❑ 3243

Deblurring image compression algorithm using deep convolutional neural network

Rafik Menassel[1,2], Abdeljalil Gattal[1,2], Fateh Kerdoud[2]
[1]Laboratoire de Vision et d'Intelligence Artificielle (LAVIA), Echahid Cheikh Larbi Tebessi University, Tebessa, Algeria
[2]Department of Mathematics and Computer Science, Echahid Cheikh Larbi Tebessi University, Tebessa, Algeria
[2]Faculty of Exact Sciences and Natural and Life Sciences, Echahid Cheikh Larbi Tebessi University, Tebessa, Algeria

Article Info

ABSTRACT

There are instances where image compression becomes necessary; however, the use of lossy compression techniques often results in visual artifacts. These artifacts typically remove high-frequency detail and may introduce noise or small image structures. To mitigate the impact of compression on image perception, various technologies, including machine learning and optimization metaheuristics that optimize the parameters of image compression algorithms, have been developed. This paper investigates the application of convolutional neural networks (CNNs) to reduce artifacts associated with image compression, and it presents a proposed method termed deblurring compression image using a CNN (DCI-CNN). Trained on a UTKFace dataset and tested on six benchmark images, the DCI-CNN aims to address artifacts such as block artifacts, ringing artifacts, blurring artifacts, color bleeding, and mosquito noise. The DCI-CNN application is designed to enhance the visual quality and fidelity of compressed images, offering a more detailed output compared to generic and other deep learning-based deblurring methods found in related work.

Corresponding Author:

Rafik Menassel
Laboratoire de Vision et d'Intelligence Artificielle (LAVIA), Echahid Cheikh Larbi Tebessi University
Route de constantine, 12022, Tebessa, Algeria
Email : r.menassel@univ-tebessa.dz

1. INTRODUCTION

A few examples of the many uses for picture deblurring include retrieving clear satellite images from hazy data, image compression, and image decompression [1], [2]. It may also be used to draw out more specific data from monitoring images with limited resolution. Deblurring images, however, has long been a difficult issue in the realm of image processing. Additionally, it implies that modern image-denoising technologies have demonstrated superior performance compared to traditional methods in addressing the issue of image blurring. A really blurred image can only be slightly clarified with traditional methods. Therefore, we focus on a more challenging goal than the preceding approaches: recovering a high-resolution image from a collection of very blurred images [3], [4]. So, we suggested applying an end-to-end deep neural network, especially for this work.

Lossy compression is unique in that it introduces losses into compressed images, such as distortions, glitches, or degradations. Then, it can only be successful if the losses imposed do not have a considerable adverse effect. One assumption is that the imposed losses must be equal to or less than the noise-induced degradation in the original image [1], [3]. Considering their direct impact on the quality of the reconstructed image, noise characteristics in image processing must be taken into account and anticipated. Noise is a component of processes such as compression and can cause degradations, glitches, and distortions. By

anticipating and analyzing noise characteristics in advance [4]-[6], practitioners can reduce the impact of these effects. This proactive approach ensures that the imposed losses do not negatively impact the final image and allows for effective control over newly introduced artifacts. Furthermore, understanding noise characteristics is essential to figuring out the ideal degree of loss, which is consistent with the general objective of improving and restoring images in a variety of applications, including satellite imaging, data retrieval, and image compression. Presently, the predominant state-of-the-art techniques for handling blurry images rely on deep neural networks [7]-[10]. These algorithms, as per existing methodologies, presume a consistent level of noise or blurring applied uniformly across all training and testing images.

In this study, we investigate an additional type of blurring that is given to images in an undetermined quantity. Deblurring, denoising, restoration, and super-resolution (SR) must thus be addressed together in a single method. The aim of this study is to devise an innovative solution, drawing inspiration from deep convolutional neural networks (CNNs), to address different distinct types of artifacts commonly caused by image compression techniques: block artifacts, ringing artifacts, blurring artifacts, color bleeding, and mosquito noise. In this paper, we introduce a new architecture called the deblurring compression images-CNN (DCI-CNN), which utilizes a convolution layer as a network layer. Our DCI-CNN was trained on blurry, compressed images from the UTKFace dataset with an unknown level of blurring. We then tested it on six benchmark images and found that, in blind situations, our architecture is empirically more effective at deblurring compressed images.

This paper is organized as follows: the second section provides an overview of related works to contextualize our contribution. The third section details the proposed framework, followed by the fourth section presenting experimental results. Our focus in this paper is specifically on the restoration of scrambled and blurred compressed images, as observed in related works.

2. RELATED WORK

Most low-level imaging problems, such as image deblurring, restoration, denoising, and SR, can be mathematically resolved by taking into account different degradation operators and noise distributions. Image capture, transmission, and compression can all lead to corruption. All of these are essential challenges in image processing and computer vision. After reviewing several works, we discovered that state-of-the-art image deblurring methods frequently employ advanced algorithms and deep learning approaches [11]-[16]. Deep CNNs have recently benefited image-deblurring tasks. Certain new techniques, like simulating iterative optimization [17], predicting the deconvolution kernel in the frequency domain [11], [18], and predicting blur direction [18], [19], substitute steps in traditional techniques with deep CNN. A number of recent approaches, like generative adversarial network work [20], [21], multi-scale [14], [22], multi-modal [16], scale-recurrent [23], and encoder-decoder [9], [24], concentrate more on end-to-end network architectures and produce ever-more-complex architectures. This section focuses on research on using nature-inspired optimizers to train artificial neural networks for anomaly identification.

According to recent advancements in deconvolution research, regularization terms are crucial for sparse picture priors because they preserve sharp edges and reduce artifacts. In the study of Xu *et al.* [11], a deep CNN for image deconvolution was described. The authors propose an image deblurring technique that connects traditional optimization-based methods with a neural network architecture. They introduce a novel separable structure as a dependable support for robust deconvolution against artifacts. The proposed network consists of two submodules that were both properly initialized and supervised-trained. When compared to previous generative models, they perform non-blind image deconvolution reasonably effectively. To generate the two network modules, the final deconvolution layer of the deconvolution CNN connects with the input of the denoising CNN.

In another study by Levin *et al.* [12], developed a method for deblurring that investigates the key components of current blind deconvolution methods. One type of solution is explicit edge detection. They come to the conclusion that the variational Bayes approximation outperforms all alternatives. The authors examined current algorithms with comparable parameters after collecting blurry data using ground truth. Additionally, their research shows that most algorithms frequently fail to satisfy the shift-invariant blur assumption. However, the authors of the work of Krishnan *et al.* [13] demonstrate how their method of image regularization, which prefers sharp images over blurry ones, may be applied in a framework for blind deconvolution. The main idea behind the method is the creation of a novel scale-invariant regularizer that significantly stabilizes the kernel estimation process and attenuates high frequencies. The resulting approach is adaptable to various models of blur creation and is quick and robust for parameter selection.

Using big data and deep neural network architecture to train blurry images is another emerging trend in image SR. As a result, image noise degradation is no longer limited to one model. Cui *et al.* [2] introduced a novel deep network cascade (DNC) model designed to incrementally enhance low-resolution images layer

by layer, with each layer featuring a small-scale factor. The DNC consists of a series of stacked collaborative local auto-encoders (CLAs). Initially, a non-local self-similarity search is conducted in each stage of the cascade to enhance the high-frequency texture features of partitioned patches within the input image. The refined image patches are subsequently input into a CLA to mitigate artifacts and promote patch compatibility. Experimental results in image SR demonstrate that with the increase in network layers, the proposed DNC exhibits a gradual enhancement of low-resolution images, delivering increasingly promising outcomes in both visual quality and quantitative performance.

While Dong *et al.* [3] introduced a deep learning approach for single-image SR, their proposed technique involves learning an end-to-end mapping directly between low-and high-resolution images. Utilizing a deep CNN, the mapping is generated by taking a low-resolution image as input and producing a high-resolution image as output. However, in contrast to existing methods that optimize each component independently, our approach optimizes all layers simultaneously. The proposed deep CNN achieves high restoration quality while maintaining practical speed suitable for online use. According to Liu *et al.* [7], domain experience from the standard sparse coding approach can be merged with the key components of deep learning to generate even better outcomes. The authors illustrate the utilization of a neural network to articulate a sparse coding model tailored for single-image SR, leveraging the benefits of end-to-end optimization across training data. Due to its cascaded topology, the network enhances SR effectiveness for both fixed and incremental scaling factors. The suggested training and testing methodologies are extendable to accommodate images with additional degradations, such as noise and blurring.

In a separate study, Dong *et al.* [9] proposed a simple yet powerful CNN-based denoising autoencoder network that can be trained end-to-end in an unsupervised manner. This addresses two challenges: sensitivity to weight initialization and reliance on large amounts of labeled data. The network uses a fully convolutional autoencoder with symmetric encoder-decoder connections. By training the network to reconstruct images, it not only removes noise from corrupted images but also learns abstract image representations. The encoder of the network is also effective for classification, achieving competitive results even without additional unlabeled data through unsupervised pre-training.

Recently, Zhang *et al.* [10] investigated the potential of using neural architecture search (NAS) methods to automatically generate optimal network topologies for low-level image restoration tasks. The authors introduced a memory-efficient hierarchical NAS approach, called hierarchical neural architecture search (HiNAS), and applied it to two specific tasks: image SR and image denoising. HiNAS utilizes gradient-based search techniques and establishes a customizable hierarchical search space, which includes inner and outer search spaces responsible for creating cell structures and estimating cell widths. The authors also propose a layer-wise architecture-offering technique for the inner search space, resulting in more flexible structures and improved performance.

Single image deblurring and guided image deblurring are the two main categories of image deblurring techniques. In single-blind image deblurring, the latent image is restored by estimating the blur kernel. Bai *et al.* [14] proposed a method for restoring sharp images at different scales of a blurry image pyramid, gradually updating the prior image with the newly recovered sharp image. This approach, known as multi-scale latent structures (MSLS), has shown superior performance in single-image blind deblurring applications, with significantly improved speed. Similarly, Vijay and Deepa [15] introduced a novel image deblurring algorithm that utilizes CNNs and incorporates distortion reduction methods. This method avoids deconvolution and performs all functions within the hidden layers of the CNN to produce a clear output image.

Contrastingly, guided image deblurring algorithms incorporate additional information, such as edges and textures, from a guidance image to facilitate deblurring. In their study, Liu *et al.* [16] presented the guided deblurring fusion network (GDFNet), a deep fusion network for joint image deblurring. This network fuses pre-deblurred images generated by multiple image deblurring streams, addressing limitations observed in single deblurring streams and guided deblurring streams. As a result, it improves the restoration of detailed contents without introducing structural inconsistencies. To provide an overview of notable contributions in deblurring, restoration, denoising, and SR methods reported in the literature, Table 1 summarizes key findings.

Table 1. Comparing different systems or methods for image deblurring

Ref	Approach	Dataset	Evaluation metrics
Levin *et al*. [12]	Explicit edge detection	− Test set: Levin dataset (32 blurred images)	SSD error, MSE, and MAP
Krishnan *et al*. [13]	Scale-invariant regularizer	− Test set: Levin dataset (32 blurred images)	SSD error
Xu *et al*. [11]	Optimization-based methods with a CNN	− Train set: 2,500 natural images − Test set :16 images	PSNR
Cui *et al*. [2]	DNC	− Train set: large amount of patches from images − Test set: 9 images	PSNR and SSIM
Liu *et al*. [7]	Sparse coding based network (SCN)	− Train set: 91 images − Test set: Set5, Set14, and BSD100	PSNR and SSIM
Dong *et al*. [3]	SRCNN	− Train set: 91 images − Test set: Set5, Set14, and BSD100	PSNR, SSIM, IFC, NQM, WPSNR, and MSSI
Vijay and Deepa [15]	Deep CNN	− Train set: large amount of images − Test set: 6 images	PSNR and MSE
Bai *et al*. [14]	MSLS prior	− Test set: 640 blurry images	Mean error, worst error, and success rate
Zhang *et al*. [10]	Memory-efficient HiNAS	− Train set: BSD50, DIV2K − Test set: Set5, Set14, BSD100, Urban100, and Manga109	PSNR and SSIM
Liu *et al*. [16]	GDFNet	− RGB/NIR datase − Ambient/flash datase	SNR, SSIM, and LPIPS

3. PROPOSED APPROACH

In astronomy and medical imaging, deblurring is a widely used technique. However, in this context, we will employ image compression to distinguish a sharp and clear image from a blurred and damaged input image. In reality, we need to locate the image that is both clear and crisp in a hazy one. Artifacts in a blurred image can be removed using a variety of methods. In this study, adaptive learning with a CNN is used to attain this aim. For example, we use the UTKFace dataset's blurred compressed images with an undetermined level of blurring to test our model of single image deblurring, which we call DCI-CNN. If we feed a blurry compressed image through our proposed deep CNN (DCI-CNN), the system will produce a clear image.

The DCI-CNN framework is composed of 22 weight layers, including eight convolutional layers, seven batch normalization layers, and seven activation layers. The convolutional layers use small 3×3 filters with a stride of 1. Each convolutional layer is followed by a batch normalization layer and a rectified linear unit (ReLU) activation. Unlike other methods, such as the supervised pretraining proposed by Xu *et al*. [11], our approach involves two stages. First, the deconvolution CNN is initialized and trained using pairs of blurry and clear images. Then, this trained network is used as a generator for deconvolved images containing outliers. These generated images, along with their clear counterparts, are used to train the outlier-rejection deconvolution CNN. The networks are integrated by modifying the architecture and fine-tuning the entire structure. In a separate study, Vijay and Deepa [15] proposed a CNN structure with two convolutional layers, ReLU layers, two Max-pooling layers, and fully connected layers. In contrast, our proposed DCI-CNN achieves impressive results without the use of fully connected layers, as it is trained with blurry images and their corresponding clear counterparts. The proposed method is illustrated in Figure 1.
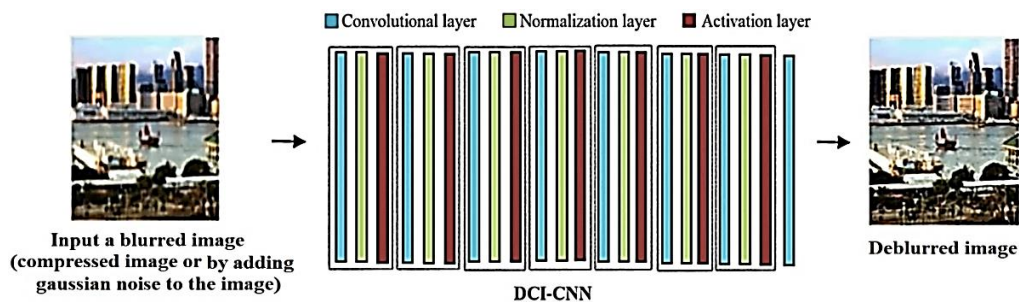
Figure 1. Architecture of the proposed approach

The transparency and interpretability of the DCI-CNN model aid in making more reliable decisions by offering insights into the significance of the input space, and analyzing the intermediate relevance filters at hidden layers. Our efforts in explaining the DCI-CNN model have yielded interesting results in terms of its processing, network representation, filter-level explanation, and classification. The following Algorithm 1 governs how our model operates.

```
Overall Algorithm 1
      Introduce RGB images of size 64.
      For i from 1 to 7 (7 blocks) do:
          1- Applying the convolutions on the image with a convolution kernel (3 × 3
          × 3), When the number of strides is 1 and apply 1 pixel of padding on all
          sides, so that the layer's outputs will have the same spatial dimensions as
          its inputs.
          2- Applying batch normalization significantly decrease the number of
          training epochs needed to train deep networks.
          3- Applying the activation function.
      End for.
      Apply a convolutional layer with 1 filter that produces the output image.
  Return Deblurred Image.
```

The proposed method for deblurring compressed images in blind scenarios holds promise for real-world applications, operating on the premise that the degradation process is unknown [25]. It features an exceptionally deep network architecture composed of symmetric convolutional layers, designed to effectively eliminate artifacts as these layers function as feature extractors. The CNN adapts to each input compressed blurred image, aiming to mitigate blurring. The structural details are elucidated in the illustration, presented in Figure 2.

When incorporated into a CNN framework, the trained model can be applied to a blurry compressed image, generating a deblurred image. Unlike adjusting hyperparameters during training, backpropagation is commonly employed along with the Adam optimization method [26]. Adam enhances the optimization process by dynamically adjusting learning rates, contributing to its efficiency.

The proposed DCI-CNN requires training a CNN. The following is a summary of the algorithm:

a. Data preparation: produce a dataset of pairs of blurry compressed images and their deblurred alternatives. The CNN will be trained using these pairs. The images should be preprocessed by resizing them to a consistent size and normalizing their pixel values.
b. Model architecture: create a CNN architecture for deblurring. The network receives a blurry image as input and returns a deblurred image as output.
c. Cost function: set up a cost function involving mean squared error (MSE). The proposed DCI-CNN involves training a CNN.
d. Training: divide the dataset into two sets: training and testing. Using the training set, we can train our CNN. The network learns to minimize the cost function during training. Use Adam, an optimizer, to adjust the network's parameters based on backpropagation gradients.
e. Testing: after training, evaluate the model on unseen test images to ensure generalization. To evaluate the quality of deblurring, use metrics such as peak signal-to-noise ratio (PSNR) and MSE.

Remember that deblurring images with CNNs is a complex operation, and the quality of the results relies on numerous factors, such as the quality and diversity of the training data, the selected architecture, and hyperparameter adjustments. Experimentation and iterative improvement are frequently essential for obtaining the best results.

3.1. Cost function

There are various metrics and measures commonly used to evaluate the regression cost functions of a neural network model. In our case, the CNN is trained by minimizing the MSE cost function, as shown in (1), which is a widely accepted approach. The MSE cost function calculates the average squared difference between the pixels of the ground truth image and the predicted image. It is important to note that there are multiple metrics and measures that can be applied to the regression cost functions of a neural network model. The MSE cost function is defined as (1):

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2) \tag{1}$$

Adam is designed to optimize model parameters by iteratively adjusting them to minimize the MSE cost function. It accomplishes this by computing the gradients of the cost function with respect to the model parameters and then utilizing these gradients to alter the parameters in a cost-cutting direction. The evaluation metrics used for assessing the quality of deblurred images, such as MSE and PSNR, provide quantitative measures to gauge the fidelity of the deblurring process, they have limitations in reflecting perceptual image quality. Using alternative metrics such as, structural similarity (SSIM) and information fidelity criterion (IFC) that consider structural similarities and align with human perception can provide a more comprehensive evaluation of deblurred images. For us, the use of quantitative measures satisfies the purpose according to previous work [11]-[13], [15].
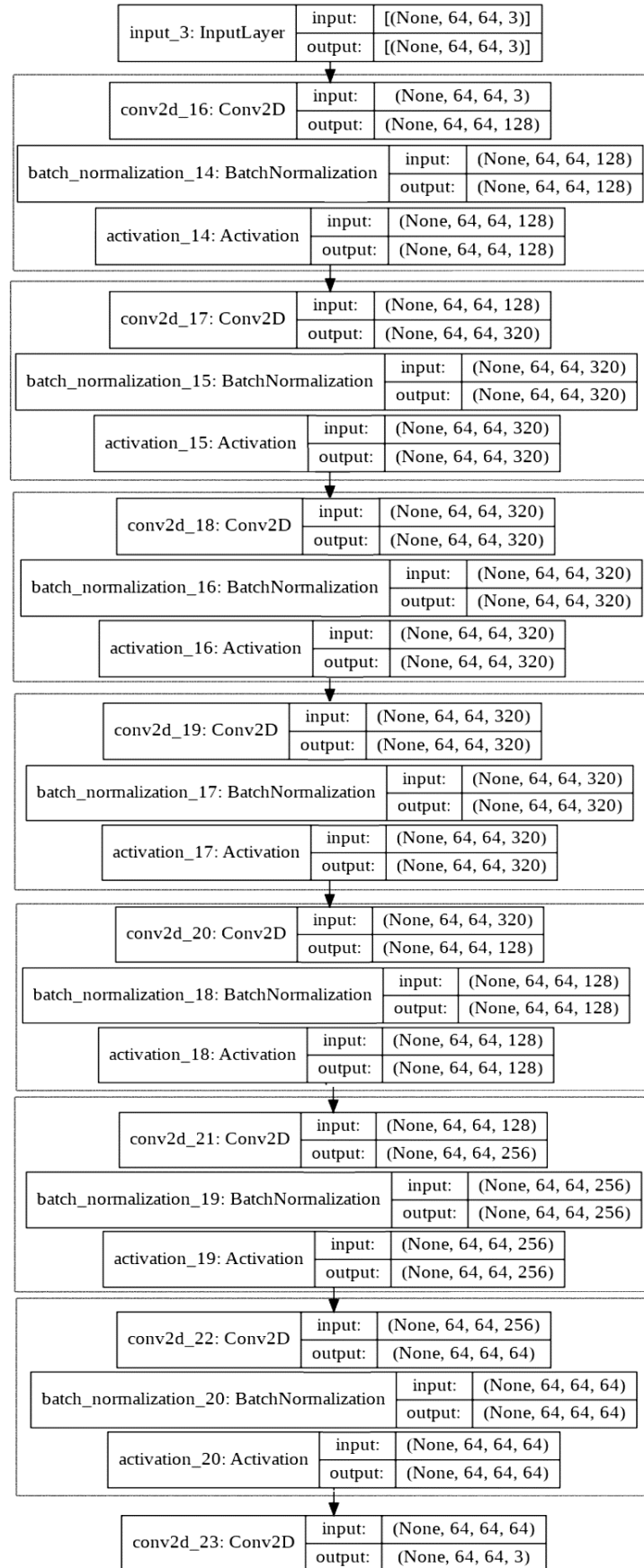
input_3: InputLayer	input:	[(None, 64, 64, 3)]
	output:	[(None, 64, 64, 3)]

conv2d_16: Conv2D	input:	(None, 64, 64, 3)
	output:	(None, 64, 64, 128)

batch_normalization_14: BatchNormalization	input:	(None, 64, 64, 128)
	output:	(None, 64, 64, 128)

activation_14: Activation	input:	(None, 64, 64, 128)
	output:	(None, 64, 64, 128)

conv2d_17: Conv2D	input:	(None, 64, 64, 128)
	output:	(None, 64, 64, 320)

batch_normalization_15: BatchNormalization	input:	(None, 64, 64, 320)
	output:	(None, 64, 64, 320)

activation_15: Activation	input:	(None, 64, 64, 320)
	output:	(None, 64, 64, 320)

conv2d_18: Conv2D	input:	(None, 64, 64, 320)
	output:	(None, 64, 64, 320)

batch_normalization_16: BatchNormalization	input:	(None, 64, 64, 320)
	output:	(None, 64, 64, 320)

activation_16: Activation	input:	(None, 64, 64, 320)
	output:	(None, 64, 64, 320)

conv2d_19: Conv2D	input:	(None, 64, 64, 320)
	output:	(None, 64, 64, 320)

batch_normalization_17: BatchNormalization	input:	(None, 64, 64, 320)
	output:	(None, 64, 64, 320)

activation_17: Activation	input:	(None, 64, 64, 320)
	output:	(None, 64, 64, 320)

conv2d_20: Conv2D	input:	(None, 64, 64, 320)
	output:	(None, 64, 64, 128)

batch_normalization_18: BatchNormalization	input:	(None, 64, 64, 128)
	output:	(None, 64, 64, 128)

activation_18: Activation	input:	(None, 64, 64, 128)
	output:	(None, 64, 64, 128)

conv2d_21: Conv2D	input:	(None, 64, 64, 128)
	output:	(None, 64, 64, 256)

batch_normalization_19: BatchNormalization	input:	(None, 64, 64, 256)
	output:	(None, 64, 64, 256)

activation_19: Activation	input:	(None, 64, 64, 256)
	output:	(None, 64, 64, 256)

conv2d_22: Conv2D	input:	(None, 64, 64, 256)
	output:	(None, 64, 64, 64)

batch_normalization_20: BatchNormalization	input:	(None, 64, 64, 64)
	output:	(None, 64, 64, 64)

activation_20: Activation	input:	(None, 64, 64, 64)
	output:	(None, 64, 64, 64)

conv2d_23: Conv2D	input:	(None, 64, 64, 64)
	output:	(None, 64, 64, 3)

Figure 2. Fully architecture of the proposed approach

3.2. Dataset preprocessing

The UTKFace dataset [27] contains about 20,000 pictures of faces with annotations of age, gender, and ethnicity. It is a large face dataset with a wide age range (from 0 to 116 years old). The stances, facial expressions, lighting, occlusion, resolution, and other aspects of the images vary greatly. This dataset has a wide range of potential applications, such as face detection, age estimation, age regression or progression, landmark placement, and more. The UTKFace dataset is exclusively available for non-commercial research purposes and its copyright belongs to the original proprietors, raising ethical problems. Figure 3 displays a few images from the UTKFace dataset.

Figure 3. Images from the UTKFace dataset

In the first step of UTKFace dataset preparation, we downsized the images from the dataset from their original dimensions to 64×64 pixels. These images are downsized using bicubic interpolation to create blurry image pairs for both training and evaluation. To test the performance of upscaling compressed blurry images, this data set is considered the ground truth dataset. The second stage is the creation of blurry datasets, for which we use a traditional blurring method known as Gaussian blur, also known as Gaussian smoothing, to adjust the blur strength of the images in the dataset. For that, we calculate the blur radius based on the percentage and then simulate additive Gaussian noise for the input images at four diverse noise levels (σ=5, 10, 15, and 20). These noisy input images are used to evaluate the performance. These images need to be compressed into a JPEG file in order to provide compressed images as input to the proposed DCI-CNN. To convert RGB data from the 0-255 scale to the 0-1 scale, we divide each RGB component by the maximum value (255).

For the training step, a set of compressed blurry images consisting of 19,000 samples and their corresponding compressed images (ground truth) are extracted from the preprocessed dataset. The effectiveness of CNN will increase with the number of images utilized for training, and it will then be ready for deblurring compressed images on the foundation of CNN. The effectiveness of CNN will increase with the number of images utilized for training, and it will then be ready for deblurring compressed images on the foundation of CNN. If the dataset is biased or lacks diversity, the model might not generalize well to new, unseen data. Thus, our method is highly resilient to variations in training data and can potentially yield even better results with a larger and more diverse set of training images. During the validating step, we can provide the blurred image as input and receive the deblurred image consisting of 1,000 samples as output. On the other hand, the test set included six benchmark images that had been used in previous study [15]. These images were intentionally blurred to serve as a comparison for the algorithm's performance. One common challenge in evaluating image deblurring models is the limited scope of testing on a small set of images. This may not accurately reflect the diverse scenarios encountered in real-world applications. To address this issue, the use of benchmark images [15] provides a standardized basis for evaluating and comparing the model's performance with established results from prior studies.

4. EXPERIMENTAL RESULTS

In this section, we explain the experiment's outcomes in context with the study questions we established earlier. When using the Adam optimizer, selecting the appropriate learning rate (lr) and number of epochs for training the proposed DCI-CNN requires a series of experiments. We start our model and Adam optimizer with one of the learning rates from the range we picked, and then train the model for a set number of epochs while monitoring the training and validation losses.

Experimentation and iteration are essential to identifying an ideal combination that results in a well-converged model without overfitting. The number of epochs in machine learning is typically determined through empirical observation and experimentation. In our case, we have chosen 1000 epochs after testing

with the loss function (MSE) and accuracy. We start with moderate learning rates and observe the training behavior before making adjustments. After testing with different learning rates, ranging from the smallest (0.00001) to (0.01), we have adopted the latter as the learning rate that offers the best optimization for the model. The results depicted in Figure 4 showcase deblurred images using various learning rates and numbers of epochs. The process involves: Figure 4(a) shows the original high-resolution images. After compression, images may exhibit blurriness or artifacts due to the loss of high-frequency details, as shown in Figure 4(b). In Figure 4(c), deblurring is performed using the Adam optimization algorithm with a learning rate of 10^{-6} and training for 10 epochs to enhance clarity and quality. Similar deblurring is shown in Figure 4(d) and is conducted with an increased learning rate of 0.001 and extended training to 50 epochs, potentially improving image quality. as shown in Figure 4(e), further extension of training to 1000 epochs with a learning rate of 0.001 in the deblurring process using Adam optimization, aiming for refined and higher-quality deblurred images. Figure 4(f) shows that after many experiments, the best learning rate is set to 0.01 and the number of epochs is set to 1000. Monitoring for overfitting or diminishing returns is crucial with prolonged training.

(a)

(b)

(c)

(d)

(e)

(f)
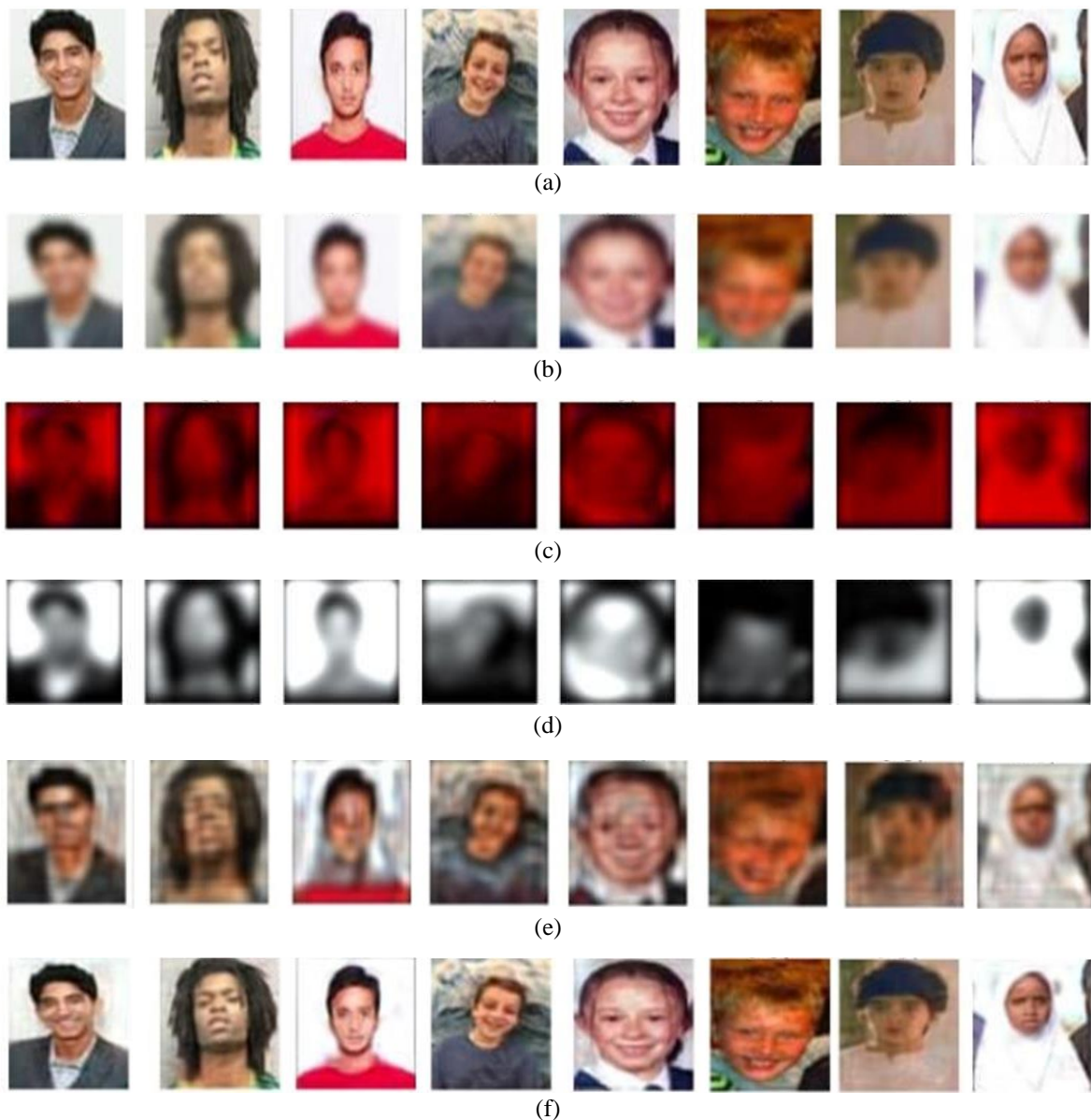
Figure 4. The results of the proposed DCI-CNN method were evaluated using different learning rates and number of epochs; (a) original and compressed images, (b) blurry compressed images, (c) deblurred images for Adam (lr=10-6) and the number of epochs=10, (d) deblurred images for Adam (lr=0.001) and the number of epochs=50, (e) deblurred images for Adam (lr=0.001) and the number of epochs=1000, and (f) deblurred images for Adam (lr=0.01) and the number of epochs=1000

Figure 5 illustrates the training error of algorithms on the preprocessed UTKFace dataset, along with the classification accuracy of the test set. The training error (MSE) and test accuracy reach their minimum error loss value (refer to Figure 5(a)) and achieve the highest classification accuracy of 0.9622 at the 1000th iteration (refer to Figure 5(b)). The figure demonstrates that as the number of iterations increases, the algorithm consistently reduces its error loss.
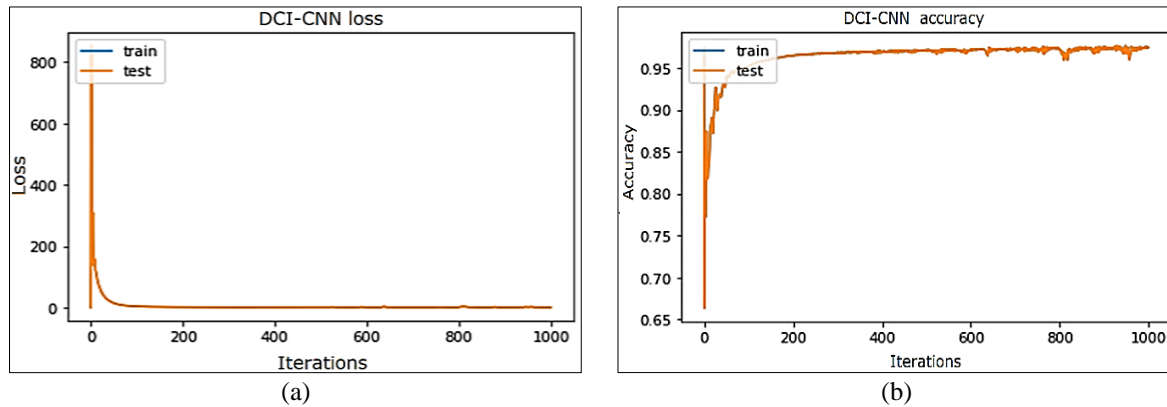

(a)	(b)

Figure 5. Experimental results on the preprocessed UTKFace dataset: (a) MSE loss and (b) accuracy

Table 2 displays the comparison results for various current methods. When compared to existing deblurring methods, the proposed approach using input images of 64×64 pixels yield very strong results. This result is compared to the work of Xu *et al*. [11], Levin *et al*. [12], Krishnan *et al*. [13], and Vijay and Deepa [15]. PSNR and MSE are two performance evaluation metrics used to compare results to the ground truth.

Table 2. A comparison of PSNR/MSE values between existing techniques and the proposed method

Method	Metrics	Image 01	Image 02	Image 03	Image 04	Image 05	Image 06	Average
Proposed DCI-CNN	PSNR	28.7308	30.1127	30.0834	32.2223	31.3071	31.9864	30.7405
	MSE	0.0013	0.0009	0.0009	0.0005	0.0007	0.0006	0.0008
Reshma-Vijay and Deepa [15]	PSNR	12.7375	14.7702	14.5135	12.7715	13.0442	10. 0090	13.5674
	MSE	0.0532	0.0333	0.0354	0.0528	0.0496	0.0998	0.0540
Xu *et al*. [11]	PSNR	7.5690	11.3556	9.4881	6.4341	10.9708	9.5322	9.2250
	MSE	0.2438	0.1151	0.1771	0.3055	0.2004	0.3562	0.2330
Krishnan *et al*. [13]	PSNR	7.3702	10.8856	8.9270	5.9557	10.3725	8.8540	8.7275
Levin *et al*. [12]	PSNR	6.3370	8.9217	8.1277	5.8330	9.9275	8.3013	7.9080

The experimental results show that the proposed approach surpasses existing methods in both PSNR and MSE metrics for each benchmark image subjected to image deblurring. Furthermore, compared to existing techniques, the proposed approach exhibits significantly reduced processing time for the entire image deblurring process, enhancing overall efficiency. Figure 6 visually presents the deblurring results of benchmark images using the proposed DCI-CNN. The figures illustrate the following: Figure 6(a) displays the original images, representing the high-resolution, uncompressed versions before any processing or alterations. In Figure 6(b), the images are presented after undergoing a process that introduces blurriness. This blurriness can be a result of compression or other factors, indicating a degradation in image quality compared to the original versions. On the other hand, the output image of the proposed method is shown in Figure 6(c).

To determine the effect of input image size on our proposed approach, we applied the same model to different sizes of the input blurry image for 32×32, 64×64, 128×128, and 256×256 pixels. The PSNR and MSE of the proposed method with varying sizes of blurry input images are presented in Table 3. We notice that as the size of the input blurred image was reduced, the PSNRs increased. When the size of the input blurry image was reduced in this situation, the MSE of the images dropped as well. As a consequence, the best results are produced when the proposed method is applied to the size of the input blurry image of 32×32 pixels.

Image 01 Image 02 Image 03 Image 04 Image 05 Image 06
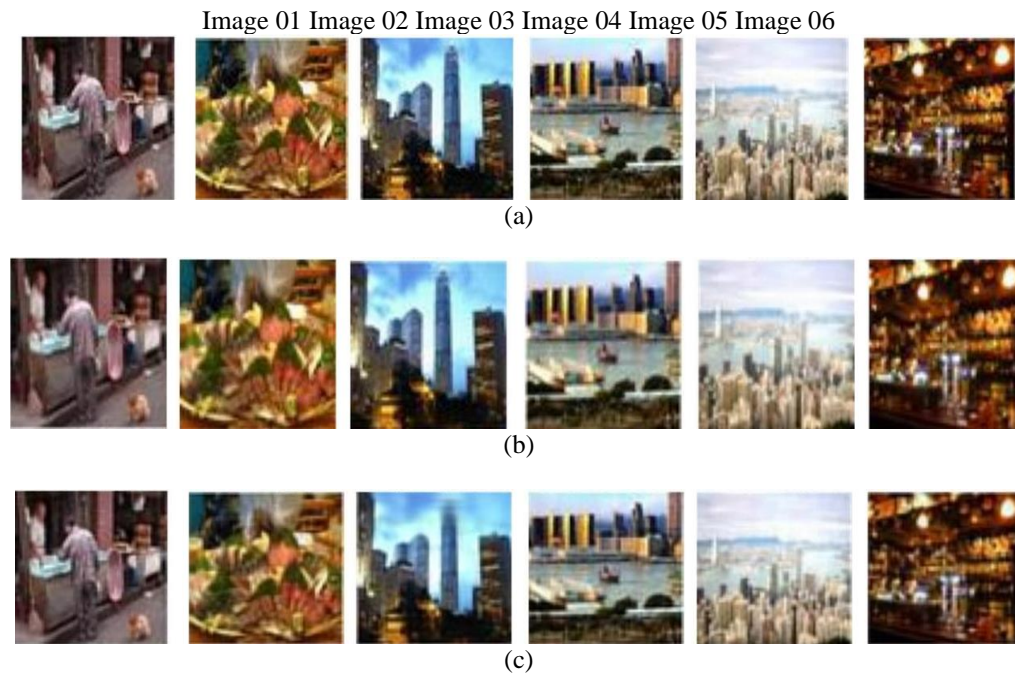

(a)

(b)

(c)

Figure 6. The proposed DCI-CNN method's results using benchmark images; (a) original images, (b) blurry images, and (c) proposed method output

Table 3. Comparison of the PSNRs and MSEs of the proposed method (DCI-CNN) for various sizes of input blurry images

Proposed method	Metrics	Image 01	Image 02	Image 03	Image 04	Image 05	Image 06	Average
DCI-CNN for 32×32 pixels	PSNR	33.5070	34.7434	34.3013	32.7963	33.3723	33.6188	33.7232
	MSE	0.0004	0.0003	0.0003	0.0005	0.0004	0.0004	0.0004
DCI-CNN for 64×64 pixels	PSNR	28.7308	30.1127	30.0834	32.2223	31.3071	31.9864	30.7405
	MSE	0.0013	0.0009	0.0009	0.0005	0.0006	0.0006	0.0008
DCI-CNN for 128×128 pixels	PSNR	18.4141	20.4095	23.4579	24.1465	21.3766	24.8268	22.1052
	MSE	0.0144	0.0091	0.0045	0.0038	0.0072	0.0032	0.0070
DCI-CNN for 256×256 pixels	PSNR	19.0524	22.1578	19.7615	15.2429	18.8517	18.1343	18.8668
	MSE	0.0124	0.0060	0.0105	0.0299	0.0130	0.0153	0.0145

In addition, we proposed the use of a deep DCI-CNN for deblurring compressed and normal images, a model that has been experimentally proven to be superior and yields convincing and competitive results when compared to non-deep learning methods and deep methods seen in related works. On the other hand, the proposed method demonstrates its effectiveness in both blurry compressed images and blurry images. To comprehensively assess the practical utility of DCI-CNN, it is important to focus on examining the method's real-world performance in crucial applications such as medical imaging, surveillance, or satellite imagery. This will help to understand its contextual effectiveness. Additionally, it is necessary to evaluate the method's scalability in handling diverse image sizes and qualities, taking into consideration the challenges posed by real-world scenarios. Furthermore, it is crucial to investigate the challenges related to deploying the algorithm in different industries, including considerations such as computational resources, integration with existing systems, and adaptability to diverse image characteristics.

Finally, the study does not explicitly address the feasibility of implementing the DCI-CNN model in real-time. The ability to be applied in real-time is a crucial factor to consider for practical use in image deblurring tasks. The successful implementation of a deep learning model, such as DCI-CNN, in real-time scenarios depends on various factors, including computational efficiency, model complexity, and hardware limitations. Without specific information on these aspects, it is uncertain whether the DCI-CNN model is suitable for real-time applications. On the other hand, this study does not address the scalability of the DCI-CNN model to handle larger or higher-resolution images, which are common in real-world applications. Therefore, in order to address this limitation, we will focus on working with patches of images instead of the entire image. This approach will allow us to effectively handle larger and higher-resolution images.

5.   CONCLUSION

In this paper, we used a novel strategy based on CNNs to decrease image compression artifacts; particularly, we proposed an efficient and principled CNN structure for this task, which enabled us to deblur these images. The purpose of our proposed deblurring method (DCI-CNN) is to produce images with more detail from blurry input images than generic approaches and other deep methods seen in related research. In this paper, we aim to introduce a new deep learning strategy that utilizes CNN at multiple convolutional layers. Our results demonstrate the success of this approach, indicating that it can be a valuable pre-processing step for a range of computer vision tasks. This is especially useful in situations where images have been heavily compressed, rendering current state-of-the-art algorithms ineffective.

Our proposed CNN-based solution improves performance by minimizing the artifacts generated by different compression algorithms (in our case, the reduction of JPEG artifacts). By addressing the problem of the ideal CNN model to deal with artefact deblurring, we discovered more than a model and structure, and we assume it will be challenging to find the best, despite the fact that our model shown a significant superiority compared to comparable research. Inspired by CNN, we will investigate if recurrent neural networks (RNN) or generative antagonist networks (GAN) can assist in improving image deblurring or restoration performance.

REFERENCES

[1]   J. -B. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, 2015, pp. 5197-5206, doi: 10.1109/CVPR.2015.7299156.

[2]   Z. Cui, H. Chang, S. Shan, B. Zhong, and X. Chen, "Deep Network Cascade for Image Super-resolution," *Computer Vision– ECCV 2014: 13th European Conference, Zurich, Switzerland, Proceedings, Part V 13*, pp. 49–64, Sep. 2014, doi: 10.1007/978-3-319-10602-1_4.

[3]   C. Dong, C. C. Loy, K. He, and X. Tang, "Image Super-Resolution Using Deep Convolutional Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 295–307, Feb. 2016, doi: 10.1109/tpami.2015.2439281.

[4]   J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image Super-Resolution Via Sparse Representation," *IEEE Transactions on Image Processing*, vol. 19, no. 11, pp. 2861–2873, Nov. 2010, doi: 10.1109/tip.2010.2050625.

[5]   J. Yang, Z. Wang, Z. Lin, S. Cohen, and T. Huang, "Coupled Dictionary Training for Image Super-Resolution," *IEEE Transactions on Image Processing*, vol. 21, no. 8, pp. 3467–3478, Aug. 2012, doi: 10.1109/tip.2012.2192127.

[6]   C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a Deep Convolutional Network for Image Super-Resolution," *Computer Vision – ECCV 2014*, pp. 184–199, 2014, doi: 10.1007/978-3-319-10593-2_13.

[7]   D. Liu, Z. Wang, B. Wen, J. Yang, W. Han, and T. S. Huang, "Robust Single Image Super-Resolution via Deep Networks with Sparse Prior," *IEEE Transactions on Image Processing*, vol. 25, no. 7, pp. 3194–3207, Jul. 2016, doi: 10.1109/tip.2016.2564643.

[8]   J. Kim, J. K. Lee, and K. M. Lee, "Accurate Image Super-Resolution Using Very Deep Convolutional Networks*," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, pp. 1646-1654, doi: 10.1109/CVPR.2016.182.

[9]   L. -F. Dong, Y. -Z. Gan, X. -L. Mao, Y. -B. Yang, and C. Shen, "Learning Deep Representations Using Convolutional Auto-Encoders with Symmetric Skip Connections," *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, AB, Canada, 2018, pp. 3006-3010, doi: 10.1109/ICASSP.2018.8462085.

[10]   H. Zhang, Y. Li, H. Chen, C. Gong, Z. Bai, and C. Shen, "Memory-efficient hierarchical neural architecture search for image restoration," *International Journal of Computer Vision*, vol. 130, no. 1, pp. 157–178, Nov. 2021, doi: 10.1007/s11263-021-01537-w.

[11]   L. Xu, J. S. Ren, C. Liu, and J. Jia, "Deep Convolutional Neural Network for Image Deconvolution," *Neural Information Processing Systems*, vol. 27, pp. 1790-1798, 2014.

[12]   A. Levin, Y. Weiss, F. Durand, and W. T. Freeman, "Understanding Blind Deconvolution Algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 12, pp. 2354–2367, Dec. 2011, doi: 10.1109/tpami.2011.148.

[13]   D. Krishnan, T. Tay, and R. Fergus, "Blind deconvolution using a normalized sparsity measure, " *CVPR 2011*, Colorado Springs, CO, USA, 2011, pp. 233-240, doi: 10.1109/CVPR.2011.5995521.

[14]   Y. Bai, H. Jia, M. Jiang, X. Liu, X. Xie, and W. Gao, "Single-Image Blind Deblurring Using Multi-Scale Latent Structure Prior," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 7, pp. 2033-2045, Jul. 2020, doi: 10.1109/TCSVT.2019.2919159.

[15]   V. J. R. Vijay and P. L. Deepa, "Image Deblurring Using Convolutional Neural Network," *IOSR Journal of Electronics and Communication Engineering*, vol. 11, no. 5, pp. 7–12, 2016, doi: 10.9790/2834-1105020712.

[16]   Y. Liu, Z. Sheng, and H.-L. Shen, "Guided Image Deblurring by Deep Multi-Modal Image Fusion," *IEEE Access*, vol. 10, pp. 130708–130718, 2022, doi: 10.1109/access.2022.3229056.

[17]   C. J. Schuler, M. Hirsch, S. Harmeling, and B. Scholkopf, "Learning to Deblur," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 7, pp. 1439–1451, Jul. 2016, doi: 10.1109/tpami.2015.2481418.

[18]   A. Chakrabarti, "A Neural Approach to Blind Motion Deblurring," *European Conference on Computer Vision (ECCV 2016)*, pp. 221–235, Oct. 2016, doi: 10.1007/978-3-319-46487-9_14.

[19]   J. Sun, W. Cao, Z. Xu, and J. Ponce, "Learning a convolutional neural network for non-uniform motion blur removal," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, 2015, pp. 769-777, doi: 10.1109/CVPR.2015.7298677.

[20]   O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin, and J. Matas, "DeblurGAN: Blind Motion Deblurring Using Conditional Adversarial Networks," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 2018, pp. 8183-8192, doi: 10.1109/CVPR.2018.00854.

[21]   Y. Chen, Y. Zhao, W. Jia, L. Cao, and X. Liu, "Adversarial-learning-based image-to-image transformation: A survey," *Neurocomputing*, vol. 411, pp. 468–486, Oct. 2020, doi: 10.1016/j.neucom.2020.06.067.

[22]   S. Nah, T. H. Kim, and K. M. Lee, "Deep Multi-scale Convolutional Neural Network for Dynamic Scene Deblurring," *2017 IEEE*

       Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017, pp. 257-265, doi: 10.1109/CVPR.2017.35.

[23] X. Tao, H. Gao, X. Shen, J. Wang and J. Jia, "Scale-Recurrent Network for Deep Image Deblurring," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recog25nition*, Salt Lake City, UT, USA, 2018, pp. 8174-8182, doi: 10.1109/CVPR.2018.00853.

[24] S. Su, M. Delbracio, J. Wang, G. Sapiro, W. Heidrich, and O. Wang, "Deep Video Deblurring for Hand-Held Cameras," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 2017, pp. 237-246, doi: 10.1109/CVPR.2017.33.

[25] S. Júnior, "Evaluating Deep Learning Techniques for Blind Image Super-Resolution within a High-Scale Multi-Domain Perspective," *AI*, vol. 4, no. 3, pp. 598–619, Aug. 2023, doi: 10.3390/ai4030032.

[26] D. P. Kingma and J. Ba, "ADAM: a method for stochastic optimization," *arXiv*, 2014, doi: 10.48550/arXiv.1412.6980.

[27] Z. Zhang, Y. Song, and H. Qi, "Age Progression/Regression by Conditional Adversarial Autoencoder," *2017 IEEE Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 2017, pp. 4352-4360, doi: 10.1109/cvpr.2017.463.

BIOGRAPHIES OF AUTHORS

Rafik Menassel ⓘ 🗗 SC ↻ was born in Algeria. He received his B.S. degree in Computer Science from University of Mentouri-Constantine (Algeria) in 2006, M.S. degree in Computer Science "Information and knowledge systems" from Echahid Cheikh Larbi Tebessi University, Tebessa (Algeria) in 2009 and he received a Ph.D. in computer science from Badji Mokhtar-University, Annaba, Algeria in 2018. Following his Ph.D., he worked as an Associate professor in the department of Mathematics and Computer Science, University of Tebessa, Tebessa, Algeria, where he did research in the areas of biometrics, writer identification, image processing, image compression, and document analysis. He has published various papers in the general areas. He can be contacted at email: r.menassel@univ-tebessa.dz.

Abdeljalil Gattal ⓘ 🗗 SC ↻ was born in Algeria. He received his B.S. degree in Computer Science from University of Skikda (Algeria) in 2004, M.S. degree in Computer Science "Information and knowledge systems" from Abbes Laghrour University of Khenchela (Algeria) in 2009 and he received his Ph.D. in 2016 from Ecole nationale Supérieure d'Informatique (ESI-Algeria) in Computer Science and focuses in Segmentation-Verification for Handwritten Digit Recognition. Currently, he is working as Associate Professor at the Department of Mathematics and Computer Science in University of Tebessa (Algeria). He supervised many Master and License students. He has published a number of papers. In addition, he has collaborated as a member on several research projects and also participated in several scientific competitions. His research interests include image analysis, pattern recognition, and recognition of handwriting. He can be contacted at email: abdeljalil.gattal@univ-tebessa.dz.

Fateh Kerdoud ⓘ 🗗 SC ↻ was born in Algeria. He received his Master degree in Systems and Multimedia from University of Tebessa, Tebessa, Algeria. He can be contacted at email: kerdoud.fateh.001@gmail.com.