. Zuo, S. Gu, and L. Zhang, "Learning deep CNN denoiser prior for image restoration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2808–2817, doi: 10.1109/CVPR.2017.300.

[10] L. Gondara, "Medical image denoising using convolutional denoising autoencoders," in *Proc. IEEE 16th Int. Conf. Data Mining Workshops (ICDMW)*, Dec. 2016, pp. 241–246, doi: 10.1109/ICDMW.2016.0041.

[11] M. A. Souibgui, S. Biswas, S. K. Jemni, Y. Kessentini, A. Fornés, J. Lladós, and U. Pal, "DocEnTr: An end-to-end document image enhancement transformer," in *Proc. Int. Conf. Pattern Recognit. (ICPR)*, 2022, pp. 1–7.

[12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Int. Conf. Neural Inf. Process. Syst. (NIPS)*, 2014, pp. 2672–2680, doi: 10.48550/arXiv.1406.2661.

[13] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 11125–11134, doi: 10.1109/CVPR.2017.632.

[14] O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin, and J. Matas, "DeblurGAN: Blind motion deblurring using conditional adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8183–8192, doi: 10.1109/CVPR.2018.00854.

[15] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of Wasserstein GANs," in *Proc. Int. Conf. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 1–20, doi: 10.48550/arXiv.1704.00028.

[16] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 1–18, doi: 10.1007/978-3-319-46475-6_43.

[17] M. Sharma, A. Verma, and L. Vig, "Learning to clean: A GAN perspective," in *Proc. 14th Asian Conf. Comput. Vis. (ACCV)*, 2018, pp. 174–185, doi: 10.1007/978-3-030-21074-8_14.

[18] A. K. Bhunia, A. K. Bhunia, A. Sain, and P. P. Roy, "Improving document binarization via adversarial noise-texture augmentation," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 2721–2725, doi: 10.1109/ICIP.2019.8803348.

[19] S. K. Jemni, M. A. Souibgui, Y. Kessentini, and A. Fornés, "Enhance to read better: A multi-task adversarial network for handwritten document image enhancement," *Pattern Recognit.*, vol. 123, Mar. 2022, Art. no. 108370, doi: 10.1016/j.patcog.2021.108370.

[20] M. A. Souibgui and Y. Kessentini, "DE-GAN: A conditional generative adversarial network for document enhancement," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 3, pp. 1180–1191, Mar. 2022, doi: 10.1109/TPAMI.2020.3022406.

[21] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998, doi: 10.1109/5.726791.

[22] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 4, pp. 509–522, Apr. 2002, doi: 10.1109/34.993558.

[23] O. Surinta, M. F. Karaaba, T. K. Mishra, L. R. B. Schomaker, and M. A. Wiering, "Recognizing handwritten characters with local descriptors and bags of visual words," in *Proc. Int. Conf. Eng. Appl. Neural Netw. (EANN)*, vol. 517, 2015, pp. 255–264, doi: 10.1007/978-3-319-23983-5.

[24] J. L. Maas, E. Okafor, and M. A. Wiering, "The dual codebook: Combining bags of visual words in image classification," in *Proc. Belgian-Dutch Artif. Intell. Conf. (BNAIC)*, 2016, pp. 1–8.

[25] S. H. Abdulhussain, B. M. Mahmmod, M. A. Naser, M. Q. Alsabah, R. Ali, and S. A. R. Al-Haddad, "A robust handwritten numeral recognition using hybrid orthogonal polynomials and moments," *Sensors*, vol. 21, no. 6, p. 1999, Mar. 2021, doi: 10.3390/s21061999.

[26] P. Inkeaw, J. Bootkrajang, S. Marukatat, T. Gonçalves, and J. Chaijaruwanich, "Recognition of similar characters using gradient features of discriminative regions," *Expert Syst. Appl.*, vol. 134, pp. 120–137, Nov. 2019, doi: 10.1016/j.eswa.2019.05.050.

[27] A. Onuean, U. Buatoom, T. Charoenporn, T. Kim, and H. Jung, "BuraphaTH: A multi-purpose character, digit, and syllable handwriting dataset," *Appl. Sci.*, vol. 12, no. 8, p. 4083, Apr. 2022, doi: 10.3390/app12084083.

[28] Z. Wang, Q. She, and T. E. Ward, "Generative adversarial networks in computer vision: A survey and taxonomy," *ACM Comput. Surv.*, vol. 54, no. 2, pp. 1–38, Jun. 2021, doi: 10.1145/3439723.

[29] A. Karnewar and O. Wang, "MSG-GAN: Multi-scale gradients for generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 7796–7805, doi: 10.1109/CVPR42600.2020.00782.

[30] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. C. Loy, "ESRGAN: Enhanced super-resolution generative adversarial networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 1–16, doi: 10.1007/978-3-030-11021-5_5.

[31] S. Fogel, H. Averbuch-Elor, S. Cohen, S. Mazor, and R. Litman, "ScrabbleGAN: Semi-supervised varying length handwritten text generation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1–12, doi: 10.1109/CVPR42600.2020.00438.

[32] C. Wu, H. Du, Q. Wu, and S. Zhang, "Image text deblurring method based on generative adversarial network," *Electronics*, vol. 9, no. 2, pp. 1–14, 2020, doi: 10.3390/electronics9020220.

[33] J. Zhao, Y. Wang, B. Xiao, C. Shi, J. Jiang, and C. Wang, "Adversarial learning based attentional scene text recognizer," *Pattern Recognit. Lett.*, vol. 138, pp. 217–222, 2020, doi: 10.1016/j.patrec.2020.07.027.

[34] S. Liu and W. Deng, "Very deep convolutional neural network based image classification using small training sample size," in *Proc. 3rd IAPR Asian Conf. Pattern Recognit. (ACPR)*, 2016, pp. 730–734, doi: 10.1109/ACPR.2015.7486599.

[35] 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, doi: 10.1109/CVPR.2009.5206848.

[36] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–14.

[37] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[38] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, Inception-ResNet and the impact of residual connections on learning," in *Proc. 31st AAAI Conf. Artif. Intell. (AAAI)*, 2017, pp. 4278–4284, doi: 10.1016/j.patrec.2014.01.008.

[39] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2261–2269, doi: 10.1109/CVPR.2017.243.

[40] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.

[41] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520, doi: 10.1109/CVPR.2018.00474.

[42] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8697–8710, doi: 10.1109/CVPR.2018.00907.

[43] D. Ciresan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3642–3649, doi: 10.1109/CVPR.2012.6248110.

[44] S. Ahlawat, A. Choudhary, A. Nayyar, S. Singh, and B. Yoon, "Improved handwritten digit recognition using convolutional neural networks (CNN)," *Sensors*, vol. 20, no. 12, pp. 1–18, Jun. 2020, doi: 10.3390/S20123344.

[45] Y. Tang, L. Peng, Q. Xu, Y. Wang, and A. Furuhata, "CNN based transfer learning for historical Chinese character recognition," in *Proc. 12th IAPR Workshop Document Anal. Syst. (DAS)*, Apr. 2016, pp. 25–29, doi: 10.1109/DAS.2016.52.

[46] S. Basu, M. Karki, S. Ganguly, R. DiBiano, S. Mukhopadhyay, S. Gayaka, R. Kannan, and R. Nemani, "Learning sparse feature representations using probabilistic quadtrees and deep belief nets," *Neural Process. Lett.*, vol. 45, no. 3, pp. 855–867, 2015, doi: 10.1007/s11063-016-9556-4.

[47] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, Jan. 2013, doi: 10.1109/TPAMI.2012.59.

[48] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *Artif. Intell. Rev.*, vol. 53, no. 8, pp. 5455–5516, 2020, doi: 10.1007/s10462-020-09825-6.

[49] S. Sae-Tang and L. Methasate, "Thai handwritten character corpus," in *Proc. IEEE Int. Symp. Commun. Inf. Technol. (ISCIT)*, Oct. 2004, pp. 486–491, doi: 10.1109/ISCIT.2004.1412893.

[50] M. Karki, Q. Liu, R. Dibiano, S. Basu, and S. Mukhopadhyay, "Pixel-level reconstruction and classification for noisy handwritten Bangla characters," in *Proc. 16th Int. Conf. Frontiers Handwriting Recognit. (ICFHR)*, Aug. 2018, pp. 511–516, doi: 10.1109/ICFHR-2018.2018.00095.

[51] G. Boracchi and A. Foi, "Uniform motion blur in Poissonian noise: Blur/noise tradeoff," *IEEE Trans. Image Process.*, vol. 20, no. 2, pp. 592–598, Feb. 2011, doi: 10.1109/TIP.2010.2062196.

[52] Q. Liu, E. Collier, and S. Mukhopadhyay, "PCGAN-CHAR: Progressively trained classifier generative adversarial networks for classification of noisy handwritten Bangla characters," in *Proc. Int. Conf. Asian Digit. Libraries (ICADL)*, 2019, pp. 3–15, doi: 10.1007/978-3-030-34058-2_1.

[53] F. E. Keddous and A. Nakib, "Optimal CNN–Hopfield network for pattern recognition based on a genetic algorithm," *Algorithms*, vol. 15, no. 1, p. 11, Dec. 2021, doi: 10.3390/a15010011.
[54] S. Gonwirat and O. Surinta, "Optimal weighted parameters of ensemble convolutional neural networks based on a differential evolution algorithm for enhancing pornographic image classification," *Eng. Appl. Sci. Res.*, vol. 48, no. 5, pp. 560–569, 2021, doi: 10.14456/easr.2021.58.
[55] L. Guo, S. Du, Y. Chi, W. Cui, P. Song, J. Zhu, S. Geng, and M. Xu, "A multi-model ensemble method using CNN and maximum correntropy criterion for basal cell carcinoma and seborrheic keratoses classification," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–6, doi: 10.1109/IJCNN.2019.8852434.
[56] S. Noppitak and O. Surinta, "DropCyclic: Snapshot ensemble convolutional neural network based on a new learning rate schedule for land use classification," *IEEE Access*, vol. 10, pp. 60725–60737, 2022, doi: 10.1109/access.2022.3180844.
[57] M. Ameryan and L. Schomaker, "A limited-size ensemble of homogeneous CNN/LSTMs for high-performance word classification," *Neural Comput. Appl.*, vol. 33, no. 14, pp. 8615–8634, Feb. 2021, doi: 10.1007/s00521-020-05612-0.
[58] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16×16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2021, pp. 1–2.
[59] C. Luo, L. Jin, and Z. Sun, "MORAN: A multi-object rectified attention network for scene text recognition," *Pattern Recognit.*, vol. 90, pp. 109–118, Jun. 2019, doi: 10.1016/j.patcog.2019.01.020.
[60] M. Cordova, A. Pinto, H. Pedrini, and R. D. S. Torres, "Pelee-Text++: A tiny neural network for scene text detection," *IEEE Access*, vol. 8, pp. 223172–223188, 2020, doi: 10.1109/ACCESS.2020.3043813.

SARAYUT GONWIRAT received the B.E. and M.E. degrees in computer engineering from the King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand. He is currently pursuing the Ph.D. degree with the Department of Information Technology, Faculty of Informatics, Mahasarakham University, Mahasarakham, Thailand. His research interests include optimization algorithm in logistic problems and deep learning in computer vision.

OLARIK SURINTA received the Ph.D. degree in artificial intelligence from the University of Groningen, The Netherlands, in 2016. He currently works at the Department of Information Technology, Faculty of Informatics, Mahasarakham University, Thailand, and also a Research Member with the Multi-Agent Intelligent Simulation Laboratory (MISL). His research interests include historical document analysis and recognition, deep learning, machine learning, and image and video classifications.