

Deep Convolutional Neural Network for SEM Image Noise Variance Classification

Sim Kok Swee, *Member, IAENG*, Lim Choon Chen, Tan Shing Chiang, Toa Chean Khim, *Member, IAENG*

Abstract—Scanning Electron Microscopy (SEM) image plays a significant role in industrial, medical, and research fields. However, image defects, including existing noise will degrade the quality of the original image. In order to denoise the image, noise estimation is crucial. In the past, several noise estimation methods are designed, but they are not effective in estimating the noise level of the SEM images. Therefore, this paper presents a novel method to effectively classify noise variance of SEM images through a deep learning algorithm. The images are classified based on their respective noise variances. Before the classification process begins, the original SEM images are added with 4 categories of White Gaussian noises. To categorize the noise level of the image, a complex Gaussian-Noise Convolutional Neural Network (GN-CNN) method with encoder layer, convolutional layer, attention layer, decoder layer, and decision layer are adopted. In this study, a total of 1200 image data from different variety and complexity (720 images for training, 240 images for validation and testing respectively) are used. The experimental study shows the capability and reliability of the developed system in classifying noise variances of SEM images with the F1 score of 93.97% and testing accuracy of 93.8%. It has outperformed other baseline deep-learning models, including Long Short-Term Memory (LSTM), Residual Network 18 (ResNet18), and Residual Network 34 (ResNet34), and pure Convolutional Neural Network (CNN). Thus, the designed network can impressively surpass human-eye performance in noise variance categorization.

Index Terms—Scanning Electron Microscopy, Convolutional Neural Network, Gaussian, Noise Variance Classification

I. INTRODUCTION

Scanning Electron Microscopy (SEM) image acts as a crucial and imperative role in industrial and research fields. Nevertheless, noises are always intrinsically introduced into a digital image during the transmission of an image [1]. These existing noises can trigger degradation to the quality of grayscale SEM images and cause corruption in the image information [2]. Consequently, some essential information on SEM images will be lost due to the

interference of white noise.

Thus, noise reduction procedure is a crucial preprocessing step to obtain a high-quality SEM image. In this regard, noise reduction acts as a fundamental step to recover the original image [3], [4], [5]. Several existing noise reduction methods are spatial domain filters and transform domain filters [6], [7], [8], [9]. Before the noise reduction process, the determination of the noise variance of an image would play a beneficial role in easing the noise reduction process. Subsequently, noise variance determination of an SEM image is mandatory to quantify the quality of the image, so that the noise reduction process can be easily carried out with a predetermined noise level. Accurate and precise measurement of noise level of a single SEM image brings useful information to the noise filtering process.

In the past, several classical signal processing methods such as shape-preserving piecewise cubic hermite autoregressive moving average (SP2CHARMA) [10], cubic spline hermite interpolation with linear least square regression (CSHILLSR) [11], and adaptive slope nearest-neighbourhood [12] have been proposed for estimating noise level in SEM image. Although these methods can be used for estimation, more efficient methods are still needed. To appraise an image's noise variance, recent research has demonstrated that deep learning algorithm has the capability in the classification of noise variance [1]. Deep learning is very useful for feature learning. This is opposed the classical signal processing methods that use handcrafted algorithms. The handcrafted algorithms apply a series of mathematical equations in features extraction. On the other hand, a neural network can learn related features which are necessary for executing a task. Besides, deep learning methods have shown a better accuracy of 5.4% over the traditional and classical baselines [13]. Moreover, the deep learning methods can detect a minor difference in images with different noises that are difficult to discover by human naked eyes. Thereby, deep learning architecture is applied to replace the manual measurement method.

Convolutional neural network (CNN) is commonly applied in the recognition and classification field. For instance, CNN has been widely implemented in face recognition [14], handwritten recognition [15], image recognition, driving posture recognition [16], vehicle model recognition [17], and so on. Overall, CNN can perform well in the recognition and categorization of images and videos. The input feature for this hierarchical neural network is normally a digital image. By passing through a few convolutional layers with subsampling layers, features are extracted from the image and passed through a fully connected layer to classify noise level.

This paper presents a novel method to effectively classify

Manuscript received 19 March 2022; revised 7 August 2022. This work was supported in part by Multimedia University, Malaysia.

K. S. Sim (Corresponding Author) is a Professor of Faculty of Engineering and Technology, Multimedia University, Jalan Ayer Keroh Lama, 75450 Melaka, Malaysia (e-mail: kssim@mmu.edu.my).

C. C. Lim is a Postgraduate student of Faculty of Engineering and Technology, Multimedia University, Jalan Ayer Keroh Lama, 75450 Melaka, Malaysia (email: junqinglim@gmail.com)

S. C. Tan is a Professor of Faculty of Information Science and Technology, Multimedia University, Jalan Ayer Keroh Lama, 75450 Melaka, Malaysia (email: sctan@mmu.edu.my)

C. K. Toa is a Postgraduate student of Faculty of Engineering and Technology, Multimedia University, Jalan Ayer Keroh Lama, 75450 Melaka, Malaysia (e-mail: toacheankhim@yahoo.com)

noise variance of SEM images through a deep learning algorithm. Our work employs Gaussian-Noise Convolutional Neural Network (GN-CNN) method which is different from other deep learning architectures such as recurrent neural network (RNN), pure convolutional neural network (CNN), pure long short term memory (LSTM), and generative adversarial network (GAN). The conventional CNN has been converted and remodeled into GN-CNN that comprises the encoder layer, convolutional layer, attention layer, decoder layer, and decision layer. Bi-directional LSTM is employed in the encoder layer, ResNet34 is used in the convolutional layer, and LSTM cell is applied in the decoder layer. The key contribution of GN-CNN lies in solving the problem of classifying high nonlinearity images with different noise variances. The slight and insignificant difference in noised SEM images could be difficult to differentiate and classify visually. This research is conducted by first adding white Gaussian noise to an original SEM image. Gaussian is considered in this research because noise is often a result of summing a large number of independent and different factors, and the sum of many random variables can be well approximated by a Gaussian distribution, which is why the Gaussian noise is a simple yet powerful model [18].

In this paper, the Gaussian noise variances used are 0, 0.01, 0.02, and 0.03. Noise variance with 0 represents original SEM images. In other words, our proposed method is capable of differentiating corrupted SEM images at different noise levels. The noise variance range is set within 0 to 0.03 since the SEM images with noise variance of 0.035 or above are extensive blur images, which are not applicable in real-life examples. For experimental study purposes, the number of class categories is set to 4 to prove the practicality of GN-CNN in dealing with noisy SEM images. All the SEM images with added noise are converted to a histogram. The deep learning model of GN-CNN is then trained by using the histograms of SEM images. Testing dataset is inserted into the system so that the classification on the histogram of images can be carried out.

Section 2 explains the related works that utilize classical handcrafted formula methods and deep learning algorithms in the noise estimation process. Section 3 describes the proposed methodology of GN-CNN. Experimental results and discussions are included in Section 4 of this paper. Section 5 concludes the findings in this paper.

II. RELATED WORKS

A. Noise Estimation Using Proposed Equations

In 2001, Thong et al. [19] developed a method to predict the signal-to-noise (SNR) ratio of an SEM image. They used the autocorrelation function to process every pixel of an image. Sim et al. [20] presented a new model named as autoregressive (AR) model. The Nearest Neighbourhood (NN) method and Linear Interpolation (LI) method were applied by authors in the AR model to forecast the signal to noise (SNR) value. The LI method employed the points that were located before and after the noisy free peak, r^{int} to execute SNR estimation. The NN method used the points that were the closest to the point at noisy free peak, r^{NF} to predict the SNR.

Sim et al. [10] proposed another new model of Shape-Preserving Cubic Hermite Autoregressive Moving Average (SPCHAMA) in determining noise level. The robustness of the method was proven to supply optimum solutions in noise estimation. Sim and Teh [12] developed an adaptive slope nearest neighborhood (ASNN) method. The method added slope constants into SNR prediction to enhance the overall accuracy. Besides, Gao et al. [21] proved that linear regression was appropriate and suitable for noise prediction as all data points were considered in the graph. Recently, Sim et al. [11] further improved the SEM noise estimation by introducing Cubic Spline Hermite Interpolation with Linear Least Square Regression (CSHILLSR). The method could estimate SNR values that were very close to the actual ones.

B. Noise Classification Using Deep Learning Algorithm

Khaw et al. [11] presented a convolutional neural network principal components analysis (PCA) for noise type recognition (CPNTR) model to recognize images with different types of noises, which were Gaussian noise, Speckle and Poisson noise, impulse and mixture of those noises. The classic CNN was merged with PCA to shorten the training period. Besides, the model was equipped with a backpropagation algorithm and stochastic gradient descent optimization techniques to effectively classify classes of noise. The CNN model contained 2 convolutional layers, 2 pooling layers, and a fully connected layer. Chuah et al. [22] used a baseline model of Convolutional Neural Network (CNN) model to recognize the different noise levels of Gaussian noise in images. The baseline CNN model contained 4 convolutional layers, 2 max-pooling layers, and a fully connected layer. Noise level classification had obtained an overall accuracy of 74.7% in classifying different noise levels of images. Momeny et al. [23] proposed a Noise-Robust Convolution Neural Network (NR-CNN) to classify the noisy images which were corrupted by missing image samples, impulse noise, packet loss, tampered images and damaged images. An adaptive resize layer and a noise map layer were adapted before the CNN architecture. The CNN architecture belonged to baseline model, which consisted of 2 convolutional layers, 2 pooling layers, and a fully connected layer. The research outcomes indicated that their proposed framework outperformed other methods such as VGG-Net-Slow and VGG-Net-Medium in the classification of noisy images. Murphy et al. [24] presented a CNN method to classify the noise level of the input data. Roy et al. [25] developed a robust image classification system that could be used to effectively recognize images that were corrupted with zero, regular, and massive noises. They employed a regular CNN that comprised 2 convolutional layers and 2 subsampling layers. The CNN was then integrated with two denoising auto-encoders (DAEs) to classify the images with massive noise

III. MATERIAL AND METHODS

This paper focuses on classifying noise variance of SEM images through a developed deep learning algorithm of Gaussian-Noise Convolutional Neural Network (GN-CNN). This research is conducted by first adding white Gaussian noise to an original SEM image [26]. The images with added

white Gaussian noise are then each converted into a histogram image so that all the images are in the same format. Images with an identical format can ease the deep learning algorithm in categorizing the noise variance. Next, the images in a histogram format are adopted to train the GN-CNN. New noisy images (testing datasets) are inserted into the system to test its performance.

A. Adding White Gaussian Noise

The equation that is used to corrupt the grayscale SEM images with White Gaussian noise is stated in Equation (1).

$$e(x, y) = A(x, y) + W(x, y), \quad (1)$$

where $A(x, y)$ denotes the actual image signal, $W(x, y)$ denotes the White Gaussian noise, and $e(x, y)$ denotes image function [11]. This equation indicates that the summation of the actual image signal and White Gaussian noise can gain image function.

Noise variance is a determination of existing Gaussian noise level in an SEM image. Generally, the noise variance of a grayscale SEM image ranges from 0.001 to 0.035. The higher the noise variance of the image, the greater the amount of image information lost. In general, the image with noise variance of greater than 0.035 turns the Signal-to-Noise ratio (SNR) value close to 0 [11]. It induces significant losses of essential information and results in a highly blurred image. SEM images with extensive blurriness are rarely applicable in real life, so the noise variance categories are set to 0, 0.01, 0.02, and 0.03. The SEM image with noise variance of 0 represents an original SEM image. The noise variance of 0.01 refers to the low amount of Gaussian noise added, whereas the noise variance of 0.03 refers to the high amount of Gaussian noise added. Equation (2) is utilized to calculate noise variance of a corrupted image.

$$\text{Noise Variance} = \frac{r(0, y) - r^{NF}(0, y)}{\text{Image Resolution}} \quad (2)$$

where $r(0, y)$ is noisy image and $r^{NF}(0, y)$ is noise-free image. Image resolution at the denominator part of the equation indicates the pixel size of an SEM image. For consistency purposes, the resolution of all the SEM images in this paper is set at a pixel size of (256x256) [27]. The SEM images are provided in a bitmap (BMP) image format.

B. Histogram Image Conversion

The grayscale SEM images are then converted to the histograms. A histogram describes the relationship between the intensity value of image and the frequency of occurrence at each specified intensity value. The histogram can be used to address image issues such as overexposure images and noisy images. All the grayscale SEM images with added white Gaussian noise are converted into histogram images so that all the images with identical format ease the deep learning algorithm in clustering the noise variance. Thereby, the input dataset of the proposed neural network method is in a histogram image format. The process is executed in MATLAB software by applying a built-in function of *imhist()*. The horizontal axis of a histogram plot refers to pixel numbers. Equation (3) states the range of pixel value lies in a

histogram.

$$p(u, v) \in [0, K - 1], \quad (3)$$

where $p(u, v)$ denotes the range of pixel values and K denotes maximum pixel size. In this case, an 8-bit grayscale SEM image represents the K value of 256 with the mathematical calculation of $2^8=256$. The vertical axis of histogram refers to frequency of occurrence where the frequencies of SEM images range from 0 to at most 4500. Fig. 1 simplifies the process of adding Gaussian noise and the process of converting to histogram in one flowchart.

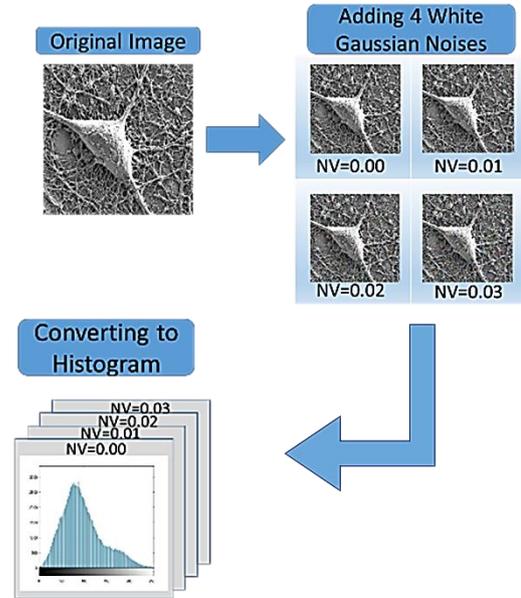


Fig. 1 Image preprocess before application of GN-CNN

C. Gaussian-Noise Convolutional Neural Network

The key contribution of GN-CNN lies in solving the problem of classifying high nonlinearity images with different noise variances. Deep learning algorithms are expected to tackle the problem of human visual classification. Hence, the conventional CNN has been remodeled into GN-CNN.

1. Overall GN-CNN architecture

The GN-CNN comprises 5 major layers of encoder layer, convolution layer, attention layer, decoder layer, and decision layer. The encoder layer consists of bi-directional Long Short-Term Memory (LSTM), whereas the convolutional layer contains Residual Networks 34 (ResNet34). The bi-directional Long Short-Term Memory (LSTM) intends to perform feature extraction on the input images. The convolution layer that consists of ResNet34 can further convolute the features and pass them to the next attention layer. The attention layer includes a soft attention mechanism to focus on essential features to prevent resources wastage. The next layer of decoder layer consists of a Long Short-

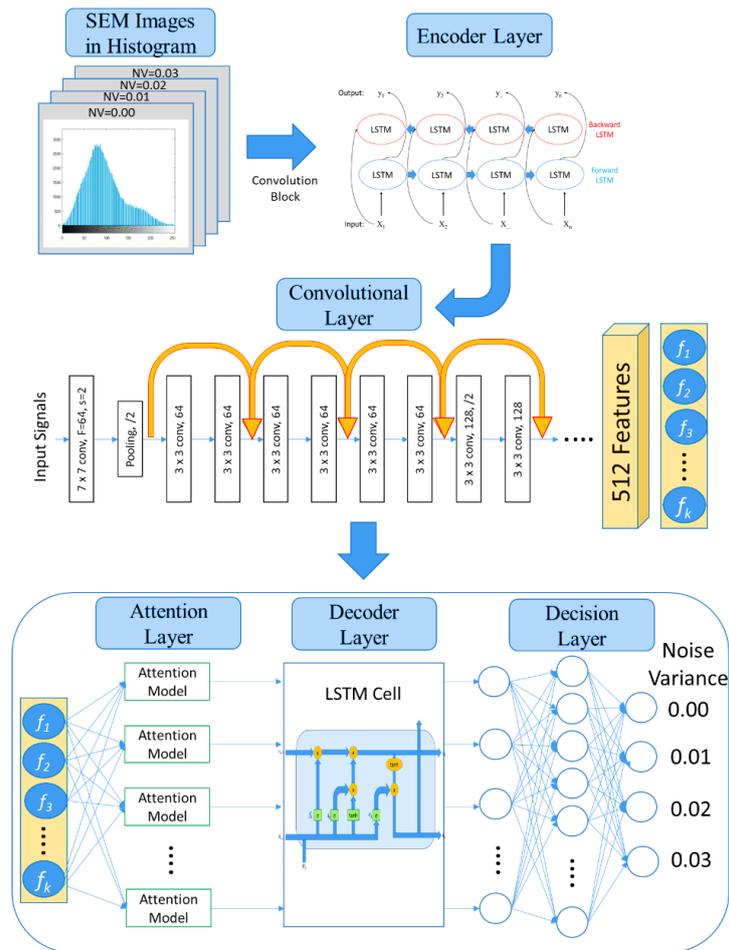


Fig. 2 Overall flow chart of the proposed GN-CNN architecture

Term Memory (LSTM) cell. The Long Short-Term Memory (LSTM) cell has 3 important gates in managing the information flow.

The output from the decoder layer is passed to the final layer of the decision layer to efficiently classify the SEM images to the correct noise variance category. The decision layer contains fully connected layer. Fig. 2 illustrates the overall flow chart of the proposed GN-CNN approach which consists of 5 layers from the encoder layer until the decision layer.

2. Encoder Layer

Encoder is implemented at the beginning process of neural network architecture to perform feature extraction on the input dataset. In this regard, LSTM network is a powerful tool in encoding input information since it can remember precious information and estimate the next result [28]. A bi-directional LSTM is used as an encoder as it encodes sequences in both forward and backward directions [29]. All the SEM images in the histogram format are resized to a size of 700 x 700 for consistency purposes [30]. Image of 700 x 700 in size is a squared size image. For simplicity purposes, usually deep learning models prefer a squared size image as input to the architecture. Before proceeding to the bidirectional LSTM, the image will initially go through the convolutional block to apply the kernel to the 2-dimensional image and extract the features. Since the image is in grayscale, only 1 input channel is used. The kernel size used is 3x3 with a stride of 2. The output of the channel is set to 64 which means that it will

extract 64 feature maps using 64 kernels. After that, these feature maps will be inserted into the bidirectional LSTM. The number of the layer is set to 1 and the hidden size is set to 64. As the LSTM is bidirectional, there will be a forward LSTM and a backward LSTM, so the output of feature maps will be 128. Fig. 3 displays the architecture of the encoder layer that consists of convolutional block and bi-directional LSTM.

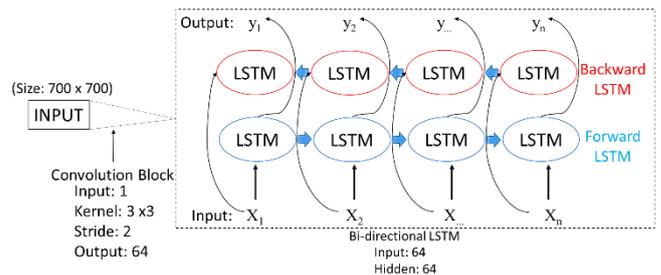


Fig. 3 Encoder layer consists of convolutional block and bi-directional LSTM

3. Convolutional Layer

The outcome features in the encoder layer will be passed to the convolutional layer. This layer consists of Residual Networks 34 (ResNet34). We opt to use ResNet34 in this layer since it is one of the most commonly used convolutional neural networks. Besides, ResNet34 is selected rather than ResNet18 since the training error of ResNet34 is lower than that of ResNet18 [31]. ResNet34 architecture can also reduce computational cost and training time as compared to other

deeper residual networks such as ResNet50. ResNet34 comprises learned rich features that bring advantageous benefits to the classification of images with a wide range of variety. By creating shortcut connections to constantly skip two convolution layers, it can enhance the dimensions [31]. The convolution kernel size in the first layer of ResNet34 is fixed at 7×7 , followed by 3×3 in the pooling layer. The last layer of processing feature map is the average pooling layer. Table 1 describes some parameters of ResNet34 in each layer.

Table 1 Parameters of ResNet34 in each layer (He et al. 2016)

Layer Name	ResNet34
conv1	7×7 , 64, stride = 2, padding = 3
pool1	3×3 , max pool, stride = 2
conv2_x	$\begin{bmatrix} 3 \times 3, & 64 \\ 3 \times 3, & 64 \end{bmatrix} \times 3$
conv3_x	$\begin{bmatrix} 3 \times 3, & 128 \\ 3 \times 3, & 128 \end{bmatrix} \times 4$
conv4_x	$\begin{bmatrix} 3 \times 3, & 256 \\ 3 \times 3, & 256 \end{bmatrix} \times 6$
conv5_x	$\begin{bmatrix} 3 \times 3, & 512 \\ 3 \times 3, & 512 \end{bmatrix} \times 3$
average pool	7×7 average pool

4. Attention Layer

Next, the convoluted features are passed to the next attention layer. A soft attention mechanism is a common technique to aid in the image classification process. It is employed to isolate the content of images in a better performance [32]. An image usually possesses unnecessary background which acts an insignificant role in the deep learning classification mechanism. The conventional CNN provides an equivalent amount of computational resources to the whole part of the image. This would waste enormous time and effort in recognizing an image. Consequently, the attention layer is adopted to supply additional attention by introducing an attention gate that provides assistance in concentrating on the selective part. Chu et al. [33] have proven that the classification model with an attention layer has higher accuracy than the model without an attention layer. In this layer, 3 single feed-forward networks, rectified linear unit (ReLU) function, and softmax function are applied in sequence to obtain the attention weighted vector. This vector is computed for the usage in the decoder layer. Equation (4) to Equation (7) explain the method in calculating the attention weighted vector. Equation (6) indicates the softmax function.

$$f_k = c_k + h_{k-1}, \quad (4)$$

$$a_k = \begin{cases} 0 & \text{for } f_k < 0 \\ f_k & \text{for } f_k \geq 0 \end{cases}, \quad (5)$$

$$\theta_k = \frac{e^{a_k}}{\sum_k e^{a_k}}, \quad (6)$$

$$W_k = \sum_k \theta_k c_k, \quad (7)$$

where c_k represents convoluted features, h_{k-1} refers to hidden state in the previous layer, a_k indicates aggression value, θ_k denotes the weight of aggression value, and W_k represents

attention weighted vector.

5. Decoder Layer

In the final layer of GN-CNN, the Long Short Term Memory (LSTM) cell is opted to serve as decoder architecture. LSTM acts a crucial role to perform the classification of noise variance by providing a final categorization value (either true or false). The LSTM cell consists of a sigmoid function which serves as fundamental in the control of information flow [34]. Besides, it has the ability to ignore insignificant information, so it saves some computational resources. The LSTM cell comprises 3 logical gates of input gate (i_k), output gate (o_k), and forget gate (f_k) as demonstrated in Fig. 4.

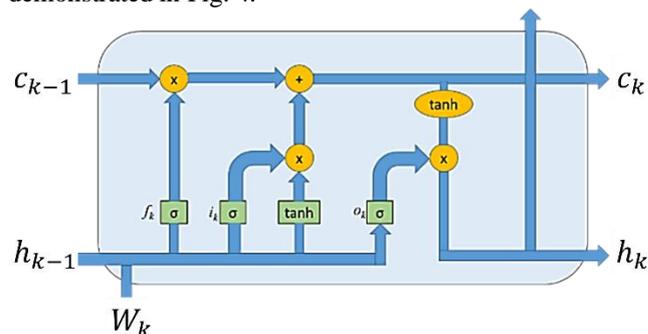


Fig. 4 Working mechanism of 3 gates in LSTM cell

Analog signals available in these 3 gates are ranged from 0 to 1. In the forget gate, LSTM decides which information to forget or to maintain. On this account, the output of LSTM cell of 0 signifies that the piece of information can be completely forgotten, whereas 1 denotes that the piece of information is essential, and it has to be maintained. Equation (8) to Equation (10) show the formulas of the three gates in details.

$$f_k = \sigma(M_f W_k + M_f h_{k-1} + b_f), \quad (8)$$

$$i_k = \sigma(M_i W_k + M_i h_{k-1} + b_i), \quad (9)$$

$$o_k = \sigma(M_o W_k + M_o h_{k-1} + b_o), \quad (10)$$

where f_k , i_k and o_k refer to notation for forget, input and output vector respectively. M represents weight metric, W_k represents attention weighted vector, h_{k-1} represents hidden state, and b represents bias.

6. Decision Layer

Decision layer is allocated at the last layer of GN-CNN to decide the classification of SEM images according to the respective noise variance. A dropout rate of 0.5 is applied in this layer. Fully connected layer is implemented in this layer to recognize the images and classify them into 4 main categories. The output of the GN-CNN system is termed in predictive score with a range from 0 to 3. The range of the predictive score targets to categorize the 4 classes of noise variance. The possible predictive scores that correspond to respective noise variance are shown in Equation (11).

$$\begin{aligned}
 & \text{Predictive Output Score} \\
 & = \begin{cases} 0 & \text{Noise Variance} = 0.00 \\ 1 & \text{Noise Variance} = 0.01 \\ 2 & \text{Noise Variance} = 0.02 \\ 3 & \text{Noise Variance} = 0.03 \end{cases} \quad (11)
 \end{aligned}$$

D. Experimental Materials

Experimental materials are mandatory tools in evaluating the performance of the proposed GN-CNN approach. A training dataset is essential to train the neural network so that the proposed neural network can execute classification in the correct way. Besides, a validation dataset is applied for the tuning purpose in training the neural network. Some hyperparameters such as batch size, epoch number and learning rate are tuned by reference to the validation accuracy to obtain the highest accuracy. A testing dataset is employed to efficiently interpret the effectiveness of a neural network in predicting an outcome result. In this paper, there are a total of 1200 SEM images. The SEM images are obtained from EUDAT Collaborative Data Infrastructure [35]. Since the 4 main category classes of noise variance are 0, 0.01, 0.02, and 0.03 so each class of noise variance contains 300 images. The training dataset, validation dataset and testing dataset that are applied in this paper are divided into a percentage ratio of 60%, 20%, and 20% respectively. On this account, 720 images are used to train the neural network, whereas 240 validation images and 240 testing images are employed to evaluate the accuracy of classification. To assure the fairness and justice of result, the training datasets are not used repeatedly in validation and testing datasets. The repeated usage of images in the training and testing process would lead to high accuracy results. The architecture can easily recognize the images in testing phase since the identical image is trained before. Fig. 5 displays the numerical distribution of the dataset.

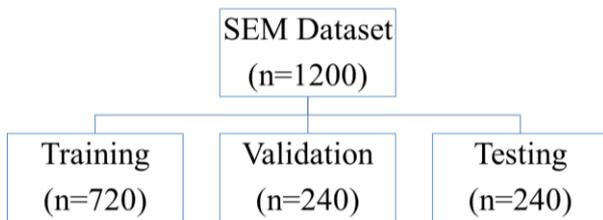


Fig. 5 Distribution of SEM dataset

IV. RESULTS & DISCUSSION

A. Results of Gaussian-Noise Convolutional Neural Network

1. Training and Validation

There are some training hyperparameters (batch size, epoch number and learning rate) that need to be declared in the GN-CNN. The neural network will execute training according to those predefined parameters. A total of 240 validation images are applied in the experimental study to adjust the hyperparameter values of the neural network. The number of epochs, dataset batch size, and learning rate had fine-tuned with different values to obtain the highest validation accuracy in the experiment.

The stopping criterion is defined to halt the training

session. Generally, the number of epochs is set to be at least 25 epochs by the reference to validation set [36]. The number of epochs is set at an optimum value of 38 as this value gives the highest validation accuracy. Fig. 6 illustrates the graph of validation accuracy at each training epoch. Besides, the batch size of the dataset in each training iteration is set at 25. It signifies that the system will randomly select 25 images for training purposes, and the process is duplicated 38 times. Moreover, the learning rate of neural network is termed at 0.0003. In other words, the highest validation accuracy is obtained with the number of epochs of 38, data batch size of 25, and learning rate of 0.0003.

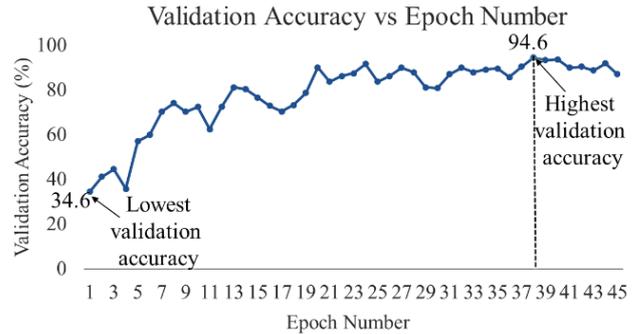


Fig. 6 Graph of validation accuracy at each training epoch

2. Testing Dataset

A total of 240 SEM images are employed as the testing dataset to verify the effectiveness of the proposed GN-CNN method. Fig. 7 illustrates the ability of the GN-CNN in predicting the noise variance of the sample SEM images.

Out of 240 images utilized in the testing phase, only several SEM images are opted to be displayed in Fig. 7. Obviously, the proposed method can effectively recognize the noise level of the grayscale SEM images and classify them into correct classes of noise. As illustrated in Fig. 7, the slight and insignificant difference in SEM images with different noise is hard to be differentiated by naked eyes. Besides, few assessment tools are adopted to further evaluate the proposed method. Assessment metrics such as recall, precision, F1 score, and accuracy are used to demonstrate the capability of the proposed method. The recall is termed as the ratio of correctly forecasted samples to the total number of samples in an expected class. On the other hand, precision is the ratio of correctly forecasted samples to the total number of samples in a predicted class (Hattori et al. 2018). F1 score creates a relationship between recall and precision as expressed in Equation (12). Accuracy is the ratio of correctly forecasted samples to the total number of samples in a class, as shown in Equation (13). Table 2 tabulates the computation of precision and recall in a multiclass classification system, whereas Table 3 shows the evaluation metrics of GN-CNN.

$$F1 \text{ Score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (12)$$

$$\begin{aligned}
 & \text{Testing Accuracy} \\
 & = \frac{\text{Number of correctly forecasted samples}}{\text{Total number of samples}} \quad (13)
 \end{aligned}$$

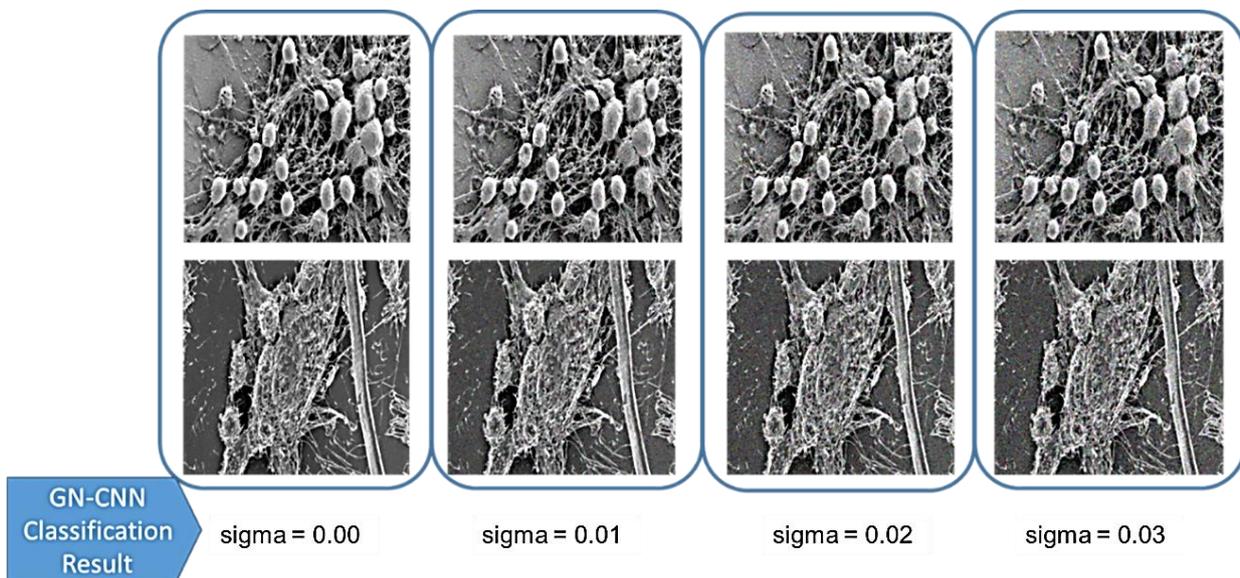


Fig. 7 GN-CNN architecture in predicting the noise variance of sample images

Table 2 Calculation of recall and precision of testing datasets in GN-CNN architecture

Predicted Class (Noise Variance)	Estimated Class (Noise Variance)				Precision
	0.00	0.01	0.02	0.03	
0.00	53	0	0	0	1.000
0.01	7	58	1	0	0.8788
0.02	0	2	58	4	0.9063
0.03	0	0	1	56	0.9825
Recall:	0.8833	0.9667	0.9667	0.9333	

Table 3 Evaluation metrics of GN-CNN architecture

Evaluation Metrics of GN-CNN	Score
Overall Recall	0.9375
Overall Precision	0.9419
F1 Score	0.9397
Testing Accuracy	93.80%

The model of data classification in this paper belongs to multiclass classification. To ease the computation purpose, we use a tabulation approach to calculate the overall recall and overall precision of GN-CNN. Overall recall is the average recall of estimated classes, whereas overall precision is the average precision of the predicted classes. As aforementioned, there are a total of 240 images used for testing purposes, so the estimated class contains 60 images. Table 2 demonstrates that the noise variance $NV = 0.01$ and $NV = 0.02$ classes have the highest recall value of 0.9667, $NV = 0.03$ (0.9333), and $NV = 0.00$ (0.8833). In the classes of noise variance with 0.01 and 0.02, only 2 samples are wrongly predicted. It can be concluded that the designed method has better recognition capability in classifying noise variance of SEM images with intermediate noise levels since the second ($NV = 0.01$) and third ($NV = 0.02$) classes have better recall and accuracy values. From Table 2, it is obvious that the system performs wrong prediction only in the next class or previous class of the noise variance. For instance, wrong prediction only occurs in the second class ($NV = 0.01$)

in the prediction for first class ($NV = 0.00$). In other words, the framework does not recognize the SEM images with $NV = 0.00$ as $NV = 0.02$ or $NV = 0.03$. This implies that the proposed system can efficiently differentiate the images with insignificant differences in noise level.

Table 2 exhibits that the class of $NV = 0.00$ has the highest precision of 1, followed by $NV = 0.03$ (0.9825), $NV = 0.02$ (0.9063), and $NV = 0.01$ (0.8788). F1 score and testing accuracy are computed by employing Equation (12) and Equation (13) respectively. It results in an F1 score of 0.9397 and testing accuracy of 93.8%, which are demonstrated in Table 3. There is only a slight difference of 0.8% between the testing accuracy (93.8%) and the validation accuracy (94.6%). The minor difference denotes that the proposed method is suitable for all the SEM datasets, including validation and testing datasets. Overall, the testing datasets show a high F1 score and testing accuracy of more than 90% reveal the capability and applicability of GN-CNN to classify the SEM images with different noise variance.

B. Comparison with Other Architectures

To prove the efficacy and potency of the proposed GN-CNN in executing noise variance classification task, 4 baseline models are designed to provide a comparison with our method. The 4 baseline algorithms are existing algorithms, including Long Short-Term Memory (LSTM), Residual Network 18 (ResNet18), and Residual Network 34 (ResNet34), and pure Convolutional Neural Network (CNN). Those models are the most basic architecture employed to perform classification.

The first baseline is Long Short-Term Memory (LSTM). It is one of the artificial Recurrent Neural Network (RNN) architecture which comprises input gate, output gate, and forget gate to regulate the information flow in a cell [34]. It is normally employed for generating classification and making a prediction. This is a conventional LSTM cell. It is not a bidirectional LSTM as used in GN-CNN. The number of inputs features and features in the hidden state are set as 700. We apply stacked LSTM by stacking three recurrent layers

together. The second baseline model is basic Residual Network 18 (ResNet18). The predefined parameters of ResNet18 which consists of CNN with 18 layers deep are used without modification. The third baseline model is Residual Network 34 (ResNet34), which is another image classification method. It comprises 34 CNN layers to further convolute the features. Even ResNet34 has been adopted in the convolution layer of GN-CNN, but the stand-alone accuracy of standard ResNet34 is still an unknown value.

Next, we employ 2 layers of Convolutional Neural Network (CNN) as the fourth baseline model. According to studies in Section 2.2, Khaw et al. [37], Momeny et al. [23], Murphy et al. [24], and Roy et al. [25] used 2 convolutional layers and 2 pooling layers in their studies. As a consequence, we use two layers CNN to represent the 4 studies as aforementioned. Thereby, the effectiveness of CNN can be determined when the comparison between GN-CNN and the CNN (4 studies) is made. The parameters used in the 2 layers CNN are described in Fig. 8.

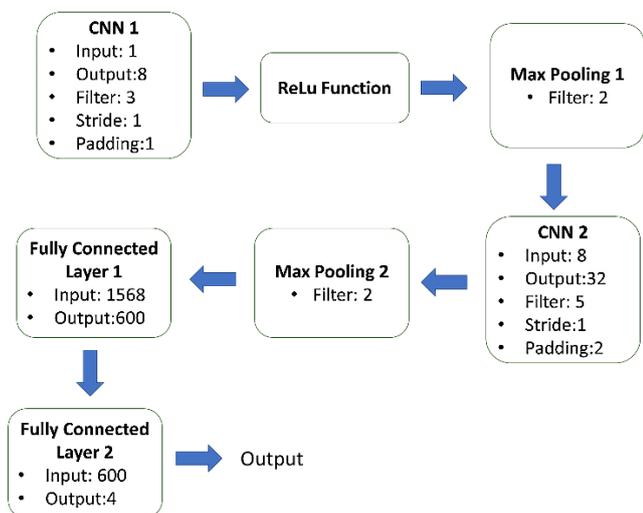


Fig. 8 Architecture of 2 layers CNN

To prevent bias of result, all the datasets (training, validation, and testing) used in all these 4 models are exactly the same as the datasets used in the GN-CNN model. To appraise the classification performance of the 4 standard models, evaluation metrics such as recall, precision, F1 score, and accuracy of each model are computed. The evaluation metrics are calculated so that a comparison can be made between the proposed GN-CNN and 4 standard metrics. Table 4 displays the comparison of 4 evaluation metrics between GN-CNN and 4 basic models.

Table 4 demonstrates that GN-CNN owns the highest recall value of 0.9375, followed by CNN (0.8625), ResNet34 (0.6208), ResNet18 (0.5583), and LSTM (0.2542). The

maximum recall value is 1, while GN-CNN has obtained a recall value (0.9375) of close to 1. Hence, it means the GN-CNN method has a high ratio of correctly forested samples to the total number of samples in an expected class. Besides, GN-CNN also performs with the highest precision value of 0.9419, followed by CNN (0.8784), ResNet34 (0.6560), ResNet18 (0.6192), and LSTM (0.3627). It indicates that GN-CNN has correctly predicted a large number of samples in a predicted class. F1 score established a relationship between recall and precision. Similarly, GN-CNN also owns the highest F1 score of 0.9397 with its highest recall and precision value, followed by CNN (0.8704), ResNet34 (0.6379), ResNet18 (0.5872), and LSTM (0.2989). High F1 score reveals that the model has higher training accuracy. The recall, precision, and F1 score have linear relationships since the evaluation metrics of all models are ranked in order. GN-CNN has obtained the highest values in all the evaluation metrics, followed by CNN, ResNet34, ResNet18, and LSTM. Besides, the proposed method of GN-CNN is more effective than the 4 existing baseline models, i.e., LSTM, ResNet18, ResNet34, CNN that are represented by CNN as all the evaluation metrics of GN-CNN are better than that of CNN. The F1 score of GN-CNN is 7.37% better than the F1 score of the 4 studies which are 2 layers CNN. Overall, our proposed method of GN-CNN model has the highest value of F1 score, recall, and precision.

Accuracy is another common metric to quantify the classification ability of the models. Table 4 shows that GN-CNN has obtained the highest accuracy of 93.8%, followed by CNN (86.3%), ResNet34 (62.1%), ResNet18 (55.8%), and LSTM (25.4%). Higher accuracy implies that the model has greater capability to successfully classify the SEM images into the correct noise variance class. In this regard, LSTM has the lowest classification capability because it only correctly classified a quarter of samples. One most possible contributor for this scenario is that the LSTM is normally well-suited to perform classification or prediction on data which is dependent on time. The prediction of time series data tasks cannot be solved by other feed-forward neural networks as they process features using time windows with a fixed size [38]. SEM images with different noise variance are independent of time because they are not time-series data. LSTM provides better applicability in classifying time series data or sequential data, such as electroencephalogram (EEG) classification, electromyography (EMG) classification, and so on. This is due to the reason of EEG and EMG are biological signals that are measured with respect to time.

ResNet18, ResNet34, CNN, and GN-CNN have obtained accuracy of more than 50%. These models have correctly classified more than half of the images with different noise variances. ResNet18, ResNet34, CNN, and GN-CNN are

Table 4 Comparison of evaluation metrics between GN-CNN and 4 existing baseline models

Model	Evaluation Metrics			
	Recall	Precision	F1 Score	Accuracy
LSTM	0.2542	0.3627	0.2989	25.4%
ResNet18	0.5583	0.6192	0.5872	55.8%
ResNet34	0.6208	0.6560	0.6379	62.1%
CNN	0.8625	0.8784	0.8704	86.3%
GN-CNN	0.9375	0.9419	0.9397	93.8%

under the category of convolutional neural network, whereas LSTM is under the category of recurrent neural network. Therefore, the convolutional neural network is more feasibly used to perform noise classification, as compared to the recurrent neural network. Most of the recurrent neural networks are well-suited for training data that are related to time.

The accuracy results of ResNet18 and ResNet34 are quite close to each other with only 6% difference between them. ResNet34 has a higher accuracy because it has deeper layers of architecture as compared to ResNet18. The higher number of convolutional layers of ResNet34 has enhanced the accuracy of the result. Nevertheless, ResNet18 and ResNet34 are still not applicable for performing noise classification for the reason of low accuracy percentage of not exceeding 80%.

Besides, 2-layer CNN gives the highest accuracy of 86% among all the standard baseline models. This is the best accuracy that we obtain after experimenting with the CNN algorithm with different parameters such as filter size, padding size, stride size, and fully connected layer. This has proven that the CNN (4 existing studies in Section 2.2) offers better efficiency in classifying noised images as compared to residual networks. Nevertheless, the proposed GN-CNN provides a better performance than the classical CNN in classifying a noisy image. The accuracy of GN-CNN is 7.5% higher than the accuracy of the 2 layers classical CNN.

Overall, the GN-CNN approach outperforms all the 4 baseline models. In this way, GN-CNN has a better performance owing to the reason that it consists of the most complicated and complex neural network architecture. Architecture of encoder layer (bi-directional LSTM), convolutional layer (ResNet34), attention layer, decoder layer (LSTM cell), and decision layer (fully connected layer) make it powerful in training raw images and giving classification outcomes. Therefore, GN-CNN undoubtedly is the most efficient and effective tactic in classifying the SEM images with 4 different noise variances. The outcome of assessment has ascertained that the proposed method serves as a better classifier of noise image.

V. CONCLUSION

To summarize, the determination of noise variance of a Scanning Electron Microscopy (SEM) image is crucial to ease the noise reduction process. Before the noise level classification process, the SEM images are added with White Gaussian noise in 4 different noise variance classes of 0, 0.01, 0.02, and 0.03. The images with added noise are then converted to histogram images. Subsequently, the convolutional neural network has been modified to a more advanced and complex architecture, called Gaussian-Noise Convolutional Neural Network (GN-CNN) in this paper. GN-CNN is developed with an intention to effectively classify the SEM images with added White Gaussian noise into 4 different noise variance categories. GN-CNN architecture consists of encoder layer (bi-directional LSTM), convolutional layer (ResNet34), attention layer, decoder layer (LSTM cell), and decision layer (fully connected layer) in sequence. This paper has employed a total of 1200 images for experimental study, with 60% for training, 20% for validation and testing respectively. 4 basic baseline models, including LSTM, ResNet18, ResNet34, and CNN are implemented to make a comparison with our proposed

method. The CNN is used to represent 4 existing studies (Khaw, Momeny, Murphy, and Roy). As a result, the proposed GN-CNN has outperformed the other basic baseline models in terms of recall, precision, F1 score, and accuracy. GN-CNN has acquired recall of 0.9375, precision of 0.9419, F1 score of 0.9397, and accuracy of 93.8%. The results have demonstrated that the GN-CNN outperforms the 4 existing studies that use 2 layers CNN as their architecture. Therefore, the experimental study has shown that the GN-CNN can effectively classify the SEM images with different noise variance through the deep learning model. Nevertheless, the accuracy still can be enhanced in a few ways. More sets of data should be implemented in the experimental study so that more features will be trained and the neural network system is able to recognize more noisy images with different features. Besides, it is expected that higher accuracy can be achieved if a classical CNN or a ResNet34 is used in the convolutional layer.

Overall, the experimental study has evidenced the capability and reliability of the developed system in classifying noise variances of SEM image. In future, the number of classification classes can be further increased to 8 or 10, instead of 4 classes. The positive results also reflect those images with higher noise variance are also applicable for use in the future experimental study. Thus, it brings numerous benefits since the designed network has impressively surpassed human-eye performance in noise variance categorization.

ACKNOWLEDGEMENT

We would like to thank Rossella Aversa et al. for providing the dataset of SEM images in EUDAT for research purposes [35].

REFERENCES

- [1] H. Y. Khaw, F. C. Soon, J. H. Chuah, and C. O. Chow, "Image noise types recognition using convolutional neural network with principal components analysis," *IET Image Process.*, vol. 11, no. 12, pp. 1238-1245, 2017, doi: 10.1049/iet-ipr.2017.0374.
- [2] K. S. Sim, F. F. Ting, J. W. Leong, and C. P. Tso, "Signal-to-noise Ratio Estimation for SEM Single Image using Cubic Spline Interpolation with Linear Least Square Regression," *Engineering Letters*, vol. 27, no.1, pp151-165, 2019.
- [3] K. S. Sim, M. A. Lai, C. P. Tso, and C. C. Teo, "Single image signal-to-noise ratio estimation for magnetic resonance images," *Journal of Medical Systems.*, vol. 35, no. 1, pp. 39-48, 2011, doi: 10.1007/s10916-009-9339-9.
- [4] M. Rakhshanfar and M. A. Amer, "Estimation of Gaussian, Poissonian-Gaussian, and Processed Visual Noise and Its Level Function," *IEEE Trans. image Process. a Publ. IEEE Signal Process. Soc.*, vol. 25, no. 9, pp. 4172-4185, 2016, doi: 10.1109/TIP.2016.2588320.
- [5] B. Xiong and Z. Yin, "A Universal Denoising Framework With a New Impulse Detector and Nonlocal Means," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1663-1675, 2012, doi: 10.1109/TIP.2011.2172804.
- [6] M. Kivanc Mihcak, I. Kozintsev, K. Ramchandran, and P. Moulin, "Low-complexity image denoising based on statistical modeling of wavelet coefficients," *IEEE Signal Process. Lett.*, vol. 6, no. 12, pp. 300-303, 1999, doi: 10.1109/97.803428.
- [7] L. Zhang, W. Dong, D. Zhang, and G. Shi, "Two-stage image denoising by principal component analysis with local pixel grouping," *Pattern Recognit.*, vol. 43, no. 4, pp. 1531-1549, 2010, doi: <https://doi.org/10.1016/j.patcog.2009.09.023>.
- [8] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering," *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080-2095, 2007, doi: 10.1109/TIP.2007.901238.
- [9] M. Elad and M. Aharon, "Image Denoising Via Sparse and Redundant Representations Over Learned Dictionaries," *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736-3745, 2006, doi:

- 10.1109/TIP.2006.881969.
- [10] K. S. Sim, M. Y. Wee, and W. K. Lim, "Image signal-to-noise ratio estimation using Shape-Preserving Piecewise Cubic Hermite Autoregressive Moving Average model," *Microsc. Res. Tech.*, vol. 71, no. 10, pp. 710–720, 2008, doi: 10.1002/jemt.20610.
- [11] K. S. Sim, C. K. Toa, and C. W. Ho, "Cubic Spline Hermite Interpolation with Linear Least Square Regression for Single Scanning Electron Microscope Image Signal-to-Noise Ratio Estimation," *IAENG International Journal of Computer Science*, vol. 47, no. 4, pp. 740–752, 2020.
- [12] K. S. Sim and V. Teh, "Image signal-to-noise ratio estimation using adaptive slope nearest-neighbourhood model," *J. Microsc.*, vol. 260, no. 3, pp. 352–362, 2015, doi: 10.1111/jmi.12302.
- [13] Y. Roy, H. Banville, I. Albuquerque, A. Gramfort, T. H. Falk, and J. Faubert, "Deep learning-based electroencephalography analysis: a systematic review," *J. Neural Eng.*, vol. 16, no. 5, pp. 37, 2019, doi: 10.1088/1741-2552/ab260c.
- [14] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, "Face recognition: a convolutional neural-network approach," *IEEE Trans. Neural Networks*, vol. 8, no. 1, pp. 98–113, 1997, doi: 10.1109/72.554195.
- [15] Y. LeCun *et al.*, "Handwritten Digit Recognition with a Back-Propagation Network," *Advances in Neural Information Processing Systems*, pp. 396–404, 1989.
- [16] C. Yan, B. Zhang, and F. Coenen, "Driving posture recognition by convolutional neural networks," in *2015 11th International Conference on Natural Computation (ICNC)*, pp. 680–685, 2015, doi: 10.1109/ICNC.2015.7378072.
- [17] J. Fang, Y. Zhou, Y. Yu, and S. Du, "Fine-Grained Vehicle Model Recognition Using A Coarse-to-Fine Convolutional Neural Network Architecture," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 7, pp. 1782–1792, 2017, doi: 10.1109/TITS.2016.2620495.
- [18] B. Hari and V. George, "Noise," in *6.02 Introduction to EECS II: Digital Communication Systems*, Massachusetts Institute of Technology: MIT OpenCourseWare, pp. 115, 2012.
- [19] J. T. L. Thong, K. S. Sim, and J. C. H. Phang, "Single-image signal-to-noise ratio estimation," *Scanning*, vol. 23, no. 5, pp. 328–336, Sep. 2001, doi: <https://doi.org/10.1002/sca.4950230506>.
- [20] K. S. Sim and N. S. Kamel, "Image signal-to-noise ratio estimation using the autoregressive model," *Scanning*, vol. 26, no. 3, pp. 135–139, 2004, doi: 10.1002/sca.4950260306.
- [21] L. Gao, Q. Du, B. Zhang, W. Yang, and Y. Wu, "A Comparative Study on Linear Regression-Based Noise Estimation for Hyperspectral Imagery," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 6, no. 2, pp. 488–498, 2013, doi: 10.1109/JSTARS.2012.2227245.
- [22] J. H. Chuah, H. Y. Khaw, F. C. Soon, and C. O. Chow, "Detection of Gaussian noise and its level using deep convolutional neural network," *TENCON 2017 - 2017 IEEE Region 10 Conference*, pp. 2447–2450, 2017, doi: 10.1109/TENCON.2017.8228272.
- [23] M. Momeny, A. M. Latif, M. Agha Sarram, R. Sheikhpour, and Y. D. Zhang, "A noise robust convolutional neural network for image classification," *Results Eng.*, vol. 10, pp. 100225, 2021, doi: <https://doi.org/10.1016/j.rineng.2021.100225>.
- [24] J. Murphy, Y. Pao, and A. Haque, "Image-Based Classification of GPS Noise Level Using Convolutional Neural Networks for Accurate Distance Estimation," in *Proceedings of the 1st Workshop on Artificial Intelligence and Deep Learning for Geographic Knowledge Discovery*, pp. 10–13, 2017, doi: 10.1145/3149808.3149811.
- [25] S. Roy, S. Hossain, M. A. H. Akhand, and K. Murase, "A Robust System for Noisy Image Classification Combining Denoising Autoencoder and Convolutional Neural Network," *Int. J. Adv. Comput. Sci. Appl.*, vol. 9, 2018, doi: 10.14569/IJACSA.2018.090131.
- [26] D. Sil, A. Dutta, and A. Chandra, "Convolutional Neural Networks for Noise Classification and Denoising of Images," in *TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON)*, 2019, pp. 447–451, doi: 10.1109/TENCON.2019.8929277.
- [27] K. S. Sim, M. E. Nia, C. P. Tso, and W. K. Lim, "Performance of new signal-to-noise ratio estimation for SEM images based on single image noise cross-correlation," *J. Microsc.*, vol. 248, no. 2, pp. 120–128, 2012, doi: 10.1111/j.1365-2818.2012.03657.x.
- [28] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.
- [29] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, 1997, doi: 10.1109/78.650093.
- [30] C. K. Toa, K. S. Sim, and S. C. Tan, "Electroencephalogram-Based Attention Level Classification Using Convolution Attention Memory Neural Network," *IEEE Access*, vol. 9, pp. 58870–58881, 2021, doi: 10.1109/ACCESS.2021.3072731.
- [31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016, doi: 10.1109/CVPR.2016.90.
- [32] K. Xu *et al.*, "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention," in *32nd International Conference on Machine Learning, ICML 2015*, vol. 3, pp. 2048–2057, 2015.
- [33] Y. Chu, X. Yue, L. Yu, M. Sergei, and Z. Wang, "Automatic Image Captioning Based on ResNet50 and LSTM with Soft Attention," *Wirel. Commun. Mob. Comput.*, pp. 8909458, 2020, doi: 10.1155/2020/8909458.
- [34] N. C. Tay, C. Tee, T. S. Ong, and P. S. Teh, "Abnormal Behavior Recognition using CNN-LSTM with Attention Mechanism," in *2019 1st International Conference on Electrical, Control and Instrumentation Engineering (ICECIE)*, pp. 1–5, 2019, doi: 10.1109/ICECIE47765.2019.8974824.
- [35] C. R. Aversa R, Modarres MH, Cozzini S, "100% SEM Dataset. EUDAT Collaborative Data Infrastructure," *EUDAT Collaborative Data Infrastructure*, 2018. <http://doi.org/10.23728/b2share.80df8606fcd4b2bae1656f0dc6db8ba>.
- [36] F. Ting, Y. Tan, and K. Sim, "Convolutional Neural Network Improvement for Breast Cancer Classification," *Expert Syst. Appl.*, vol. 120, Nov. 2018, doi: 10.1016/j.eswa.2018.11.008.
- [37] H. Y. Khaw, F. C. Soon, J. H. Chuah, and C. O. Chow, "Image Noise Types Recognition Using Convolutional Neural Network with Principal Components Analysis," *IET Image Process.*, vol. 11, 2017, doi: 10.1049/iet-ipr.2017.0374.
- [38] F. A. Gers, D. Eck, and J. Schmidhuber, "Applying LSTM to Time Series Predictable Through Time-Window Approaches BT - Neural Nets WIRN Vietri-01," pp. 193–200, 2002.