Al-Salam Journal for Engineering and Technology
Journal Homepage: http://journal.alsalam.edu.iq/index.php/ajest
e-ISSN: 2790-4822 p-ISSN: 2958-0862

Image Noise Detection and Classification Based on Combination of Deep Wavelet and Machine Learning

Rusul A. Al Mudhafar [1]*, Nidhal K. El Abbadi [2]

[1]Computer Science Department, Faculty of Computer Science and Mathematics, University of Kufa, Najaf, Iraq.
[2]Computer Techniques Engineering Department, Al-Mustaqbal University College, Babylon, Iraq.

*Corresponding Author: Rusul A. Al Mudhafar

DOI: https://doi.org/10.55145/ajest.2024.03.01.003
Received May 2023; Accepted August 2023; Available online August 2023

ABSTRACT: In the last decade, the number of digital images has increased dramatically. Noise is unwanted particles or signals contaminating the image during the captured image and transmission. Image noise reduces the image quality and increases the processing failure ratio. It is highly recommended to remove the noise, and before removing the noise, we have to know the type of noise, which highly assists in suggesting the proper de-noise algorithm. This study introduces a method to effectively detect and recognize image noise of various types (Gaussian, lognormal, Rayleigh, Salt & Pepper, and Speckle). The proposed model consists of two stages: the first stage is detecting the noise in an image using Convolutional Neural Network. The second stage classifies the noisy images into one of five types of noise using a new method based on a combination of deep wavelet machine learning classifiers, we select five machine learning classifiers (support vector machine, decision tree, random forest, logistic regression, and K-nearest neighbor) to choose the more efficient classifier ultimately. The combination of wavelet with machine learning, specifically SVM, can highly enhance the results, where the accuracy was (91.30 %) through many experiments conducted to build a sturdy classification model.

Keywords: Deep Wavelet, Gaussian noise, Lognormal, Rayleigh, Salt and Pepper, Speckle

1. INTRODUCTION

Images are essential multimedia for humans to acquire outside information and play an important role in human activities. However, in the process of digital image acquisition, conversion, and transmission, some isolated pixel points with random positions and random values (image noise) appear in the image due to the factors of the imaging device itself or the influence of the external environment [1]. We study image noise because it is a common problem in image processing and computer vision that can significantly affect the quality and accuracy of image analysis results. The presence of image noise can make it difficult to accurately detect and analyze image features, such as edges or textures, and can also introduce errors in image classification and recognition tasks. Therefore, it is important to study image noise and develop techniques to remove or reduce it, so that we can improve the quality and reliability of image analysis results. Many fundamental noise types, including Salt-and-Pepper noise, Gaussian noise, Poisson noise, and Speckle noise, can degrade the image's quality. These noises are added due to faulty memory location, post-filtering, compression, weak focal length, and other adverse conditions that may be present in the atmosphere or from image-capturing gadgets [2]. Noise classification allows us to apply the best denoising method to an image after determining the most likely noise distribution in the image [3]. Noise detection and Classification can be challenging, as several potential issues can arise. Some common problems include:

• Lack of labeled data: One of the biggest challenges in noise detection and Classification is the availability of labeled data for training machine learning models. In many cases, sufficient data may be available to accurately train a model, leading to poor performance and accuracy [4].

• Variability in noise sources: Noise can come from a wide range of sources, which can vary in their characteristics. For example, traffic noise may sound very different from industrial noise, and this variability can make it difficult to classify and distinguish between different types of noise accurately [5].

**Corresponding author: Rusula.mudhafar@student.uokufa.edu.iq*

• The complexity of noise signals: Noise signals can be complex and challenging to analyze, particularly when they contain multiple frequency components or are affected by environmental factors such as reverberation or interference. This complexity can make it difficult to detect and classify noise accurately [6].

• Noise in real-world environments: Noise detection and classification systems may need to operate in real-world settings where the conditions may be less controlled and more challenging. For example, background noise levels may be high, or the system may need to distinguish between noise and other environmental sounds [7].

• Limited computational resources: Noise detection and classification systems may need to operate in real-time and be implemented on hardware with limited computational resources. This can limit the complexity of the algorithms that can be used and may impact the system's accuracy [8].

• Adaptability to new noise types: A noise detection and classification system trained on one set of noise types may not perform well on new or previously unseen types of noise. As a result, the system may need to be retrained or adapted to new kinds of noise over time [9].

The primary objective of this paper is to present a novel approach for image noise detection and classification, combining the power of deep wavelet techniques with machine learning algorithms. By addressing the challenges mentioned earlier, we aim to develop a robust and efficient system that can accurately identify and classify various noise types present in images.

Furthermore, this research has significant real-world applications. Noise-free images are critical for various fields, including medical imaging, autonomous vehicles, surveillance, and satellite imagery analysis. By effectively removing or reducing noise from images, our proposed method can enhance the performance and reliability of these applications.

2. RELATED WORK

Many researchers work on noise type classification. Research by Hiremath [10] classifies noise using a pre-trained neural network using transfer learning. They have considered two types of noise; Electronic noise and Impulse noise. They thought of 400 different images, among which 200 images are for each noise present in an image. The result shows the Average accuracy obtained at 94.37. The benefit of this method was high accuracy rate (AR) of Salt & pepper noise than Gaussian noise. Based on our review of the paper, it appears that the author did not specifically mention how many types of noise were classified. The article refers to "noise" in general. Tripathi [11] conducted a study to classify three types of noises (Gaussian, Poisson, and Salt & Pepper). A CNN and UNET-based model architecture is designed, implemented, and evaluated. The facial image dataset is processed and then used to train, validate and test the models. The training and validation accuracy for the CNN model is 99.87% and 99.92%, respectively. The advantage of the UNET model is that it can get optimal PSNR and SSIM values for different noises. The paper does not explicitly mention how many types of noise are classified in the proposed method. Huo and XiaoXuan [12] a low-accuracy recognition algorithm was solved using a Back Propagation) BP (neural network. First, image noise and two recognition methods are evaluated. Second, the algorithm's input value and network structure are designed. The BP neural network recognition algorithm's training technique and decision criteria are given. Matlab software is used to simulate and identify salt and pepper noise spots. This algorithm has a deficient noise leakage number, False Alarming Ratio(FAR), and a reasonable identification effect for Salt and pepper noise. The experimental results presented in this paper are limited to a small set of images, which may not represent the wide range of image types and noise types that could be encountered in practice. Liu et al. [13] introduced a new method to classify four types of noises (Gaussian, Poisson, Salt & Pepper, and Speckle) based on deep neural networks. The application of neural networks involves feature extraction, activation function, and network training. According to their finding, the accuracy of the CNN model was 93.7%. The classification model achieves a reasonable recognition rate for single or more types of noises. The drawback of this paper only considers three types of image noise (Gaussian, impulse, and speckle), which may not cover all the noise types that could be encountered in practice. Chuah et al. [14] presents a Convolutional Neural Network (CNN) model to effectively recognize the presence of Gaussian noise and its level in images. They used 12000 and 3000 standard test images for training and testing purposes. Different noise levels are introduced to these images. The overall accuracy was 74.7%. The research only considers Gaussian noise, which may not cover all the noise types that could be encountered in practice.

In comparison to the existing studies, our research presents a novel and comprehensive approach to image noise detection and classification. Unlike previous works that considered a limited number of noise types, we aim to recognize a broader spectrum of noise types that may arise from various image acquisition and environmental conditions.

To achieve this, we propose a combination of deep wavelet techniques and machine learning algorithms to accurately identify and classify multiple types of image noise. This integrated approach takes advantage of both the powerful feature extraction capabilities of deep wavelet analysis and the ability of machine learning models to generalize from data.

3. RESEARCH TOOLS

Many tools (algorithms and techniques) are used in this proposal, and in this section, a brief explanation will be presented.

3.1 CONVOLUTION NEURAL NETWORK

CNN or ConvNet is an Artificial Neural Network used to analyze visual images. CNNs have a shared-weight architecture of the filters or convolution kernels that slide through input features and provide translation-equivariant responses called activation maps [15]. Each CNN consists of an input, hidden, and output layer, and the input layer includes the input matrix. The hidden layers have layers that convolute the input and output after masking them by the activation function [16]. The hidden layers consist of convolutional, pooling, normalization, flattened, and fully connected. The activation function in the hidden layers is typically Rectified Linear Unit (ReLU). The output layer includes the final classification matrix [17].

3.2 DEEP WAVELET TRANSFORMS

Wavelet approaches are practical tools for good data representations and feature extractions compatible with the most available classification algorithms. In conjunction with a deep neural network, the wavelet transform permits the generation of dependable features that are locally stable to tiny deformations. A deep wavelet consists of many layers, where one layer's output serves as the next layer's input. Each layer consists of three operations, as shown in Figure 1 [18], [19].

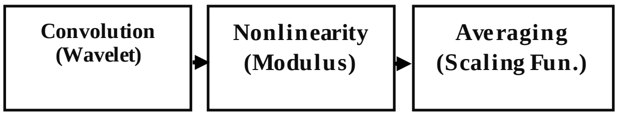


FIGURE 1. - Operations of Wavelet scatter

First-layer scattering coefficients are obtained by convolving the input signal or image (X) with a wavelet filter ($\psi\lambda1$) at a certain scale (J) and taking the modulus of the resulting coefficients, followed by low-pass filtering with a scaling filter (φJ). The coefficients is determined by Eq. (1).

$$S1, J(\lambda1, x) = |X * \psi\lambda1| * \varphi J(x) \qquad (1)$$

Second-layer scattering coefficients are obtained by convolving the input signal or image (X) with a wavelet filter ($\psi\lambda1$) at a certain scale (J), taking the modulus of the resulting coefficients, followed by convolving again with another wavelet filter ($\psi\lambda2$) at a different scale and taking the modulus, and finally low-pass filtering with a scaling filter (φJ). The coefficients is determined by Eq. (2).

$$S2, J\left((\lambda1, \lambda2), x\right) = \left||X * \psi\lambda1| * \psi\lambda2\right| * \varphi J\ (x) \qquad (2) \quad \text{for } 1 < J \leq 2.$$

Note that the parameter (J) is the width of the low-pass filter, determines the length of local translation invariance, as well as the number of scales available from the transform, φ denotes a low-pass filter, ψ denoted wavelet, λ is the rotation operations, since S1, J and S2, J are the outputs of low-pass filters, they can be down-sampled according to the filter width 2 power J [20], [21].

3.3 MACHINE LEARNING (ML)

Is a part of computer vision and Artificial Intelligence (AI) that focuses on making machines learn without being directly programmed. It focuses on scientific goals and applications that depend on optimization and prediction. ML can be divided into three groups (supervised, unsupervised, and reinforcement learning) [22].

In this proposal, we use many machine learning classifiers and ultimately identify the optimum one. The suggested machine learning used in this proposal are:

• Support Vector Machine (SVM) Classifier:

An SVM is a supervised learning model with learning algorithms that look at data to classify it and predict what will happen next. SVMs are one of the best ways to make predictions because they use statistical learning frameworks or the theory of Vector Clustering (VC). The SVM training algorithm makes a model that gives new instances of a flag to tell one class from another [23].

• Random Forest (RF) Classifier:

RF is a well-known and influential ensemble-supervised classification algorithm [4]. RF has been efficiently used in several machine learning applications, including many in bioinformatics and medical imaging, due to its high accuracy, resilience, and capacity to offer insights by ranking its features. RF comprises a set of decision trees, each formed by the bagging algorithm without pruning, making a "forest" of classifiers voting for a specific class [24].

• Decision tree (DT) Classifier:

A decision tree classifier is a quick and effective method for categorizing instances in massive datasets with numerous variables. There are two primary concerns in the building of decision trees: (a) the growth of the tree to classify the training dataset effectively and (b) the pruning stage, in which unnecessary nodes and branches are deleted to increase classification accuracy [25].

• Logistic Regression (LR) Classifier:

Logistic regression (LR) is a method of statistical modeling in which the probability of a category is linked to a set of variables that explain it [26]. LR can also solve problems with more than one class, which can be done differently. One option is to replace the sigmoid function with a softmax function considering more than one class. You can also use the ones-all method, which involves doing a binary LR for each category and counting the data of the desired class as one class and all the other points as another class [27].

• K-Nearest Neighbor (KNN) Classifier:

KNN is one of the most straightforward machine learning algorithms based on supervised learning. It assumes the similarity between the new and existing cases, places the recent case in the category most similar to the existing categories, stores all the relevant data, and classifies a unique data point based on the similarity. When new data emerges, it can be quickly classified into a suitable category using the K-NN technique. This approach may be used for regression and classification problems but is predominantly utilized for classification issues [28].

3.4 PERFORMANCE MEASURES

To evaluate the performance of the proposed models, some measures were used, such as accuracy, recall, precision, and F1-score, whose values can be found by building the confusion matrix of the proposed model.

• Confusion Matrix

The confusion matrix is a cross table that records the number of occurrences between two rates, the true/actual Classification, and the predicted Classification [29].

• Accuracy

Accuracy evaluates the overall effectiveness of any algorithm. Accuracy is determined by Eq. (3) [30].

$$\text{Accuracy} = \frac{\sum_{i=1}^{n}(TP_i + TN_i)}{\text{Total}} \times 100\% \qquad (3)$$

• Recall

The recall is a measure used to evaluate the effectiveness of an algorithm to identify positive labels based on the actual. A recall is determined by Eq. (4) [31].

$$\text{Recall} = \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n}(TP_i + FN_i)} \times 100\% \qquad (4)$$

• Precision

The precision is a measure used to evaluate the class agreement of the data labels with the positive labels given by the algorithm based on the prediction. Precision is determined by Eq. (5) [32].

$$\text{Precision} = \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n}(TP_i + FP_i)} \times 100\% \qquad (5)$$

• F1-Score

The F1-Score is a measure used to evaluate the relations between data's positive labels and those given by an algorithm. An F1-Score is determined by Eq. (6) [33].

$$F1 - \text{Score} = \frac{2(\text{Recall} \times \text{Precision})}{(\text{Recall} + \text{Precision})} \qquad (6)$$

4. METHODOLOGY

The details of the proposed method to detect and classify the noise type are presented in this section. The flowchart of the proposed method is shown in Figure 2. This proposal consists of two main stages, checking whether the image is noisy or not using Convolutional neural networks and then classifying the type of noise in a noisy image based on a mix of deep wavelet and machine learning classifiers (Support Vector Machine (SVM)). The suggested approach is effective at classifying five types of noise, including Gaussian, lognormal, Rayleigh, Salt & Pepper, and Speckle.

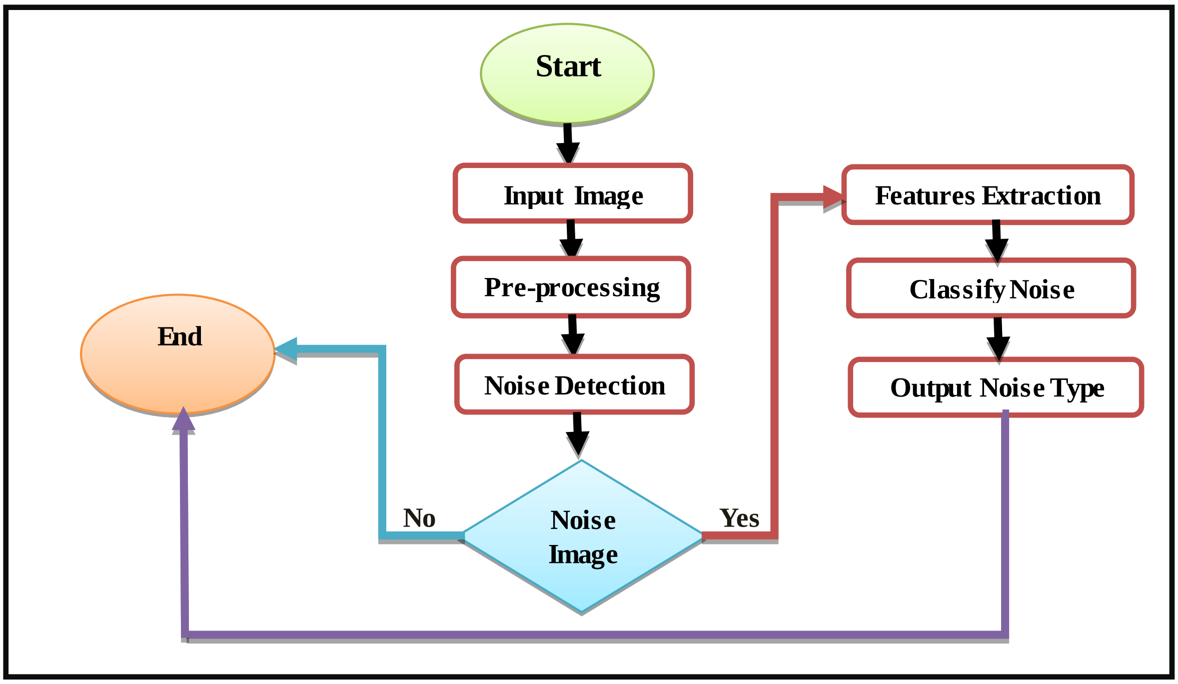


FIGURE 2. - Flow chart of the proposed method

4.1 NOISE DETECTION

This model aims to distinguish the noisy image from the clear image based on Convolutional Neural Network. The input image is pre-processed and then input into the CNN network to detect noise in an image, as shown in Figure 3.

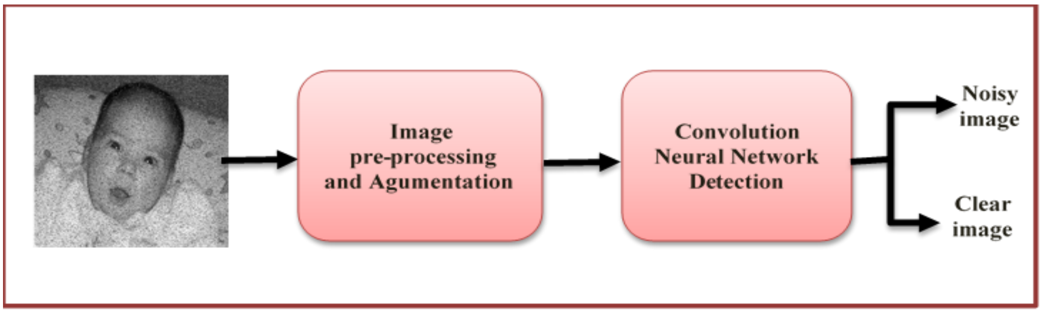


FIGURE 3. - Structure of noise detection

4.1.1 PRE-PROCESSING

In the pre-processing first, we have to prepare the dataset. The dataset used in this proposal consists of 22,000 images collected from many datasets. The images in this dataset are noisy (different noise types) and clear images. The dataset was divided into two sets, one for training and the other for testing, as shown in Table 1. Also, in this step, all the training images are converted into grayscale, resized to 150×150, and applied Augmentation.

Table 1. - Shows the number of images used for training and testing the detection model

Index	Type	No. the dataset	No. of training	No. of testing
1	noisy	14000	12600	1400
2	clear	8000	7200	800
Total		22000	19800	2200

4.1.2 DATA AUGMENTATION

Increase the number of images in the training dataset in real-time during training by using different shapes for one image to get rid of overfitting. The image augments that are applied have a shear range = 0.1, zoom range = 0.1, and horizontal flip.

4.1.3 PROPOSED CONVOLUTION NEURAL NETWORK

The proposed CNN consists of five convolutional layers, ReLU as an activation function to convert linear data to nonlinear data, a kernel with size (3×3), and five max-pooling layers. The five convolutional layers contain (16, 32, 64, 128, and 256) filters, as shown in Figure 4. The max-pooling layer follows each layer. After that, the features map is input to flatten the layer to convert it from 2D to 1D, followed by two fully connected (FC), where the first FC consists of (512) channels and the second FC consists of one channel. We suggested using Sigmoid as an activation function. The details of the proposed CNN architecture for detection noise in an image are shown in Table 2. The number of epochs used in this proposal is equal to 20.

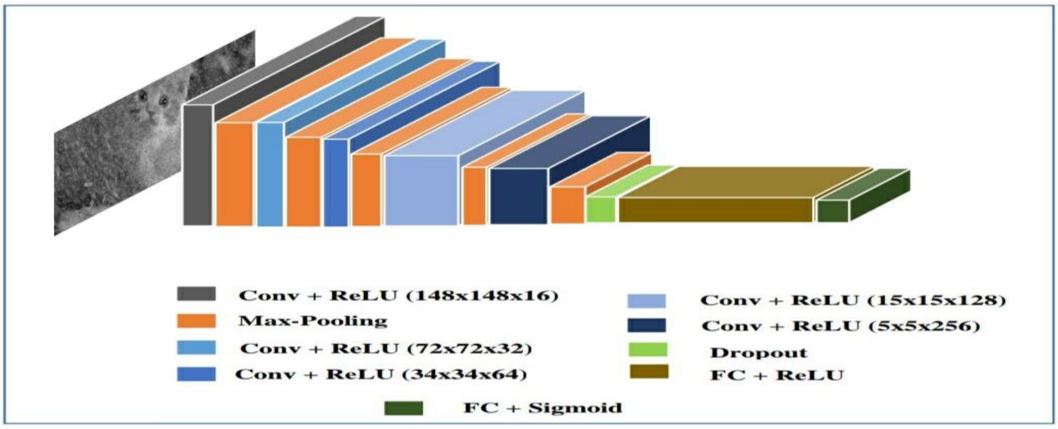


FIGURE 4. - The architecture of Proposed CNN

Table 2. - The architecture of Proposed CNN for Detection

Layer (type)	Output Shape	Number of Parameters
Input Layer	(150,150,1)	0
conv2d (16 filters)	(148,148,16)	160
MaxPooling2D	(74,74,16)	0
conv2d (32 filters)	(72,72,32)	4640
MaxPooling2D	(36,36,32)	0
conv2d (64 filters)	(34,34,64)	18496
MaxPooling2D	(17,17,64)	0
conv2d (128 filters)	(15,15,128)	73856
MaxPooling2D	(7,7,128)	0
conv2d (256 filters)	(5,5,256)	295168
MaxPooling2D	(2,2,256)	0
Dropout	(2,2,256)	0
Flatten	(1024)	0
Dense	(512)	524800
Dense	(1)	513
Total parameters: 917,633		
Trainable parameters: 917,633		
Non-trainable parameters: 0		

4.2 NOISE CLASSIFICATION

After completing the noise detection, it is time to determine the type of noise using a combination of wavelet scatter and machine learning classifiers to classify five types of noise (Gaussian, lognormal, Rayleigh, Salt & Pepper, and Speckle). The combination of deep wavelet and SVM for noise classification involves using deep wavelet to extract features from the signal and then used as input to the SVM algorithm, which learns to classify the noise type depending on the characteristics extracted. To extract features regarding the designation of noise types, the flowchart of the proposed method is shown in Figure 5. The Deep wavelet for noise classification that is shown in Algorithm 1.

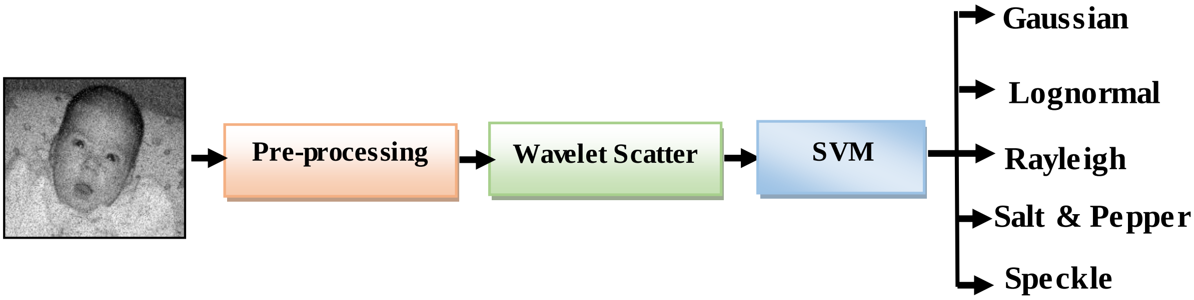


FIGURE 5. - Structure of noise classification

Algorithm 1: Noise Classification	
Input:	Pre-processed images.
Output:	Classified noise.
Begin:	
Step 1:	Input the images to the proposed Deep wavelet
Step 2:	For each image in the training set. a) Calculate the wavelet scattering coefficients. b) Flatten the scattering coefficients into a 1D feature vector. c) Add the feature vector and corresponding label to the training feature matrix and label vector.
Step 3:	Define a set of hyperparameters to be tuned
Step 4:	Training an SVM classifier with the current set of hyperparameters on the training feature.
Step 5:	Compute the error for all trainable and hyperparameters.
Step 6:	Repeating steps 4 and step5 until reaching the minimal error possible.
Step 7:	Selecting the hyperparameter combination that has the least amount of error.
Step 8:	Train an SVM classifier on the entire training feature using the selected hyperparameter values.
Step 9:	Evaluate the classification accuracy on the testing set by comparing the predicted labels to the true labels.
Step 10:	Predict images and classification.
End.	

4.2.1 PRE-PROCESSING

This step first focuses on preparing the dataset. The dataset used in this proposal is (9_classes_noisy_image_dataset) from Kaggle. The total number of images used for this proposal was 1100 images with a size of 600×464. The dataset divides into two sets, one for training and the other for testing, as shown in Table 3.

Table 3. - Images used for training and testing

Index	Noise type	Number of images	No. of training	No. of testing
1	**Gaussian**	200	181	19
2	**lognormal**	250	222	28
3	**Rayleigh**	250	228	22
4	**Salt & Pepper**	200	175	25
5	**Speckle**	200	179	21
	Total	**1100**	**985**	**115**

The pre-processing step also includes labeling the training images and resizing the image from the original size (600×464) into (28×28), the optimum size for a deep wavelet.

4.2.2 PROPOSED DEEP WAVELET TRANSFORMS

A deep wavelet network is a mathematical technique used for signal processing and feature extraction. It works by decomposing a signal into a set of wavelets that are defined by a specific frequency and time domain. The two important factors are (L and J). L is the depth of the wavelet transform and refers to the number of times the signal is decomposed into wavelets. Each level of decomposition captures the features of the signal at a different scale. The higher the depth of the wavelet transform, the finer the detail that can be extracted from the signal. In this proposal, the number of layers used is (8) layer. The decomposition process is typically performed using a filter bank that consists of a low-pass filter and a high-pass filter. The low-pass filter is used to extract the low-frequency components of the signal, while the high-pass filter is used to extract the high-frequency components of the signal. where (J) is the scaling factor used to determine the level of signal analysis. J represents the number of times the wavelet transform is applied, and increases with the desired level of detail to be obtained, in this proposal (J) equal 3. The wavelet coefficients represent the frequency and time-domain characteristics of the signal at different scales (refers to the frequency range that a particular wavelet coefficient represents). These coefficients can be analyzed to extract useful features from the signal. Feature extraction involves selecting the most significant coefficients that capture the important characteristics of the signal. These features can then be used to train a machine learning model to classify the signals based on their genre.

The architecture of the proposed wavelet model for Classification has summarized in Table 4. The number of epochs used in this proposal is equal to 50, where the batch size was 64.

Table 4. - The architecture of the Proposed wavelet for Classification

Layer (type)	Output Shape	Number of Parameters
Input Layer	(28,28)	0
Scattering2D	(217,3,3)	0
Flatten	(1953)	0
Dense	(512)	1000448
Dense	(5)	2565

Total parameters: 1,003,013

Trainable parameters: 1,003,013

Non-trainable parameters: 0

5. RESULTS AND DISCUSSION

The performance of the proposed is measured by using many experimental tests.

5.1 PERFORMANCE OF THE PROPOSED CNN

The first test is to measure the performance of the noise detection method. A noisy image is detected by using the CNN network before recognition of the noise type. This test used (200 images, 100 were noisy images, while the other 100 were clear images without noise). The accuracy achieved was 98%, recall 100%, precision 96%, and the F1 score for this test was 98%, as shown in the confusion matrix shown in Table 5. In this test, just four clear images are classified as noisy images, this is almost better than noisy images classified as a clear image. This case may happen due to blur images or low-quality images.

Table 5. - Confusion matrix for noisy image detection

		Prediction value	
		Noisy image	**Clear image**
	Noisy image	100	0
Actual Value	**Clear image**	4	96

In addition, we measured the optimal number of epochs and the number of folds that can give the best performance. The accuracy is measured when we use a different number of epochs and a different number of folds, and the results are shown in Figure 6. It is noted that the best result is achieved when the number of epochs is equal to 20 and the number of folds similar to five.

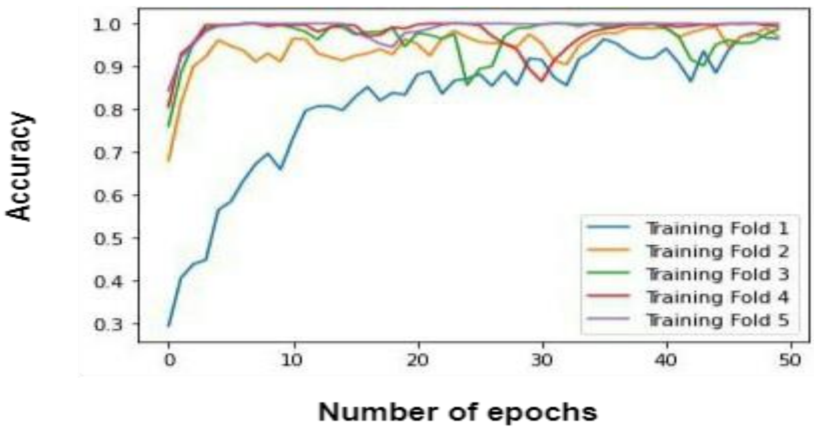


FIGURE 6. - The accuracy vs. epochs according to K-fold cross-validation

5.2 PERFORMANCE OF THE CLASSIFICATION MODEL

The performance of the Classification of the noise type is one of the necessary tests. The Classification proposed method was tested by using 115 images with five different noise types, as shown in Table 1. The results are shown in the confusion matrix in Figure 7. Table 6 lists the classification report accuracy for every noise. The performance measurements are (recall, precision, F1 score, and support).

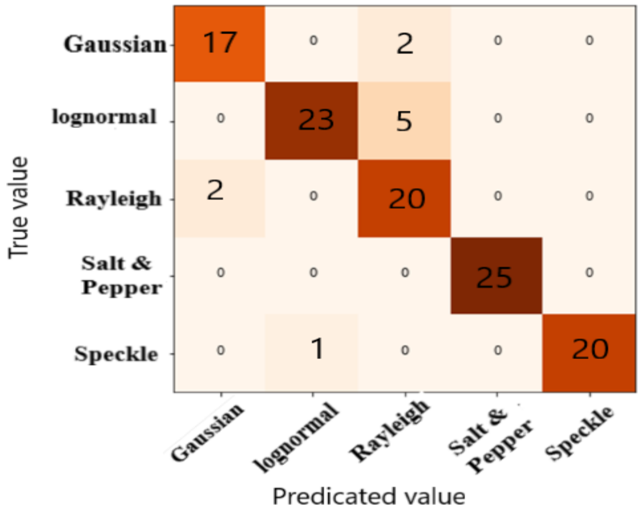


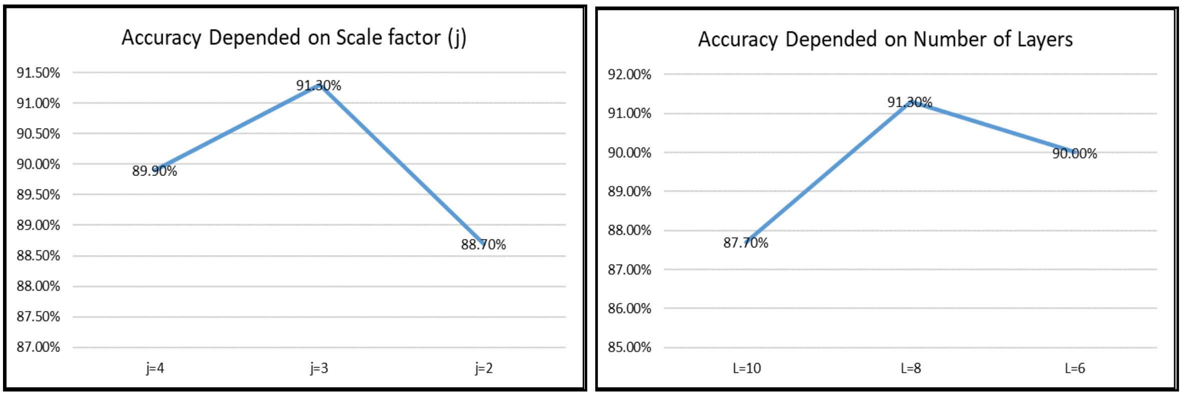
FIGURE 7. - A Confusion Matrix of a Classification Model

The overall accuracy for the proposed classification model for five noise types was 91.30% in testing with new images not included in the training. The significant similarity between the selected noise types, a challenge for most researchers, reduces the classification accuracy.

Table 6. - Classification Report of the proposed method for five noise types

Noise type	Precision	Recall	F1-Score	Support
Gaussian	89%	89%	89%	19
Lognormal	96%	82%	88%	28
Rayleigh	74%	91%	82%	22
salt & pepper	100%	100%	100%	25
Speckle	100%	95%	98%	21
Accuracy	91%			115
Marco avg	92%	92%	91%	115
Weighted avg	92%	91%	91%	115

The maximum number of wavelet scales in the scaling factor (j) and The number of wavelet layers (L) used in the wavelet scatter model was tested to determine the best value, which gives a more accurate classification, as shown in Figure 8.



(a) (b)

FIGURE 8. - (a) The Relationship between The Scaling factor and accuracy. (b) the relationship between the number of wavelet layers and accuracy

As mentioned in the methodology, we suggested using five machine learning combined with wavelet scatter. For that, we test the performance of combination wavelet scatter with each proposed machine learning. The results are listed in Table 7. We conclude that combining the wavelet scatter with SVM gives the best results, although the other combination methods give promised results. Using SVM with wavelet scatter is the best choice for the Classification of noise types, so we recommend to used SVM with wavelet scatter for the Classification of noise-type contaminated images.

Table 7. - Performance of the suggested classification methods

Method	Accuracy
Wavelet + SVM	91.30%
Wavelet + RF	88.60%
Wavelet + LR	88.60%
Wavelet + KNN	88.69%
Wavelet + DT	88.69%

To determine the robustness of the proposed method, we try to compare it with other techniques, all implemented using the same dataset and environment. We measured the performance of machine learning without combination with another algorithm, and we selected some of the public deep learning classifiers, as shown in Table 8. We conclude that the combination of wavelet scatter with machine learning performs better than the deep learning methods and the machine learning classifiers.

Table 8. - Comparison accuracy between the CNN network, machine classifier, and the proposed method

	Methods	Accuracy
CNN Networks	Fine-Tuned VGG16+SVM	83.22%
	Inception v3	77.33%
	VGG-16	71.85%
	Fine-Tuned ResNet50	81.33%
	Fine-Tuned VGG19	85.56%
Machine Classifier	SVM	53%
	RF	85.2%
	LR	30.4%
	KNN	85.2%
	DT	85.2%
Proposed method	**Wavelet + SVM**	**91.30%**

Finally, we tried to find some methods that classify noise types to compare our proposed method with them, but unfortunately, we found just the papers listed in Table 9. To make a fair comparison, we used the same dataset used by each paper and classified the same type of noise for each one. The results proved the high efficiency of the proposed method.

Table 9. - Comparison accuracy between the proposed method with other research

Ref.	**Type of classes**	**method**	**Accuracy**	**Accuracy of the proposed model**
[1]	1-Salt & pepper 2-Gaussian	neural network	94.37%	99.37%
[4]	1-Gaussian 2-Poisson 3-Salt & Pepper 4-Speckle	deep neural networks	93.7%	95.83%
[5]	1-Gaussian and its level	CNN	74.7%	93.60%

6. CONCLUSION

This paper's main contribution is detecting whether the images are noisy or clear. A new branch in the field is the classification of five types of noise (Gaussian, lognormal, Rayleigh, Salt and Pepper, and Speckle). Also, using a wavelet scatter model and machine classifiers opens new horizons in this field and other fields. To our knowledge, few papers have worked on classifying two or three types of image noise. The results show good performance with this proposal. The accuracy for noise detection and classification has been achieved at 98% and 91.30%, respectively. As a result, it was achieved and by addressing the suggested future work, we can continue to advance image noise research and its practical applications, ultimately benefiting a wide range of domains that rely on accurate and reliable image analysis. It is recommended to increase the noise types by combining wavelet scatter and other classifiers.

FUNDING

No funding received for this work

ACKNOWLEDGEMENT

I would like to thank the reviewers, editors and staff who contributed by reviewing my study.

CONFLICTS OF INTEREST

The authors declare no conflict of interest

REFERENCES

[1] Q. Wang, J. Ma, S. Yu, and L. Tan, "Noise detection and image denoising based on fractional calculus," Chaos, Solitons & Fractals, pp. 1–11, 2020, doi: 10.1016/j.chaos.2019.109463.

[2] H. Y. Khaw, F. C. Soon, J. H. Chuah, and C. O. Chow, "Image noise types recognition using a convolutional neural network with principal components analysis," IET Image Process, pp. 1238–1245, 2017, doi: 10.1080/16864360.2021.1892846.

[3] D. Sil, A. Dutta, and A. Chandra, "Convolutional Neural Networks for Noise Classification and Denoising of Images," TENCON IEEE Region 10 Conference (TENCON), pp. 447–451, 2019, doi: 10.1049/iet-ipr.2017.0374.

[4] S. Patil and S. S. Sherekar, "Noise detection and classification using machine learning: A review," J. Ambient Intell. Humanized Comput., vol. 10, no. 1, pp. 169–183, 2019, doi: 10.1007/s12652-018-0896-2.
[5] X. Zhang, L. Hu, and X. Lu, "A review on noise classification techniques based on machine learning," Appl. Sci., vol. 9, no. 22, p. 4794, 2019, doi: 10.3390/app9224794.
[6] S. Kim, "Noise Classification Techniques for Urban Sound Environment: A Comprehensive Survey," IEEE Access, vol. 9, pp. 10943–10957, 2021, doi: 10.1109/ACCESS.2021.3054587.
[7] Y. Li, G. Wang, J. Li, B. Li, and H. Yang, "Noise robust speaker verification based on deep neural networks," IEEE/ACM Trans. Audio, Speech, Lang. Process., vol. 27, no. 2, pp. 385–398, Feb. 2019, doi: 10.1109/TASLP.2018.2883208.
[8] S. Baccour, K. Ouni, and A. M. Alimi, "Real-time environmental sound recognition: Challenges and approaches," Appl. Acoust., vol. 109, pp. 117–128, 2016, doi: 10.1016/j.apacoust.2016.01.003.
[9] D. Barchiesi, D. Giannoulis, and D. Stowell, "Acoustic classification of multiple simultaneous bird species: A multi-label approach and sound selection method," J. Acoust. Soc. Am., vol. 138, no. 2, pp. 892–904, 2015, doi: 10.1121/1.4926443.
[10] P. S. Hiremath, "Identification of Noise in an Image using Artificial Neural Network," Int. J. Eng. Res. Technol., vol. 10, no. 2, pp. 345–348, 2021.
[11] M. Tripathi, "Facial image noise classification and denoising using neural network," Sustain. Eng. Innov., vol. 3, no. 2, pp. 102–111, 2021, doi: 10.37868/sei.v3i2.id142.
[12] H. Yi and M. XiaoXuan, "Image noise recognition algorithm based on BP neural network," 2020 Chinese Control and Decision Conference (CCDC), Hefei, China, pp. 106–111, 2020.
[13] F. Liu, Q. Song, and G. Jin, "The classification and denoising of image noise based on deep neural networks," IEEE Access, vol. 7, pp. 2194–2207, 2020, doi: 10.1007/s10489-019-01623-0.
[14] J. H. Chuah, H. Y. Khaw, F. C. Soon, and C. Chow, "Detection of Gaussian Noise and Its Level using Deep Convolutional Neural Network," Proc. of the 2017 IEEE Region 10 Conference (TENCON), Malaysia, Nov. 5–8, pp. 2447–2450, 2017.
[15] T. Bergs, C. Holst, P. Gupta, and T. Augspurger, "Digital image processing with deep learning for automated cutting tool wear detection," Procedia Manuf., vol. 48, pp. 947–958, 2020, doi: 10.1016/j.promfg.2020.05.134.
[16] H. Hakim and A. Fadhil, "Survey: Convolution neural networks in object detection," J. Phys. Conf. Ser., vol. 1804, no. 1, 2021, doi: 10.1088/1742-6596/1804/1/012094.
[17] M. Sarigul, B. M. Ozyildirim, and M. Avci, "Differential Convolutional Neural Network," Neural Networks, vol. 116, pp. 279–287, Dec. 2019, doi: 10.1016/j.neunet.2019.04.013.
[18] B. Soro and C. Lee, "A wavelet scattering feature extraction approach for deep neural network-based indoor fingerprinting localization," Sensors (Switzerland), vol. 19, no. 8, 2019, doi: 10.3390/s19081790.
[19] Z. Liu, G. Yao, Q. Zhang, J. Zhang, and X. Zeng, "Wavelet Scattering Transform for ECG Beat Classification," Comput. Math. Methods Med., vol. 2020, 2020, doi: 10.1155/2020/3215681.
[20] E. C. Djamal, R. I. Ramadhan, M. I. Mandasari, and D. Djajasasmita, "Identification of post-stroke EEG signal using wavelet and convolutional neural networks," Bull. Electr. Eng. Inform., vol. 9, no. 5, pp. 1890–1898, 2020, doi: 10.11591/eei.v9i5.2005.
[21] R. Leonarduzzi, H. Liu, and Y. Wang, "Scattering transform and sparse linear classifiers for art authentication," Signal Process., vol. 150, pp. 11–19, 2018, doi: 10.1016/j.sigpro.2018.03.012.
[22] Q. Bi, K. E. Goodman, J. Kaminsky, and J. Lessler, "What is machine learning? A primer for the epidemiologist," Am. J. Epidemiol., vol. 188, no. 12, pp. 2222–2239, 2019, doi: 10.1093/aje/kwz189.
[23] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," ACM Trans. Intell. Syst. Technol., vol. 2, no. 3, pp. 27:1–27:27, Apr. 2011, doi: 10.1145/1961189.1961199.
[24] D. Petkovic, R. Altman, M. Wong, and A. Vigil, "Improving the explainability of Random Forest classifier – User-centered approach," Pacific Symp. Biocomput., vol. 0, no. 212669, pp. 204–215, 2018, doi: 10.1142/9789813235533_0019.
[25] D. M. Farid, L. Zhang, C. M. Rahman, M. A. Hossain, and R. Strachan, "Hybrid decision tree and naïve Bayes classifiers for multi-class classification tasks," Expert Syst. Appl., vol. 41, no. 4 PART 2, pp. 1937–1946, 2014, doi: 10.1016/j.eswa.2013.08.089.
[26] L. Dong, J. Wesseloo, Y. Potvin, and X. Li, "Discrimination of Mine Seismic Events and Blasts Using the Fisher Classifier, Naive Bayesian Classifier, and Logistic Regression," Rock Mech. Rock Eng., vol. 49, no. 1, pp. 183–211, 2016, doi: 10.1007/s00603-015-0733-y.
[27] R. A. de Paula et al., "Mitigation of nonlinear phase noise in single-channel coherent 16-QAM systems employing logistic regression," Opt. Quantum Electron., vol. 53, no. 9, pp. 1–9, 2021, doi: 10.1007/s11082-021-03149-7.
[28] S. Zhang, X. Li, M. Zong, X. Zhu, and D. Cheng, "Learning k for kNN Classification," ACM Trans. Intell. Syst. Technol., vol. 8, no. 3, pp. 1–19, 2017, doi: 10.1145/2990508.

[29] D. Xinyang, W. Haifeng, H. Xiaojun, and W. Liang, "An improved method to construct basic probability assignment based on the confusion matrix for classification problem," Information Sciences, vol. 340, pp. 250–261, 2016.

[30] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," Information Processing & Management, vol. 45, no. 4, pp. 427–437, 2009, doi: 10.1016/j.ipm.20.