



Deep Learning-Based Noise Type Classification and Removal for Drone Image Restoration

Waqar Ahmed² · Sajid Khan^{1,2} · Adeeb Noor³ · Ghulam Mujtaba²

Received: 31 March 2023 / Accepted: 1 October 2023 / Published online: 1 November 2023
© King Fahd University of Petroleum & Minerals 2023

Abstract

Recent advancements in deep learning have enabled significant progress in image noise type classification and denoising systems. Researchers working on deep learning-based image multi-type denoising either use a single-stage or a two-stage denoising approach. The single-stage approach proposes designing a single denoising autoencoder (DAE), whereas the two-stage approach first classifies the noise type, followed by applying a noise-specific filter. The problem with the single-stage approach is that a generalized DAE fails to be effective. Two-stage approaches work on a limited number of noise types, as researchers typically address only two or three noise types. This paper proposes a framework for two-stage multi-type image denoising that provides classification and denoising of four types of noise with a per-class classification accuracy of 98.2–100% and a denoising technique that obtained promising PSNR and SSIM values for various types of noise, ensuring effective image restoration. The proposed methodology can be applied to any field that requires image denoising without prior knowledge of the type of noise.

Keywords Autoencoders · Convolutional neural networks · Deep learning · Drones · Noise classification · Denoising

1 Introduction

In recent years, significant advancements in machine learning and deep learning have revolutionized various industries and applications, with researchers exploring novel algorithms inspired by diverse fields, such as alpine skiing, coronavirus mask protection, and gravitational search strategies [1–4]. In image processing, photogrammetry, aerospace, particle

physics, geology, medicine, and materials science are just a few intriguing applications [5]. Most of the image processing algorithms are applied as preprocessing steps before localization [6–10], involving image enhancement before feature extraction [11], or the application of deep learning before diagnosis [12].

Image denoising is a hot topic of interest for researchers working in image processing. Image denoising techniques aim to eliminate noise and generate a clean image from the noisy image [13]. Additionally, they reduce noise while preserving an image's key characteristics. Noise in images is a common problem in this digital era, typically caused by sensor issues in cameras or during image transmission from one source to another. Various types of noise, including Gaussian noise, as well as others such as Salt and Pepper noise, Speckle noise, and Poisson noise, can be observed in images, among many others [14–16].

Conventionally, researchers working in image denoising [16–21] often assume prior knowledge of the noise type and design algorithms accordingly. In Yin et al. [17], the authors introduce a novel image filtering technique called Side Window Filtering, which leverages non-local image information to better preserve image edges while removing noise. El Helou and Süsstrunk [18] propose a blind universal Bayesian

✉ Adeeb Noor
arnoor@kau.edu.sa

Waqar Ahmed
waqar.mscsf19@iba-suk.edu.pk

Sajid Khan
sajidkhan@iba-suk.edu.pk

Ghulam Mujtaba
mujtaba@iba-suk.edu.pk

¹ Department of Computer Science, School of Engineering, Central Asian University, Tashkent, Uzbekistan

² Department of Computer Science, Sukkur IBA University, Sukkur, Pakistan

³ Department of Information Technology, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 80221, Saudi Arabia



image denoising algorithm capable of learning the noise level from a single noisy image and performing denoising. Mahaoui et al. [19] introduce an image denoising method based on compressive sensing and regularization constraints. This method utilizes a sparse representation of the image and incorporates prior knowledge about the image's structure through a regularization term to achieve noise reduction. Mao et al. [14] and Xu et al. [13] also focus on image denoising using sparse representation. Sparse representation-based image denoising may face challenges when dealing with images containing structured noise since the dictionary might not adequately capture the noise structure, potentially resulting in incomplete denoising results.

Thanh and Enginoğlu [20] propose an iterative mean filter for image denoising. The method applies a mean filter iteratively, adapting the filter kernel based on the image content and noise level. The challenge with such approaches is that they require prior knowledge of the noise type. However, in many cases, the noise type cannot be determined in advance. Therefore, it becomes essential to predict the type of noise before applying image denoising.

In recent years, researchers have increasingly focused on convolutional neural networks (CNNs), which have consistently outperformed many conventional image processing techniques. These techniques can serve as pre-processing, post-processing, or even as the primary processing step to enhance image processing performance, computer vision applications, and methods related to image recognition [21]. Deep learning has garnered significant attention from researchers due to its ability to surpass the performance of classical methods. Specifically, in the realm of image classification, deep learning plays a pivotal role in learning how to classify images based on their features and patterns. Deep learning is also widely applied to several trending topics, including face identification, object detection, and various forms of image recognition [22]. Recognizing the potential of deep learning, a select group of researchers have focused on noise reduction as a pre-processing step for image denoising in tasks such as classification [23, 24], segmentation [25], and object detection [26].

This paper proposes a two-stage framework that classifies the noise type followed by denoising based on the predicted noise type. In the first stage, a Convolutional Neural Network (CNN)-based classifier is utilized. The CNN is trained to classify the noisy image into four classes: Gaussian, Salt & Pepper, Poisson, and Speckle. For the second stage of denoising, four separate denoising autoencoders (DAEs) have been trained to handle each classified case using its specialized DAE. Lastly, the performance of the proposed framework is evaluated at different stages of the experiment by calculating the peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM).

This paper is divided into five sections. Section 1 provides an overview of image noise and denoising. Section 2 discusses related work. The proposed framework architecture for image noise type classification and image denoising, which is based on a two-stage network, is discussed in Sect. 3. Section 4 describes the proposed approach. Section 5 depicts the results and discussions. Section 6 contains the conclusion, which includes future perspectives on the proposed framework.

2 Related Work

Momeny et al. [24] proposed a convolutional neural network named Noise-Robust Convolutional Neural Network (NR-CNN). They introduced a noise map layer that identifies pixels corrupted by impulse noise. This layer also detects tempered, missing, damaged, and lost pixels. Instead of denoising the identified noisy pixels, their network excludes these pixels during classification. When trained on noisy data, the proposed NR-CNN outperformed well-known models, such as VGG-Net Medium, VGG-Net Slow, GoogleNet, and ResNet. Their work underscores the significance of methods that address noisy pixels. However, disregarding noisy pixels may not be ideal, as these pixels might conceal features distinguishing one object from another. For instance, salt noise on a digit “eight” might cause it to resemble a digit “nine” if ignored. Therefore, instead of exclusion, restoring the noisy pixels may be a more favorable approach.

In another paper, Koziarski et al. [27] conducted a comprehensive experimental analysis of the influence of noise on deep neural network classification. They explored various approaches, including collecting noisy samples to train the classifier effectively on both noisy and non-noisy data. However, they noted that collecting such samples could be costly. As an alternative, they attempted to introduce noise to the existing samples. They concluded by incorporating CNN-based denoising support to enable classification to work seamlessly on both noisy and non-noisy images.

Liu et al. [28] employed a deep neural network to classify and denoise image noise. Their training dataset was a combination of CIFAR-10, BSD500, and MINST. In comparison to existing methods, they showcased that their denoising network achieved higher PSNR and SSIM values, while their classification network attained greater accuracy. They introduced a denoising approach using their CNN-based model, wherein they predicted the noisy pixels and restored each pixel from the original noisy image.

In another paper, Tripathi [29] showed that traditional denoising methods are hampered by their time-consuming and rigid nature. To tackle these issues, the study presented a novel strategy that integrated image classification and denoising. This involves the design, implementation, and



assessment of a model architecture centered around convolutional neural networks (CNN) and UNET. The methodology is applied to a dataset of facial images used for training, validation, and testing. During preprocessing, the images are resized to 48×48 dimensions, normalized, and subjected to various noise types.

Importantly, preprocessing techniques slightly differ for each model. The CNN model demonstrates remarkable training and validation accuracy, reaching 99.87% and 99.92%, respectively. The UNET model also excels in achieving optimal Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) values across different noise scenarios. This study contributes to the field by offering an innovative approach that improves both image classification and denoising tasks for grayscale images.

Joon et al. [30] introduced a deep convolutional neural network method for Gaussian noise detection and estimation of noise levels. The proposed CNN model accurately detects the presence and level of Gaussian noise in images. They utilized 12,000 standard test images for testing and 3000 for training. The overall accuracy in classifying the ten noise level classes was 74.74%. Notably, this study did not introduce any denoising technique.

Roy et al. [31] proposed a deep-learning approach for noisy image classification. In this paper, they first used autoencoders (AEs) to reconstruct the noisy images and then classified the reconstructed images. Denoising autoencoders (DAE), convolutional denoising autoencoders (CDAE), two hybrid AEs (DAE-CDAE and DVAE-CDAE), and the denoising variational autoencoder (DVAE) were also used in the denoising step. This study analyzed five hybrid models to classify noisy images: DAE-CNN, CDAE-CNN, DVAE-CNN, DAE-CDAE-CNN, and DVAE-CDAE-CNN. They conducted their research using MNIST and CIFAR-10 datasets.

Table 1 summarizes the contributions of all the mentioned works in this section. When creating the table, multiple factors were considered, including noise classification, denoising, noise types, color and high-resolution images, and noise type classification and denoising methods from the related studies. Notably, our findings indicate a gap in the literature, as there is no existing study where image noise classification is followed by image denoising for colorful high-resolution images using Denoising Autoencoders (DAEs). There is a pressing need to denoise color images for practical applications, including pre-processing drone images for object identification, detection, and segmentation.

Furthermore, it's worth noting that most authors have traditionally treated image noise classification and denoising as separate problems, often working with grayscale or low-resolution images due to computational constraints and data availability. Some authors attempted multi-type denoising by

employing generalized deep learning networks for all types of noise. However, such a one-size-fits-all approach may not yield optimal results compared to using specialized autoencoders for each noise type. Given that the image noise type is often unknown in advance, the approach of noise type classification followed by denoising holds great potential. It can facilitate various tasks, including image recognition, object detection, segmentation, and more.

3 Proposed Framework Architecture

This section provides a concise description of the proposed framework architecture. To enhance understanding, we have created two different figures illustrating the architecture. Figure 1 illustrates the basic operation of the framework for 32×32 pixel images, while the second figure demonstrates how our framework handles high-resolution drone-captured images. Figure 2 incorporates additional methods for processing high-resolution drone-captured images within the framework. Our framework has been efficiently trained and tested on an extensive dataset of 32×32 pixel images. Moreover, it can be adapted to work with high-resolution images by incorporating a few additional steps. Additionally, the framework has been tested with drone-captured images, showcasing a practical application of image noise type classification and denoising.

For a general understanding, Fig. 1 outlines the framework architecture's workflow. It begins by taking the noisy image as input. The first network is responsible for detecting the noise type in the image, while the second network loads the specialized denoising autoencoder (DAE) based on the result from the first network to denoise the image. Figure 1 provides an overview of the framework's overall operation.

In Figs. 1 and 2, Network 1, named “Image Noise Type Classification,” utilizes a Convolutional Neural Network (CNN) to classify the type of image noise. CNNs are employed due to their capacity to deeply learn hidden patterns and features within images, which often outperforms classical methods in various classification tasks.

Network 2, named “Denoising Autoencoders,” consists of four specialized DAE models: Gaussian DAE, Salt & Pepper DAE, Speckle DAE, and Poisson DAE. DAEs are chosen for their ability to reconstruct the output and eliminate noise from noisy input, effectively restoring it to its original state. This performance surpasses that of traditional denoising techniques in many cases.

Specialized DAEs have been developed for each noise type. We conducted multiple experiments comparing generalized DAEs to specialized DAEs to assess their performance based on PSNR and SSIM. Our findings indicate that specialized DAEs designed for specific image noise types consistently outperform generalized DAEs for a wide range



Table 1 Comparison of related studies

Reference, authors, and paper title	Noise type classification	Denoising	Noise types	Color & high-resolution images	Methods
[24] Momeny, M., Latif, A., Agha Sarram, M., Sheikhpour, R., Zhang, Y.: “A noise robust convolutional neural network for image classification”	Yes	No	1 noise type (Impulse)	Color: Yes, high-resolution: Yes	Classification: NR-CNN, Denoising: None
[27] Kozianski, M., Cyganek, B.: “Image recognition with deep neural networks in presence of noise-dealing with and taking advantage of distortions”	Yes	Yes	2 noise types (Gaussian, Salt & Pepper)	Color: Yes, high-resolution: No	Classification: CNN, Denoising: DNN
[28] Liu, F., Song, Q., Jin, G.: “The classification and denoising of image noise based on deep neural networks”	Yes	Yes	4 noise types (Gaussian, Salt & Pepper, Speckle, Poisson), and mixed types	Color: Yes, high-resolution: No	Classification: CNN, Denoising: CNN
[29] M. Tripathi, “Facial image noise classification and denoising using neural network”	Yes	Yes	3 noise types (Gaussian, Salt & Pepper, Poisson)	Color: No, high-resolution: No	Classification: CNN, Denoising: UNET
[30] Chuah, J.H., Khaw, H.Y., Soon, F.C., Chow, C.O.: “Detection of gaussian noise and its level using deep convolutional neural network”	Yes	No	1 noise type (Gaussian)	Color: No, high-resolution: No	Classification: CNN, Denoising: None
[31] Roy, S., Ahmed, M., Akhand, M.: “Noisy image classification using hybrid deep learning methods.”	Yes	Yes	1 noise type (Gaussian)	Color: No, high-resolution: No	Classification: CNN, Denoising: DAEs



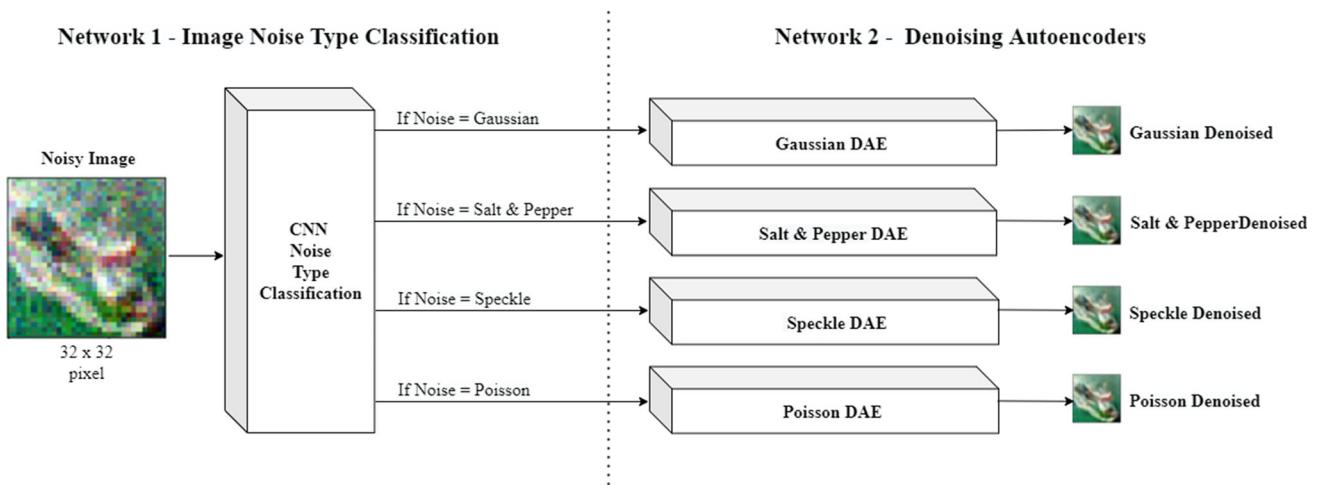


Fig. 1 Proposed framework architecture

of images. Further details and comparative results are presented in Sect. 5.

We have incorporated several additional methods into the framework at various stages to handle high-resolution drone-captured images. These methods are activated when the framework receives a high-resolution image as input. In Fig. 2, the methods “Image sampling method,” “Prediction noise result method,” “Noisy image cropping method,” and “Denoised images joiner method” serve distinct purposes, all of which are elaborated on in the Proposed Approach section.

For a fundamental understanding, these methods are responsible for overseeing a comprehensive process that involves cropping a high-resolution image into 32×32 pixel sections, sequentially passing them through the networks, and finally reassembling them to their original resolution as a single output.

Our framework introduces a novel approach by seamlessly integrating image noise type classification and denoising tasks into a unified system. Unlike conventional methods, we treat these tasks individually during the initial stages, ensuring dedicated attention and optimization for each task. This unique approach enables us to achieve improved performance in both classification accuracy and denoising effectiveness. Our investigations have revealed that the one-stage approach of deep learning denoising is not an effective solution, as evident from our findings.

Furthermore, in the context of previous studies on image noise type classification, it is a common practice to scale down the images to a lower resolution to reduce computational complexity and memory requirements. However, we have recognized the potential limitations of this approach, particularly in scenarios where the noise patterns may not be effectively captured at lower resolutions, resulting in reduced classification accuracy.

To address this concern and improve the accuracy of noise type classification, we have incorporated additional methods into our framework. Rather than relying solely on scaled-down images, we have introduced a novel strategy: extracting multiple samples from each input image. More precisely, we crop four samples from the corners and one sample from the center of the image. This technique enables us to capture diverse noise patterns and variations present in different regions of the image, ensuring a more comprehensive representation for classification.

By incorporating these additional methods, we observed a significant improvement in the accuracy of our classification results, rendering our approach more robust and precise in handling various noise types. We are confident that this innovative modification represents a significant contribution to the novelty of our proposed framework and sets it apart from previous studies that rely solely on scaled-down images for noise-type classification.

4 Proposed Approach

This section provides a clarification of the proposed framework as shown in Fig. 2. We approached and connected the two tasks individually, ensuring that the framework functions as a cohesive system encompassing image noise type classification and denoising. Since classification and denoising represent distinct tasks, we handled them separately in the initial stages, encompassing dataset creation, pre-processing, training, testing, and implementation.

Rather than developing the framework using a high-resolution image dataset, we opted to build it using a lower-resolution CIFAR-100 image dataset. This choice was driven by two key considerations. Firstly, training models with high-resolution images demands substantial processing



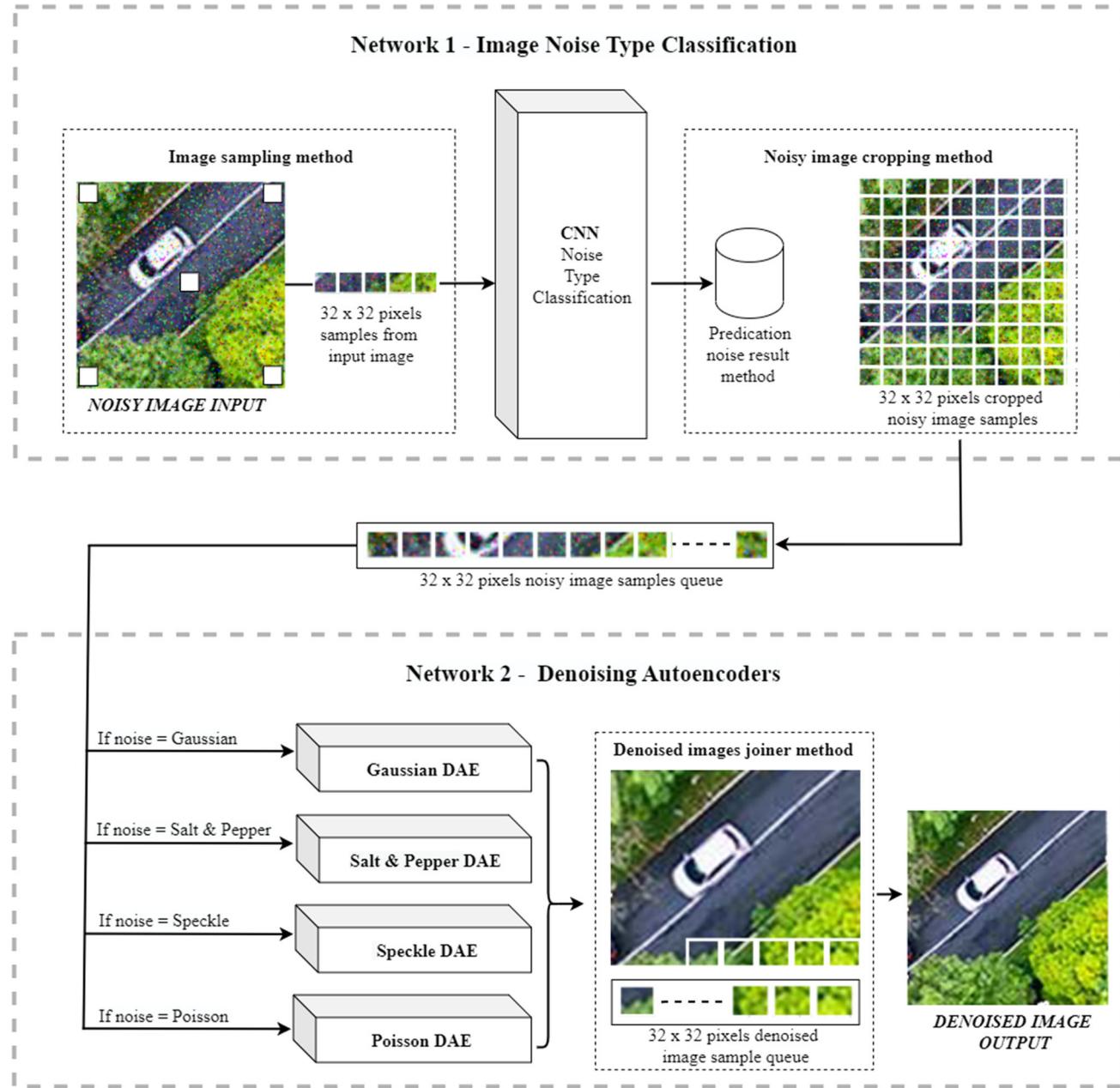


Fig. 2 Proposed framework architecture with additional methods for high-resolution images

power and time. Secondly, high-resolution datasets typically offer fewer images and classes. Consequently, we constructed the framework using the CIFAR-100 dataset. Additionally, we incorporated a few supplementary steps to enable our framework to handle high-resolution drone-captured images effectively.

In the subsequent sections, we delve into the development of the first network based on convolutional neural network (CNN) for image noise type classification. We also describe the second network, which employs specific denoising autoencoders (DAEs) for denoising particular types of

noise, along with additional methods for processing high-resolution digital images within the framework.

In this section, we also present our proposed approach, complemented by convenient access to the research code GitHub repository [32] accompanied by comprehensive documentation. This repository facilitates the seamless implementation of the deep learning-based noise type classification and denoising techniques delineated in this paper. This resource holds significant potential to aid future research endeavors and foster advancements in this domain.

4.1 Datasets

For image noise type classification and denoising, we employed well-known, publicly available datasets, including CIFAR-10, CIFAR-100, BSD500, and a set of randomly collected drone-captured images from the internet. Both CIFAR-10 and CIFAR-100 datasets consist of 60,000 vibrant images, with the main distinction being that CIFAR-100 contains 100 classes, each with 600 images, while CIFAR-10 comprises ten classes, each with 6000 images. All images in these datasets are of size 32×32 pixels.

The BSD500 dataset contains random natural high-resolution images, while we gathered more than 100 random drone-captured images from the internet for practical applications. In our approach, we utilized the CIFAR-100 dataset for training and validation purposes, while CIFAR-10 was employed for testing. Additionally, we used the BSD500 dataset and the self-collected drone-captured image dataset to assess and validate the performance of our trained classification and denoising networks on high-resolution images.

4.2 Datasets Creation and Pre-processing for Image Noise Type Classification

To construct the dataset for the first network, we utilized MATLAB to introduce four different types of noise—Gaussian, Salt & Pepper, Speckle, and Poisson—into 60,000 images from the CIFAR-100 dataset, ensuring an equal distribution of noise levels. Figure 3 provides a visual representation of these image noise types. This process yielded a total of 240,000 images, with 60,000 images allocated to each noise class for image noise type classification.

Subsequently, we divided the image data into an 80:20 ratio for training and validation purposes in the Convolutional Neural Network (CNN). For each class, we had 48,000 images for training and 12,000 images for validation. In the subsequent step, we transformed the data into categorical labels, categorizing them into four classes to align with the network's design for classifying four noise types. It is essential to inform the CNN model that the data is categorical. The images from the directory were converted into Python

arrays, shuffled, and stored as NumPy arrays for training and testing.

4.3 Implementing CNN

We employed TensorFlow to implement a convolutional neural network (CNN) for our tasks. In the construction and implementation of the CNN, we utilized six two-dimensional convolutional layers, each employing a 'relu' activation function. These layers were designed to learn patterns and features from the input images and classify the noise type. To streamline the network's computation and reduce parameter count, a two-dimensional max-pooling layer was placed after each CNN layer.

The hyperparameters of these layers were fine-tuned with consideration for the network's accuracy and size, aiming to create an effective yet compact model. The model's architecture is visually represented in Fig. 4. The final two layers consist of dense (fully connected) layers, with a dropout layer following the first dense layer to prevent overfitting and enhance the model's generalization. The last layer's output nodes were set to four, corresponding to the network's four classes, and used a 'softmax' activation function for categorical classification.

Following 30 epochs of training, the model achieved an accuracy of 98%. Training and validation accuracy are depicted in Fig. 5, while Fig. 6 illustrates the training and validation loss.

4.4 Dataset Creation and Pre-processing for Denoising

To construct the dataset for denoising autoencoders (DAEs), we utilized the Scikit-image library in Python to assess network performance. The nature of noise in an image can vary, depending on how the noise is introduced. We introduced four different types of noise, each with a 10% noise factor Gaussian, Salt & Pepper, Speckle, and Poisson into all 60,000 images from the CIFAR-100 dataset. This process yielded a total dataset of 240,000 images for the generalized DAE, with

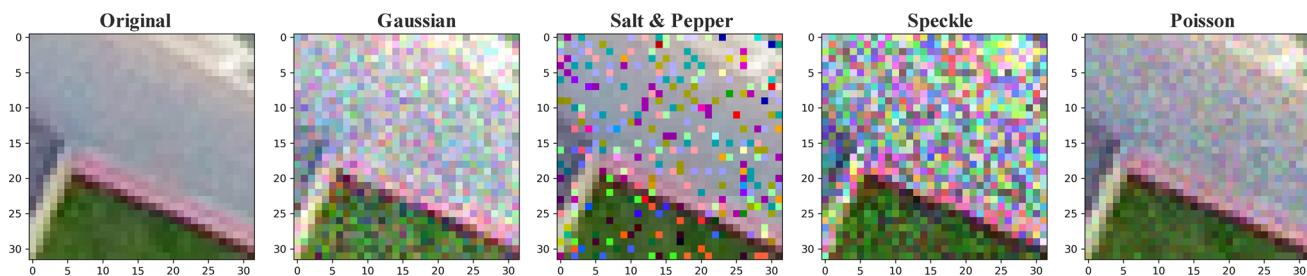


Fig. 3 Image noise types demonstration on 32×32 pixels cropped image sample of a drone-captured image



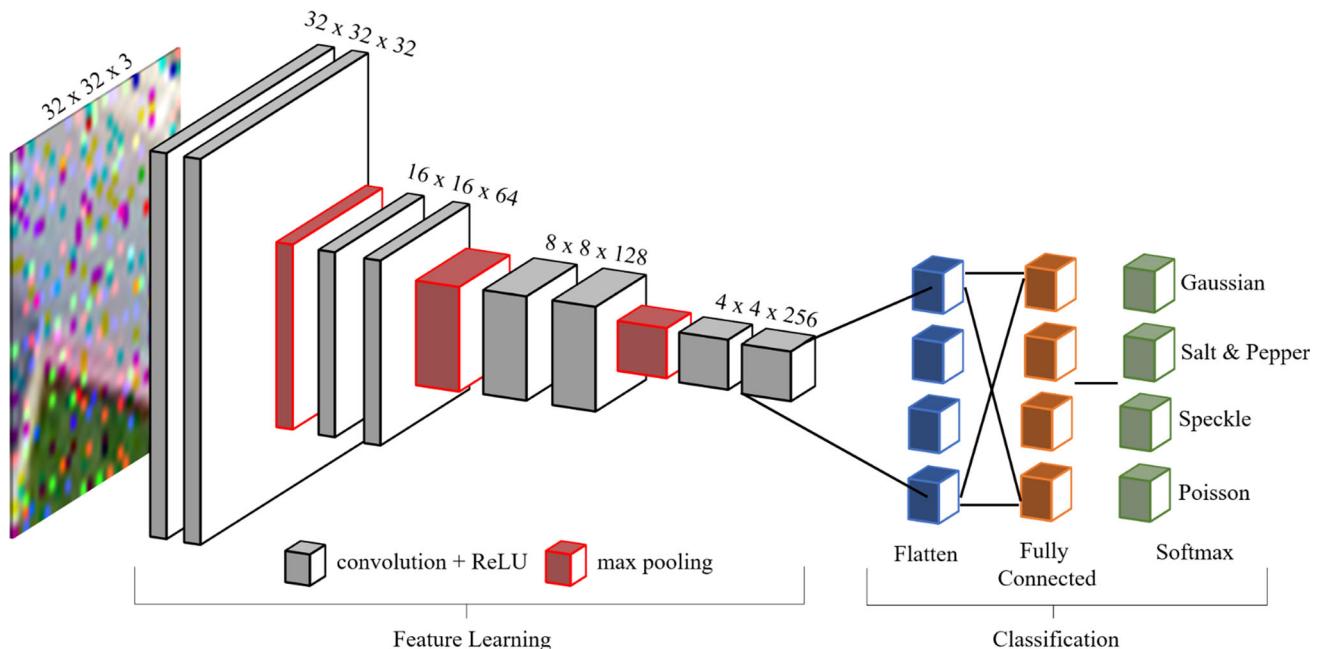
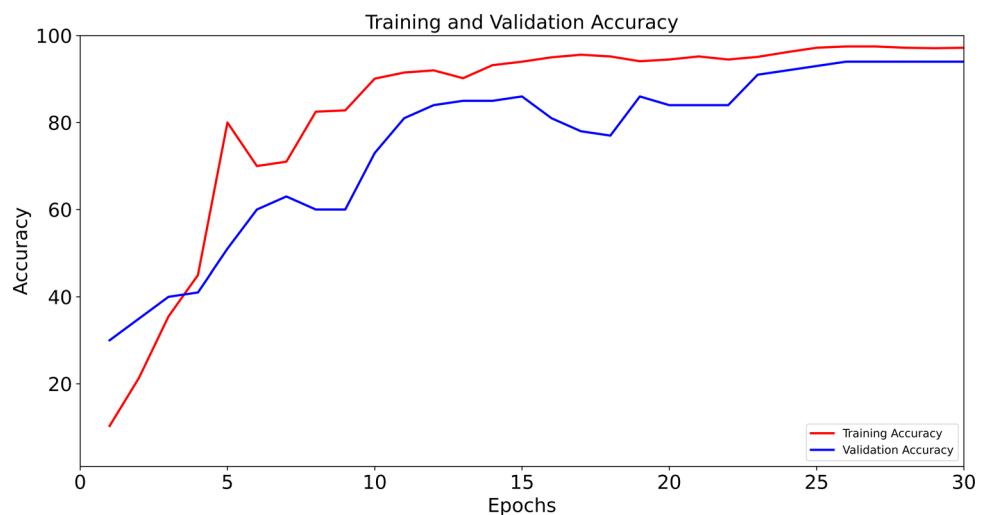


Fig. 4 Proposed CNN network for image noise type classification

Fig. 5 Network 1 training and validation accuracy



60,000 images dedicated to each noise type for specialized DAEs.

For the specialized DAEs, the image datasets were split into 50,000 images for training and 10,000 for validation for each noise type. The dataset for the generalized DAE was divided into 200,000 training images and 40,000 validation images. Similar to the classification network, the image datasets were transformed into Python arrays, shuffled, and stored as NumPy arrays to facilitate the training of both specialized and generalized DAEs.

4.5 Implementing Specialized DAEs and Generalized DAE

An autoencoder is an unsupervised artificial neural network designed to replicate its input in the output. When working with image datasets, the autoencoder transforms an image into a lower-dimensional representation, which can then be reconstructed to reveal the original image. A specific type of autoencoder is the Denoising Autoencoder.

Denoising autoencoders are trained to utilize a hidden layer for reconstructing a model from its inputs. To implement Denoising Autoencoders for denoising images based on the results of the first network for image noise type classification, we trained a total of five models. Each Denois-



ing Autoencoder was individually trained with 40 epochs, provided with both noisy and clean data. Four of these autoencoders were specialized DAEs, namely Gaussian DAE, Salt & Pepper DAE, Poisson DAE, and Speckle DAE. Additionally, one model was a Generalized DAE, included for performance comparison between specialized and generalized DAEs.

4.6 Additional Methods for High-Resolution Images

Initially, we tested our denoising autoencoders (DAEs) on 32×32 pixel images. To accommodate high-resolution images, we introduced several additional methods developed in the Python programming language. These methods were strategically integrated at various stages within the

framework, as depicted in Fig. 2. They are designed to be activated exclusively when the framework encounters a high-resolution image.

4.6.1 Image Sampling Method

To predict noise from a high-resolution image, we found it necessary to take more than just one sample from the image. During our experiments, we observed that the first network could not consistently predict the correct noise type for certain drone-captured high-resolution images, particularly when the image contained dark areas. Such occurrences are common in drone-captured images taken at night or in low luminance conditions. As an example, Fig. 7 displays an image sample with regions of darkness.

Fig. 6 Network 1 training and validation loss

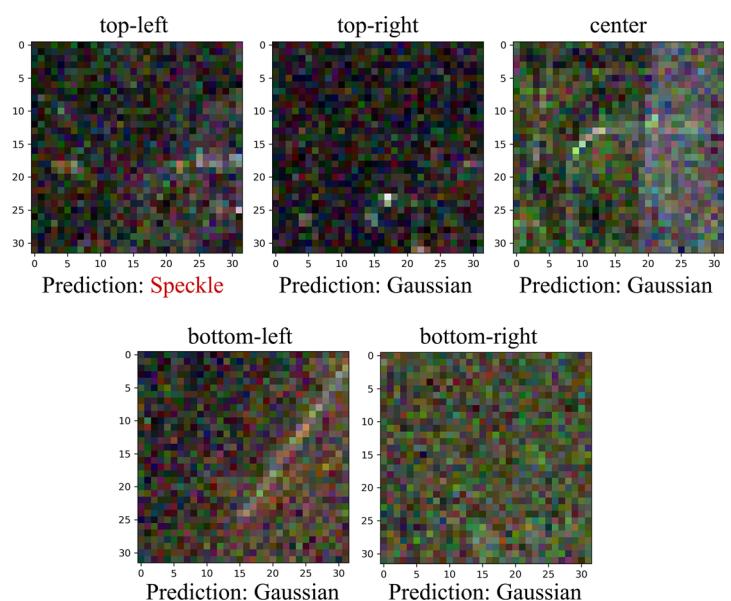
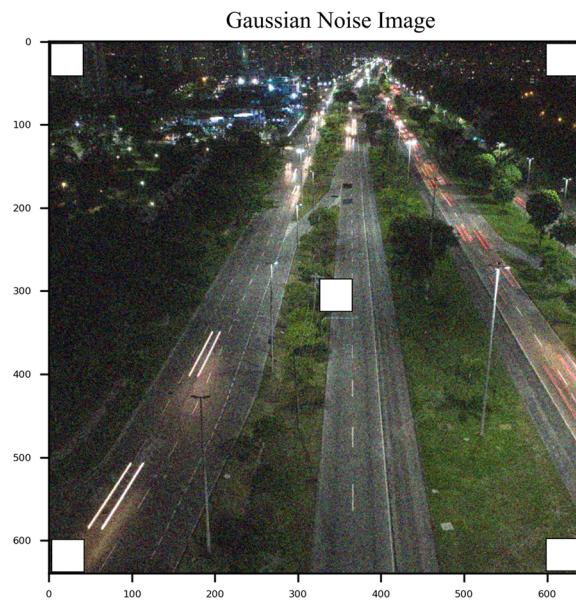
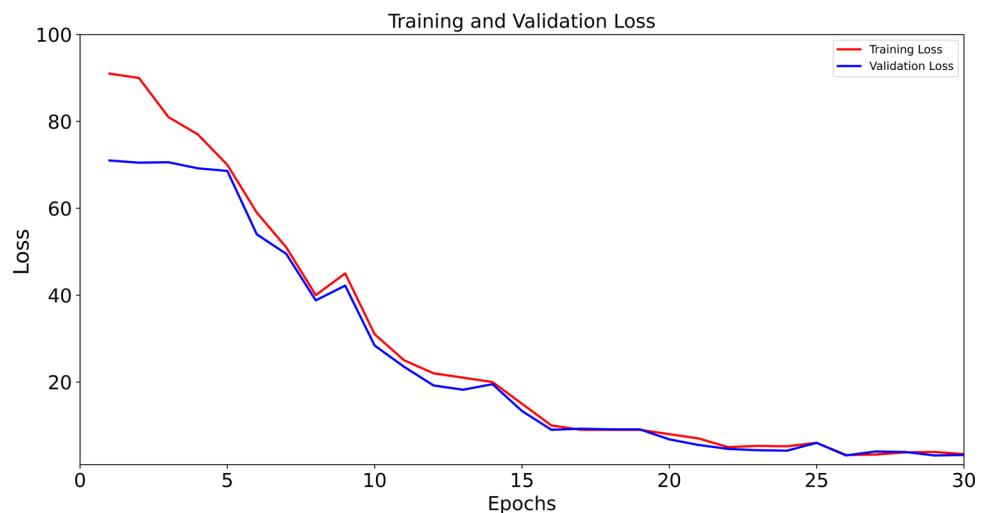


Fig. 7 Drone-captured image at night and has low luminance and prediction results



When the network attempts to predict the noise type based on a single 32×32 pixel sample from an image, it may produce incorrect predictions because the sample could be entirely black or lack precise noise detail. To address this, we adopted a method of cropping five samples of 32×32 pixels from a high-resolution image. These samples include four from each corner and one from the center of the image. Subsequently, these cropped samples are individually passed to the first network for image noise type prediction.

4.6.2 Prediction Noise Result Method

This method collects results for each sample processed by the “image sampling method” and stores them in an array. Based on all sample results, the final result is calculated. The logic behind the final result is straightforward: the noise class with the highest number of predictions is considered the noise type of the input image.

For instance, if, out of 5 samples, 4 or 3 samples are correctly predicted as Salt & Pepper, then the image is determined to have Salt & Pepper noise, and the corresponding Salt & Pepper Denoising Autoencoder (DAE) is activated to remove the noise. In the case of the image sample shown in Fig. 7, where four out of five samples are correctly predicted as Gaussian, and one is erroneously predicted as Speckle, the specific Gaussian DAE is activated to eliminate Gaussian noise from the image.

It’s important to note that this research exclusively addresses images with a single type of noise; it does not handle mixtures of different noises within a single image. This technique effectively predicts noise types for images with a singular noise type.

4.6.3 Noisy Image Cropping Method

Once the noise type is predicted by the “prediction noise result method,” the high-resolution image is divided into equal 32×32 pixel samples. These cropped image samples are then organized in a queue with the appropriate order. Depending on the final result obtained from the “prediction noise result method,” a specific Denoising Autoencoder (DAE) is activated. Subsequently, all image samples from the queue are individually processed by the specific DAE, which denoises each image sample.

4.6.4 Denoised Image Joiner Method

After each image sample is denoised, this method inserts the image samples into the denoised sample queue in proper order. Then it puts each sample in the same position and joins all the image samples to reconstruct the drone-captured high-resolution denoised image. This way, all these methods work

together to handle the drone-captured high-resolution image using the same DAEs trained for 32×32 pixel images.

Our proposed approach for image denoising and classification offers several key advantages over existing methods. Firstly, we provide comprehensive integration by seamlessly combining image noise type classification and denoising tasks within a single framework. This integration streamlines the workflow, eliminating the need for separate systems. Secondly, we treat image classification and denoising as distinct tasks, allowing us to focus on their unique challenges and achieve more accurate results. Thirdly, we selected a small-resolution CIFAR-100 image dataset, ensuring computational efficiency and accessibility for a wider range of users. Despite the small-resolution dataset, our approach effectively handles high-resolution images, showcasing its flexibility for real-world applications. Lastly, our framework leverages specialized Denoising Autoencoders (DAEs) instead of Generalized Autoencoders, resulting in improved denoising performance tailored to specific types of noise. By capitalizing on these advantages, our proposed framework offers a robust and efficient solution for image noise type classification and denoising, making it a promising choice for future research.

5 Results and Discussion

This section provides a summary and discussion of all the experiments we conducted to evaluate our proposed framework. We initially performed two tasks separately and then integrated them into a single system. After training and validating “Network 1 - Image Noise Type Classification” using the CIFAR-100 dataset, we tested this network with images from the CIFAR-10 dataset. Subsequently, we introduced additional methods into the framework and tested our first network on high-resolution images.

For high-resolution image noise prediction, we used a combination of images from the BSD500 dataset and a selection of randomly captured drone images from the internet. The results of noise prediction by the first network are presented in Fig. 8 for 32×32 pixel images and Fig. 9 for high-resolution images.

The root mean square error (RMSE) can be computed using the formula:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

where y_i represents the actual label and \hat{y}_i denotes the predicted label.

Table 2 presents the RMSE values for different types of noise that have corrupted low-resolution images (32×32)



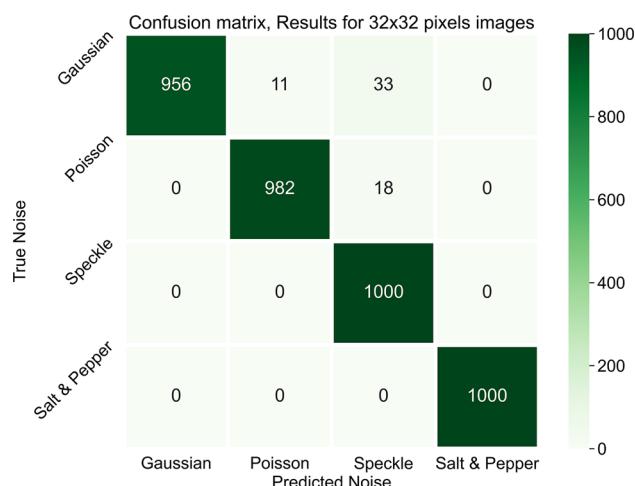


Fig. 8 Noise type prediction results for 32 × 32 images

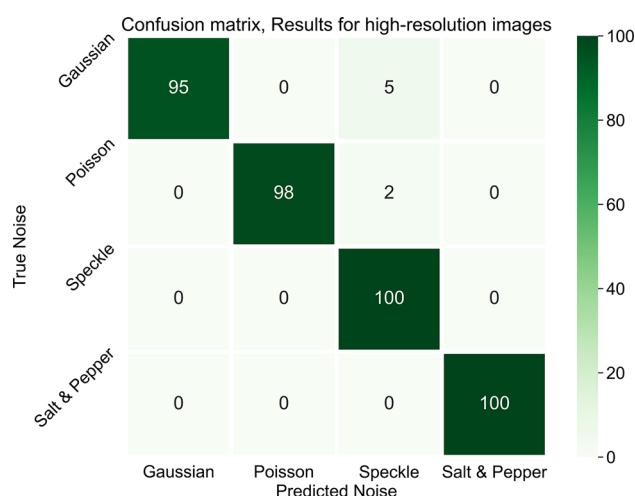


Fig. 9 Noise type prediction results for high-resolution images

and high-resolution images of various sizes. The noise types include Gaussian, Poisson, Speckle, and Salt & Pepper. For example, the RMSE for 32 × 32 resolution images was 0.2098 for Gaussian noise, 0.1342 for Poisson noise, and 0 for Speckle and Salt & Pepper noise. Similarly, for high-resolution images, the RMSE values were 0.2236 for Gaussian noise, 0.1414 for Poisson noise, and 0 for Speckle and Salt & Pepper noise.

To assess the performance of RMSE, the squared difference $(y_i - \hat{y}_i)^2$ was treated as 1 for false negatives and 0 for true positives. Notably, the RMSE for Speckle noise and Salt & Pepper noise turned out to be 0, indicating the absence of false negatives for these two noise types. However, the RMSE was highest for Gaussian noise, ranging from 4.44 to 5%, which indicates the presence of false negatives in this particular case.

Figures 8 and 9 display the performance of the first network. For both CIFAR-10 32 × 32 pixel images and high-resolution images (a combination of BSD500 and self-collected drone-captured images with dimensions greater than or equal to 1024 × 1024 pixels), the Speckle and Salt & Pepper noise types were predicted perfectly. There was a minor misclassification for the Poisson and Gaussian noise types. This misclassification may be attributed to the similarity in the nature of these two noises. Sometimes, even to the human eye, it is challenging to distinguish between Poisson and Gaussian noise types in colorful images. Network 1 achieved an impressive overall accuracy of 98%, correctly classifying all four noise types without any issues.

To evaluate the performance of our proposed method, we conducted a meticulous comparison of image noise type classification with the previous studies [28, 29]. Notably, among the reviewed literature, these studies closely aligned with our specific topic, making them valuable references for our research.

Table 2 RMSE calculation for all four noise types for different resolutions of images

Resolution/noise type	Gaussian	Poisson	Speckle	Salt & Pepper
32 × 32	0.2098	0.1342	0	0
High-resolution	0.2236	0.1414	0	0

Table 3 Accuracy comparison between the proposed method and [28, 29]

Test dataset	Noise types	[28] (%)	[29] (%)	Proposed method (%)
CIFAR-10	Gaussian, Salt & Pepper, Speckle, Poisson	99.8	NA	99.6
MNIST	Gaussian, Salt & Pepper, Speckle, Poisson	88.5	NA	91.2
BSD500	Gaussian, Salt & Pepper, Speckle, Poisson	93.7	NA	98.3
Facial dataset (Grayscale)	Gaussian, Salt & Pepper, Poisson	NA	94.7	98.6



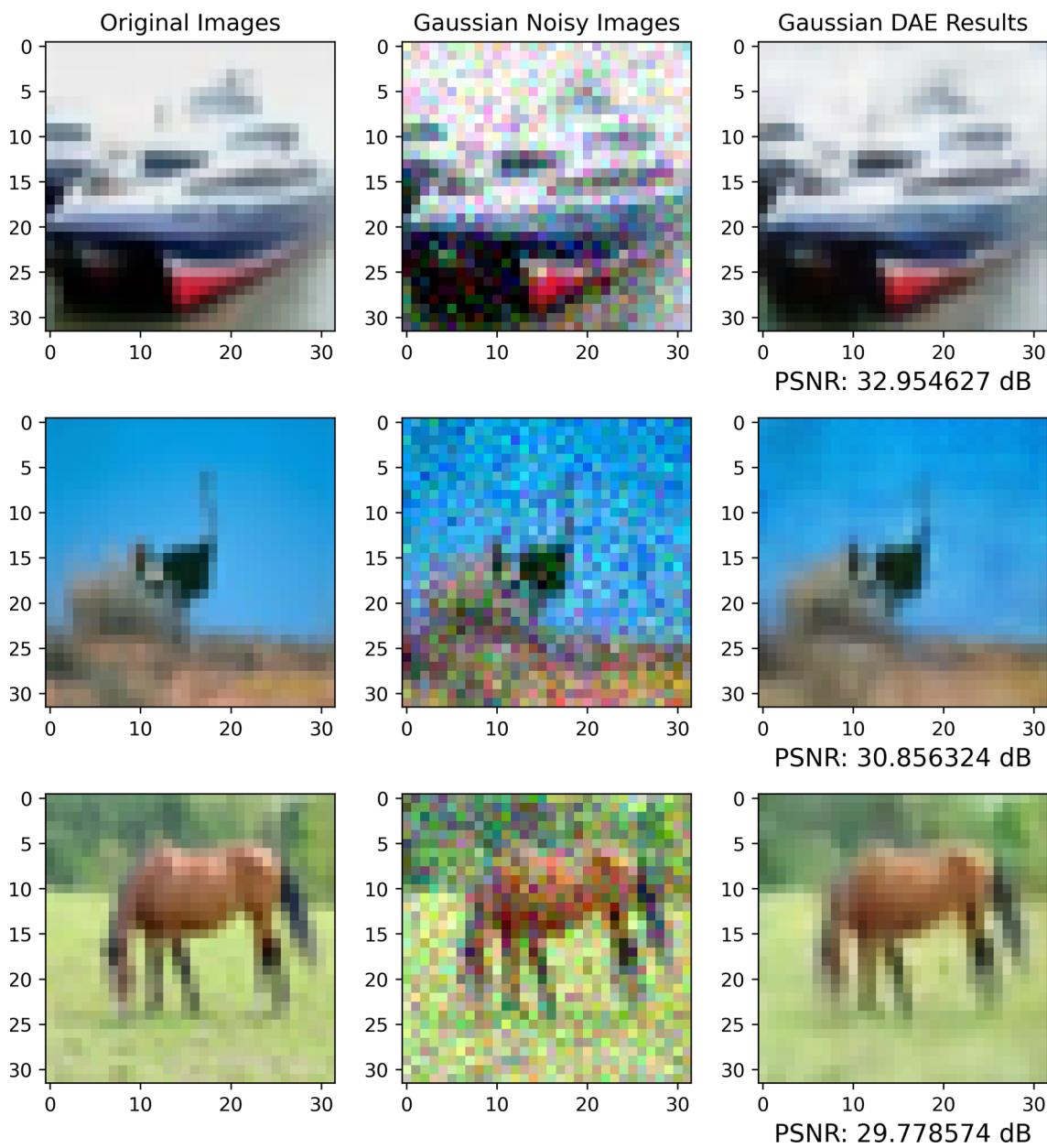


Fig. 10 Specialized Gaussian DAE results

We trained our image noise type classification model with the same datasets and conditions used in [28, 29] to test the performance of our method against the previously proposed state-of-the-art methods for a fair and unbiased comparison, and the comparison results are given in Table 3.

Table 3 discusses the comparison of our proposed method with state-of-the-art methods in [28, 29]. The classifier in [28] was trained using CIFAR-10, MNIST, and BSD500 datasets by the authors. Therefore, for a fair comparison, we trained and tested our network on CIFAR-10, MNIST, and BSD500 datasets. It can be seen from the table that the proposed classifier achieved accuracy that is closer to the claimed accuracy

in [28] for the testing dataset of CIFAR-10, whereas it clearly outperformed [28] for the testing datasets of MNIST, and BSD500.

To compare with [29], we chose the facial dataset used in [29]. However, since our classifier requires a 3D matrix for training and testing purposes, we stacked the noisy grayscale image three times to fulfill the requirement of our classifier for a three-dimensional input image while maintaining the grayscale property for fair comparison. Additionally, as our architecture is a four-class classifier, to train it on three classes, we excluded speckle images from the training process to deactivate all the weights and neurons corresponding



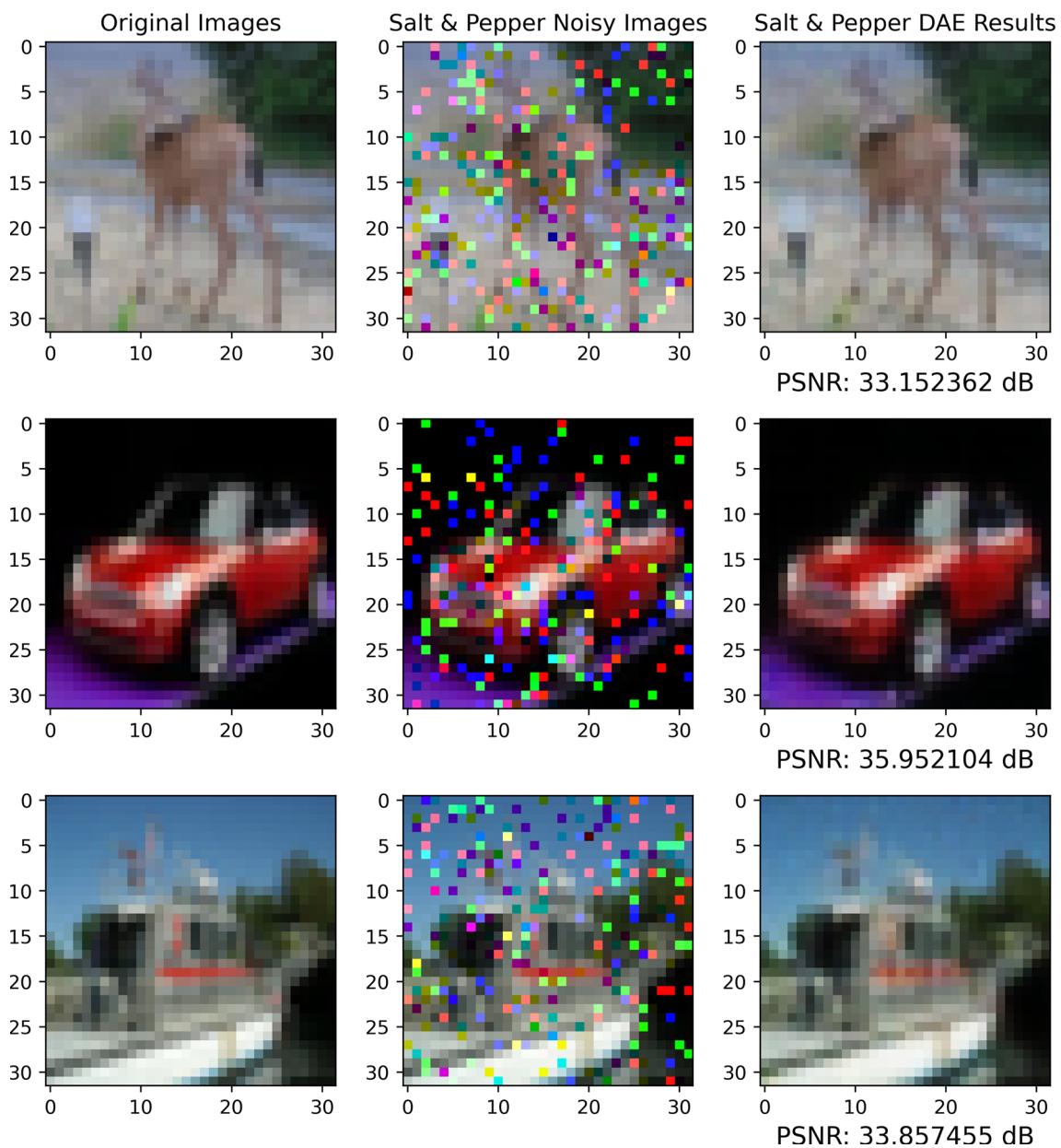


Fig. 11 Specialized Salt & Pepper DAE results

to the absent class. For the facial dataset, the proposed method clearly outperformed [29] by achieving 3.9% higher accuracy, as can be seen in Table 3.

Comparatively, our proposed method outperformed the state-of-the-art in terms of noise-type classification for low-resolution images. For high-resolution images, it was not possible to provide a fair comparison, as [28, 29] did not mention criteria for dealing with such images. The high accuracy achieved by the proposed method for high-resolution images demonstrates its superiority and novelty, positioning it as an effective architecture with the potential to become a real-world application in the mentioned domain.

After training and validating “Network 2—Denoising Autoencoders,” all Denoising Autoencoders (DAEs) were initially tested on the CIFAR-10 noisy image dataset. The CIFAR-10 noisy dataset, which contains all four types of noisy images, was created using MATLAB to test the denoising network’s performance. We added 10% of the noise of each noise type to all images of the CIFAR-10 dataset. The results for each specialized DAE are shown in Figs. 10, 11, 12, and 13.





Fig. 12 Specialized Speckle DAE results

For the evaluation of all four specialized denoising autoencoders (DAEs), we calculated the Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) for the denoised images shown in Figs. 10, 11, 12 and 13. The results are presented in Table 4.

It can be observed from Table 4 that the specialized DAEs exhibit varying denoising performance when applied to different images. The Gaussian DAE achieved PSNR values ranging from 29.7786 to 32.9546 and SSIM values between 0.8047 and 0.8764. The S&P DAE demonstrated excellent denoising capabilities with PSNR values ranging from

33.1524 to 35.9521 and SSIM values between 0.8808 and 0.9441.

The Speckle DAE showed competitive results with PSNR values ranging from 32.1199 to 35.4452 and SSIM values between 0.8576 and 0.933. However, the Poisson DAE exhibited the lowest denoising performance among the specialized DAEs, achieving PSNR values between 28.6745 and 31.2358, and SSIM values ranging from 0.7798 to 0.8377.



Fig. 13 Specialized Poisson DAE results

On average, the specialized DAEs provided a reasonable PSNR of 32.313 and a mean SSIM of 0.863. These results suggest that the specialized DAEs are effective in denoising various types of images, with performance variations depending on the noise type and image content. The comparison between specialized DAEs and generalized DAE results are shown in (Fig. 14).

We also trained a generalized DAE to compare its performance against the specialized DAEs. We evaluated the PSNR values of 1000 randomly selected noisy CIFAR-10 images, with 250 images from each noise type.

Our results showed that specialized DAEs outperformed the generalized DAE for 782 out of the 1000 images. Among these, 663 images had significantly higher PSNR values, with differences ranging from 5 to 7 dB, while the remaining images exhibited minor differences of 1–2 dB.

For the remaining 218 images, the generalized DAE performed better, producing higher PSNR values. Among these, 104 images showed differences in PSNR values between 1 and 3 dB, while the rest had differences below 1 dB.



Table 4 PSNR and SSIM comparison of specialized DAEs on various images

DAE type/image	PSNR	SSIM
Gaussian DAE (Boat Image, Fig. 10)	32.9546	0.8764
Gaussian DAE (York Image, Fig. 10)	30.8563	0.8291
Gaussian DAE (Horse Image, Fig. 10)	29.7786	0.8047
S&P DAE (Deer Image, Fig. 11)	33.1524	0.8808
S&P DAE (Car Image, Fig. 11)	35.9521	0.9441
S&P DAE (Boat Image, Fig. 11)	33.8575	0.8968
Speckle DAE (Bulldozer Image, Fig. 12)	34.1006	0.9023
Speckle DAE (Bird Image, Fig. 12)	32.1199	0.8576
Speckle DAE (Horse Image, Fig. 12)	35.4452	0.933
Poison DAE (Deer Image, Fig. 13)	29.6279	0.8014
Poison DAE (Car Image, Fig. 13)	31.2358	0.8377
Poison DAE (Truck Image, Fig. 13)	28.6745	0.7798
Average	32.313	0.863

The superiority of specialized DAEs over the generalized DAE is evident in denoising images with specific noise types. After thoroughly testing specialized DAEs on CIFAR-10 images, we applied them to denoise higher-resolution images.

Figures 15, 16, 17, and 18 showcase drone-captured 640×640 pixel images randomly obtained from the internet to illustrate noise removal. This choice enables us to present clear and easily interpretable examples on paper.

In Fig. 15, Gaussian noise has introduced a minor brightness effect in the image. After denoising, the Gaussian noise and brightness are effectively removed, resulting in a PSNR value of 31.269 dB.

Figure 16 displays Salt & Pepper noise, which is successfully removed by the specialized DAE, resulting in a PSNR value of 28.701 dB.

Similarly, Fig. 17 demonstrates Speckle noise removal, with the denoised image achieving a PSNR value of 32.072 dB.

In Fig. 18, the denoising process effectively removes Poisson noise, resulting in a PSNR value of 30.077 dB. It's worth noting that image denoising also enhances color saturation in some noisy images.

In all these figures, the specialized DAEs demonstrate their effectiveness in noise elimination and image reconstruction, consistently yielding good PSNR values.

6 Conclusion and Future Work

The proposed framework serves the purpose of classifying noise types for image denoising. It has been applied in various practical applications, such as reconstructing and denoising drone-captured images. Using specialized DAEs instead of a generalized approach has led to better PSNR values for the reconstructed images.

Nevertheless, the current framework does exhibit certain limitations that require attention to enhance its overall effectiveness. Firstly, the first network designed for noise type identification is optimized for images with a single type of noise, and its accuracy diminishes when confronted with images containing a mixture of noises. Secondly, the denoising capability of the second network is limited to images with noise levels up to 30%, rendering it less effective for denoising images with higher noise densities. Consequently, the framework may not be suitable for applications where images exhibit elevated noise levels. These limitations underscore the need for further refinement and development to enhance the framework's versatility and performance across a broader spectrum of noise scenarios.

In future work, the framework could be enhanced to achieve real-time performance in practical applications. One potential approach involves expanding the range of supported noise types and training additional specialized denoising autoencoders (DAEs) within the framework to address various noise types in images. Additionally, retraining the framework with a higher-resolution image dataset could significantly reduce the reconstruction time for high-resolution images. Another avenue for improvement is the implementation of a noise type detection strategy, where the framework automatically identifies the noise type in the source and applies the corresponding specialized DAE tailored for that specific noise type over multiple consecutive frames, thereby enhancing denoising results. Furthermore, the incorporation of voting-based or ensemble classification mechanisms using consecutive frames could further enhance the accuracy of the noise-type classifier. These proposed enhancements aim to optimize the framework's efficiency and robustness, making it more suitable for real-world denoising tasks.



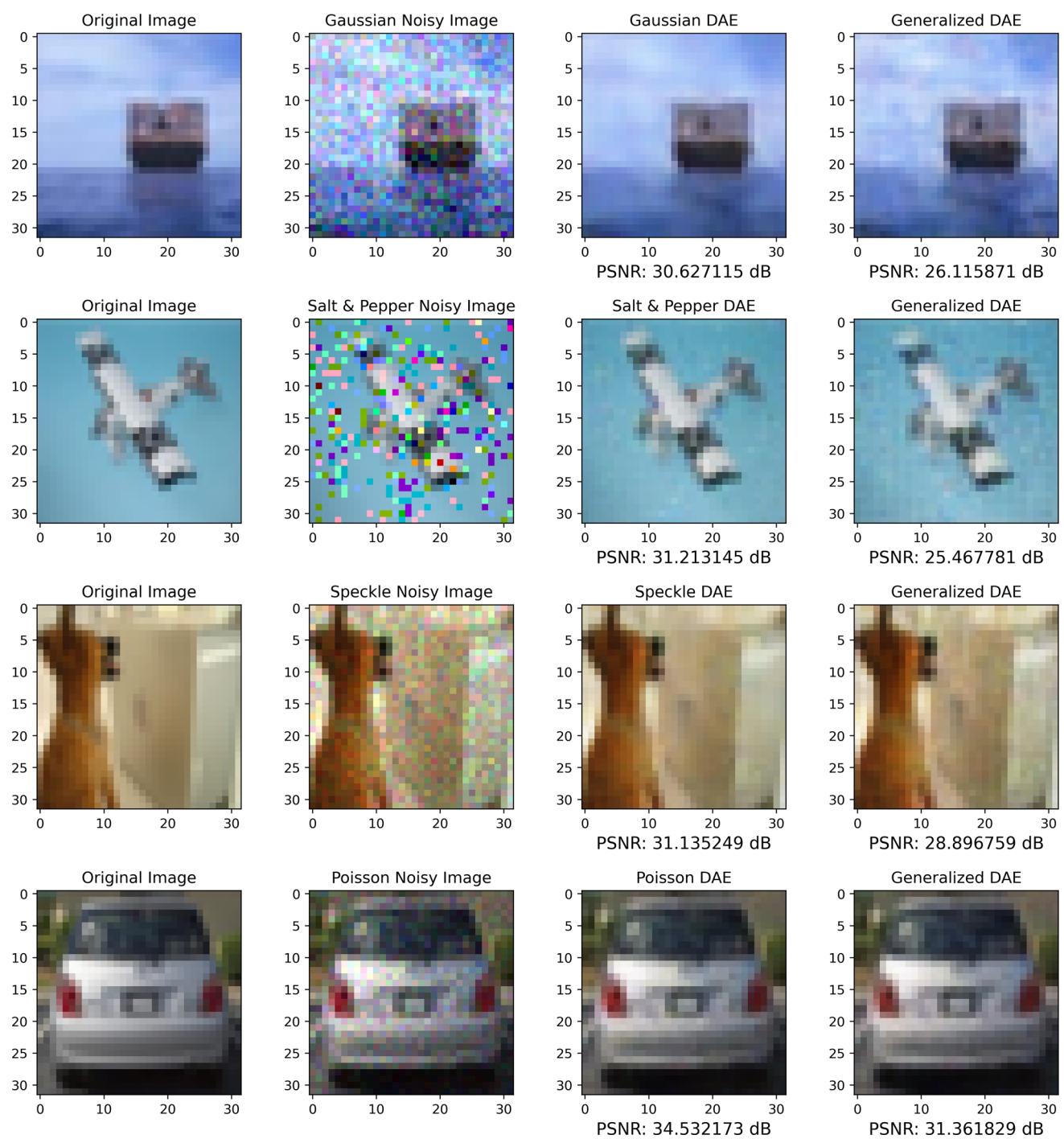


Fig. 14 Comparison between specialized DAEs and generalized DAE results



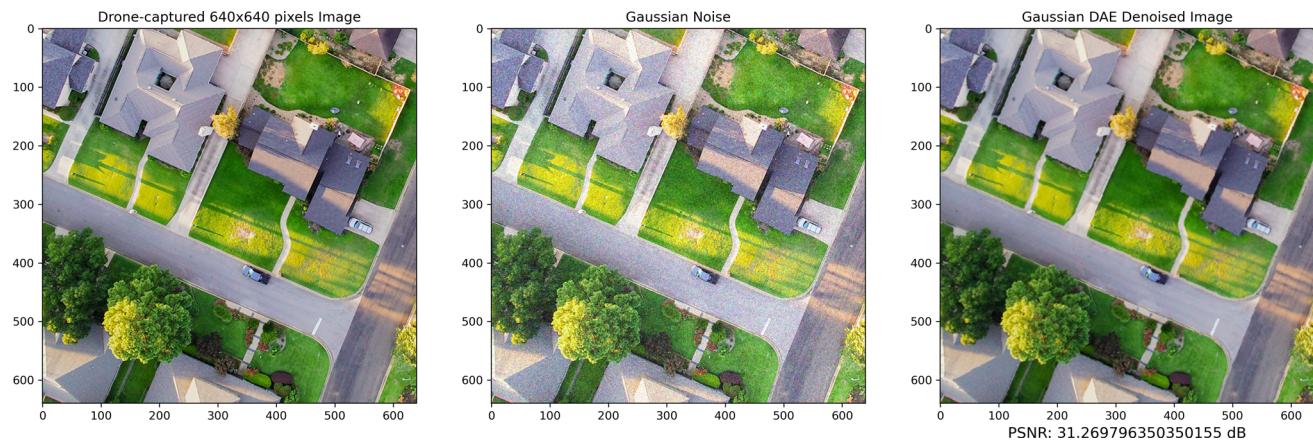


Fig. 15 Specialized Gaussian DAE result for a 640×640 pixels drone-captured image



Fig. 16 Specialized Salt & Pepper DAE result for a 640×640 pixels drone-captured image

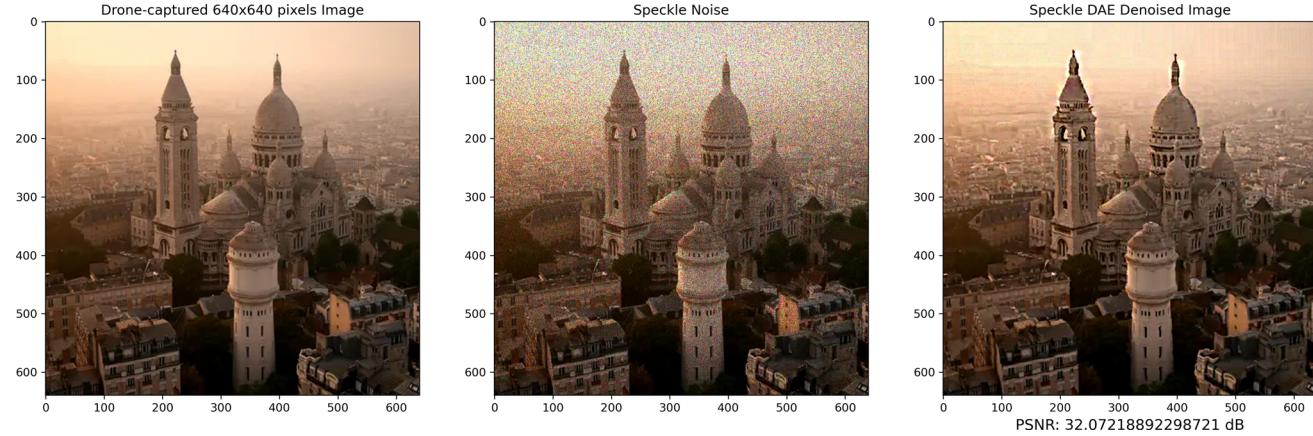


Fig. 17 Specialized Speckle DAE result for a 640×640 pixels drone-captured image



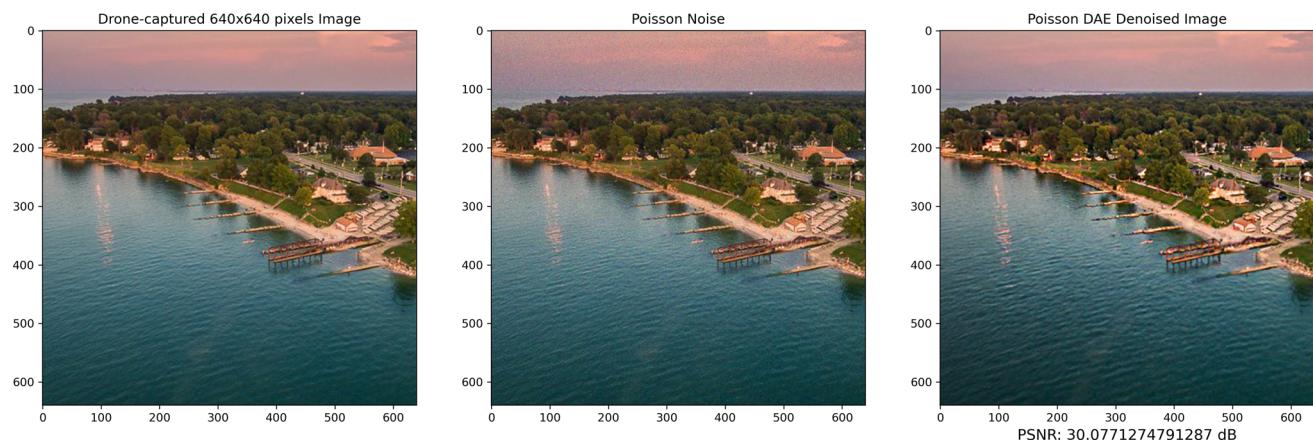


Fig. 18 Specialized Poisson DAE result for a 640×640 pixels drone-captured image

Author Contributions Conceptualization and methodology, SK and WA; experiments, WA and SK; validation, WA and AN; writing-review and editing, WA, SK, and GM; project administration and funding acquisition, AN. All authors have read and agreed to the published version of the manuscript.

Funding This research work was funded by Institutional Fund Projects under Grant No. (IFPIP:169-611-1443). The authors gratefully acknowledge the technical and financial support by the Ministry of Education and King Abdulaziz University, DSR, Jeddah, Saudi Arabia.

Data Availability Data available in Github repository.

Declarations

Conflict of interest The authors declare no conflict of interest.

References

- Yuan, Y.; Shen, Q.; Wang, S.; Ren, J.; Yang, D.; Yang, Q.; Fan, J.; Mu, X.: Coronavirus mask protection algorithm: a new bio-inspired optimization algorithm and its applications. *J. Bionic Eng.* **20**(4), 1747–1765 (2023). <https://doi.org/10.1007/s42235-023-00359-5>
- Yuan, Y.; Yang, Q.; Ren, J.; Fan, J.; Shen, Q.; Wang, X.; Zhao, Y.: Learning-imitation strategy-assisted alpine skiing optimization for the boom of offshore drilling platform. *Ocean Eng.* **278**, 114317 (2023). <https://doi.org/10.1016/j.oceaneng.2023.114317>
- Yuan, Y.; Ren, J.; Wang, S.; Wang, Z.; Mu, X.; Zhao, W.: Alpine skiing optimization: a new bio-inspired optimization algorithm. *Adv. Eng. Softw.* **170**, 103158 (2022). <https://doi.org/10.1016/j.advengsoft.2022.103158>
- Yuan, Y.; Mu, X.; Shao, X.; Ren, J.; Zhao, Y.; Wang, Z.: Optimization of an auto drum fashioned brake using the elite opposition-based learning and chaotic k-best gravitational search strategy based grey wolf optimizer algorithm. *Appl. Soft Comput.* **123**, 108947 (2022). <https://doi.org/10.1016/j.asoc.2022.108947>
- Russo, F.: Edge detection in noisy images using fuzzy reasoning. *IEEE Trans. Instrum. Meas.* **47**(5), 1102–1105 (1998)
- Khan, S.; Lee, D.H.; Khan, M.A.; Gilal, A.R.; Iqbal, J.; Waqas, A.: Efficient and improved edge detection via a hysteresis thresholding method. *Curr. Sci.* (2020)
- Bustince, H.; Barrenechea, E.; Pagola, M.; Fernandez, J.: Interval-valued fuzzy sets constructed from matrices: application to edge detection. *Fuzzy Sets Syst.* **160**(13), 1819–1840 (2009)
- Thaha, M.M.; Kumar, K.P.M.; Murugan, B.S.; Dhanasekeran, S.; Vijayakarthick, P.; Selvi, A.S.: Brain tumor segmentation using convolutional neural networks in MRI images. *J. Med. Syst.* **43**, 1–10 (2019)
- Qiao, Y.; Truman, M.; Sukkarieh, S.: Cattle segmentation and contour extraction based on mask R-CNN for precision livestock farming. *Comput. Electron. Agric.* **165**, 104958 (2019)
- Sobehi, R.A.; Tekli, J.: Comparing deep learning models for low-light natural scene image enhancement and their impact on object detection and classification: overview, empirical evaluation, and challenges. *Signal Process. Image Commun.* **109** (2022)
- Jasti, V.D.P.; Zamani, A.S.; Arumugam, K.; Naved, M.; Pal-lathadka, H.; Raghuvanshi, A.; Kaliyapremal, K.: Computational technique based on machine learning and image processing for medical image analysis of breast cancer diagnosis. *Secur. Commun. Netw.* **2022** (2022)
- Abdou, M.A.: Literature review: efficient deep neural networks techniques for medical image analysis. *Neural Comput. Appl.* **34**, 5791–5812 (2022)
- Xie, J.; Xu, L.; Chen, E.: Image denoising and inpainting with deep neural networks (2012)
- Eng, H.-L.; Ma, K.-Y.K.: Noise adaptive soft-switching median filter. *IEEE Trans. Image Process.* **10**, 242–251 (2001)
- Wang, T.; Qiu, J.; Fu, S.; Ji, W.: Distributed fuzzy h filtering for nonlinear multirate networked double-layer industrial processes. *IEEE Trans. Ind. Electron.* **64**, 5203–5211 (2017)
- Civicioglu, P.: Using uncorrupted neighborhoods of the pixels for impulsive noise suppression with ANFIS. *IEEE Trans. Image Process.* **16**, 759–773 (2007)
- Yin, H.; Gong, Y.; Qiu, G.: Side window filtering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8758–8766 (2019)
- El Helou, M.; Süsstrunk, S.: Blind universal Bayesian image denoising with gaussian noise level learning. *IEEE Trans. Image Process.* **29**, 4885–4897 (2020)
- Mahdaoui, A.E.; Ouahabi, A.; Moulay, M.-S.: Image denoising using a compressive sensing approach based on regularization constraints. *Sensors* **22**(6), 2199 (2022)
- Thanh, D.N.H.; Enginoglu, S.: An iterative mean filter for image denoising. *IEEE Access* **7**, 167847–167859 (2019)



21. Yamashita, R.; Nishio, M.; Do, R.K.; Togashi, K.: Convolutional neural networks: an overview and application in radiology. *Insights Imaging* **9**(4), 611–629 (2018)
22. Perrotta, C.; Selwyn, N.: Deep learning goes to school: toward a relational understanding of AI in education. *Learn. Media Technol.* **45**(3), 251–269 (2020)
23. Anand, R.; Shanthi, T.; Sabeenian, R.S.; Veni, S.: Real time noisy dataset implementation of optical character identification using CNN. *Int. J. Intell. Enterp.* **7**(1–3), 67–80 (2020)
24. Momeny, M.; Latif, A.; Agha Sarram, M.; Sheikhpour, R.; Zhang, Y.: A noise robust convolutional neural network for image classification. *Results Eng.* **5**, 100072 (2019)
25. Mansour, R.F.; Alfar, N.M.; Abdel-Khalek, S.; Abdelhaq, M.; Saeed, R.A.; Alsaqour, R.: Optimal deep learning based fusion model for biomedical image classification. *Expert Syst.* **39**(3), 12764 (2021)
26. Ajay, P.; Goyal, L.M.; Vashistha, S.; Kumar, V.; Kumar, A.: Unsupervised hyperspectral microscopic image segmentation using deep embedded clustering algorithm. *Scanning* **2022** (2022)
27. Kozierski, M.; Cyganek, B.: Image recognition with deep neural networks in presence of noise—dealing with and taking advantage of distortions. *Integr. VLSI J.* **24**(4), 337–349 (2017)
28. Liu, F.; Song, Q.; Jin, G.: The classification and denoising of image noise based on deep neural networks. *Appl. Intell.* **50**(7), 2194–2207 (2020)
29. Tripathi, M.: Facial image noise classification and denoising using neural network. *Sustain. Eng. Innov. Eng.* **3**(2), 25 (2021). <https://doi.org/10.37868/sei.v3i2.id142>
30. Chuah, J.H.; Khaw, H.Y.; Soon, F.C.; Chow, C.O.: Detection of gaussian noise and its level using deep convolutional neural network. In: IEEE Region 10 Annual International Conference Proceedings/TENCON, vol. 2017-Decem, pp. 2447–2450. IEEE (2017)
31. Roy, S.; Ahmed, M.; Akhand, M.: Noisy image classification using hybrid deep learning methods. *J. Inf. Commun. Technol.* (2018)
32. Ahmed, W.: Deep Learning-based Noise Type Classification and Removal for Drone Image Restoration—GitHub Repository. <https://github.com/waqr-ahmed51/Deep-Learning-based-Noise-Type-Classification-and-Removal-for-Drone-Image-Restoration> (2023)

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

