Image and Texture Independent Deep Learning Noise Estimation Using Multiple Frames

Hikmet Kirmizitas*, Nurettin Besli
Department of Electrical and Electronics Engineering, Harran University, Sanliurfa, Turkey
hkirmizitas@harran.edu.tr

***Abstract*—In this study, a novel multiple frame based image and texture independent Convolutional Neural Network (CNN) noise estimator is introduced. Noise estimation is a crucial step for denoising algorithms, especially for ones that are called "non-blind". The estimator works for additive Gaussian noise for varying noise levels. The noise levels studied in this work have a standard deviation equal to 5 to 25 increasing 5 by 5. Since there is no database for noisy multiple images to train and validate the network, two frames of synthetic noisy images with a variety of noise levels are created by adding Additive White Gaussian Noise (AWGN) to each clean image. The proposed method is applied on the most popular gray level images besides the color image databases such as Kodak, McMaster, BSDS500 in order to compare the results with the other works. Image databases comprise indoor and outdoor scenes that have fine details and richer texture. The estimator has an accuracy rate of 99 % for the classification and favourable results for the regression. The proposed method outperforms traditional methods in most cases. And the regression output can be used with any non-blind denoising method.**

***Index Terms*—Deep learning; Multiple frames; Noise estimation.**

I. INTRODUCTION

Image noise estimation is a critical step for denoising methods. Denoising methods try to reduce the noise in noisy and corrupted images to get a clearer result. In doing so, most methods need to know the noise level inherent in the noisy image. The success of these algorithms relies on estimating the true added noise level. Hence, research must be done on estimating noise level. Today, most of the denoising algorithms are non-blind, meaning that they take noise level as a parameter before denoising operation. The non-blind methods include Wiener filter [1], Non-Local Means [2] and Block-Matching, and 3D Filtering (BM3D) [3]. Convolutional Neural Network (CNN) is used as the state-of-the-art image denoising algorithms [4], [5]. There are studies on blind and non-blind versions. Wrong noise level parameter degrades the quality of the denoised image. Thus, it must be handled with care. In the literature, several noise estimation methods [6]–[8] have been studied. In addition, solutions based on Principal Component Analysis (PCA) [9], in which the smallest eigenvalue of the covariance matrix is chosen, are studied. There are mainly 3 types of noise estimation methods: the first is filter based [10], the second is patch based [11]–[13], and the last is transform based [14]–[17] noise estimation method. Filter based methods work for single image and pass the original noisy input image from a high pass filter, and the output of the filter is used to compare with the original noisy image. The difference of two images is taken to decide the noise level. But these approaches suffer from suppressing the image details and smoothing the rich texture and fine details of the clean image. In patch based approaches, images are thought to consist of patches that are of size N×N. In patch based methods, the standard deviations of patches are investigated and the one having the least standard deviation among all the patches is taken. The disadvantage of these methods is to overestimate the noise level for images that have small noise levels and underestimate when the image has high noise levels. Therefore, in these types of methods, the success of the estimation depends highly on the inherent noise level. In statistical approaches, the change in the kurtosis values is affected by the noise types and level in the image. Noise estimation results when used with denoising methods can increase performance [11].

In recent years, denoising with multiple images and CNN based methods has started to be used. CNNs are among the state-of-the-art methods for denoising. In CNNs, a deep learning architecture is trained with noisy images and the corresponding clean noise free images as inputs and outputs for the deep network, respectively. In [18], the noise is detected in the image and the noise level is classified into 10 bins. In this study, if the image does not have noise, it is classified as noise free. Apart from the noise free class, there are 9 more classes that have a standard deviation equal to 10 to 90 increasing 10 by 10. The CNN architecture is MatConvNet which has 4 convolutional layers, 2 max-pooling layers, 1 Rectified Linear Unit (ReLU) layer, and finally a softmax layer for classification purposes. Noisy images are classified by implicitly putting each to a class. Since noise level is not calculated explicitly, it cannot be used with the other non-blind denoising methods. In [19], the noise is estimated in a pixelwise manner. Since real world noisy image is different from synthetically corrupted noisy images, instead of a global scalar noise level, the noise level of every pixel is estimated using deep learning. It is a successful work that surpasses most of the state-of-the-art methods (Liu, Tanaka, and Okutomi [20], Pyatykh,

Manuscript received 24 January, 2022; accepted 17 October, 2022.

Hesser, and Zheng [9], and Chen, Zhu, and Heng [21]). They use a stack of residual patches. In these residuals patches, there is no pooling or interpolation operation. But still the noise estimation results can be improved. In [22], the noise level is proposed to be estimated using Singular Value Decomposition (SVD) and a neural network. The tail parts of the singular values of an image grow with increasing noise level, constituting a measurement for noise level. Hence, these singular values are used as inputs to the neural network, and the standard deviation of noise is assigned as output of the network. Thus, the estimation of noise level is made possible with their model.

The aim of this work is to accurately find out the noise level for denoising applications, especially for non-blind applications requiring accurate noise level as an input parameter to obtain effective solutions.

II. SQUEEZENET

SqueezeNet is a small Deep Neural Network (DNN) architecture with fewer parameters compared to AlexNet. Due to small model size and number of the parameters, SqueezeNet requires less communication between servers during distributed training, less bandwidth for transferring the network model, and is more suitable for hardware implementation. SqueezeNet has one fiftieth fewer parameters than AlexNet with sufficient accuracy. Furthermore, SqueezeNet requires less than 0.5 MB storage space when compressed [23].

SqueezeNet, different from other deep learning architectures, is a Fully Convolutional Network (FCN) that only performs convolution operations in convolution layers and fire modules. A typical fire module consists of the following layers: a squeeze convolution layer that has only 1×1 filters and an expand layer that has a combination of 1×1 and 3×3 filters as shown in Fig. 1. In squeeze layer, one ninth of less storage occupation is achieved because 1×1 filters use one ninth space compared to 3×3 filters. In addition, three tunable dimensions are described for these layers: s1×1, e1×1, and e3×3. S1×1 is the count of filters in the squeeze layer all of which are 1×1, e1×1 is the count of 1×1 filters, and e3×3 is the count of 3×3 filters in the expand layer. Moreover, the dimension decrease occurs by setting s1×1 to be less than (e1×1 + e3×3). Thus, the count of input channels is decreased to 3×3 filters.

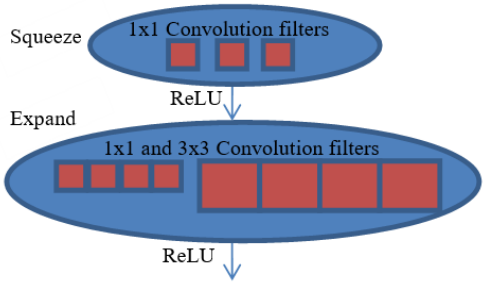

Fig. 1. Detailed description of the SqueezeNet fire modules.

The Fully Convolutional Network (FCN) works well in matching images with or without noise [24]. However, when using FCN, the number of convolutional layers must be chosen according to its generalization capability. If the number of layers is chosen too small or too large, the network can converge to an undesirable point. Too few layers as low as 10 might not be enough to be successful. Most denoising CNNs with too many convolutional layers cause the result to lose fine details. Therefore, we used the SqueezeNet as base CNN model in order to estimate the noise level of an image. The architecture of standard SqueezeNet is shown in Fig. 2(a).

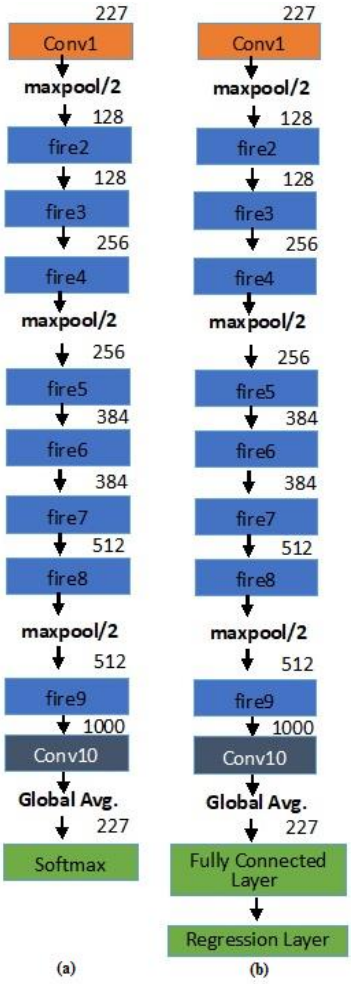

Fig. 2. (a) SqueezeNet architecture; (b) Modified SqueezeNet architecture.

III. PROPOSED METHOD

In this study, the SqueezeNet deep learning architecture is utilized as a base CNN network. Since fast shooting and high-resolution images can be obtained with current technology, it seems possible to acquire multiple frames of an image with similar noise. In the proposed method, the difference image, which is obtained by subtracting two noisy frames of an image, is fed to the Neural Network. Therefore, the input of the network is the difference of two noisy images, and the output of the network is the estimated noise level of the corresponding image. In this study, the classification of the noise level of the image will be carried out with the standard SqueezeNet model as seen in Fig. 2(a). Since noise level is not calculated explicitly in the classification, the regression of the noise level of the image will be carried out with a modified SqueezeNet model in which the last two layers are replaced with a Fully Connected Layer (FCL) and a regression layer as in Fig. 2(b).

As mentioned earlier, there is no database for noisy multiple images to train and validate the network. Two frames of synthetic noisy images with a variety of noise levels are created by adding Additive White Gaussian Noise (AWGN) to each clean image as given in Fig. 3.

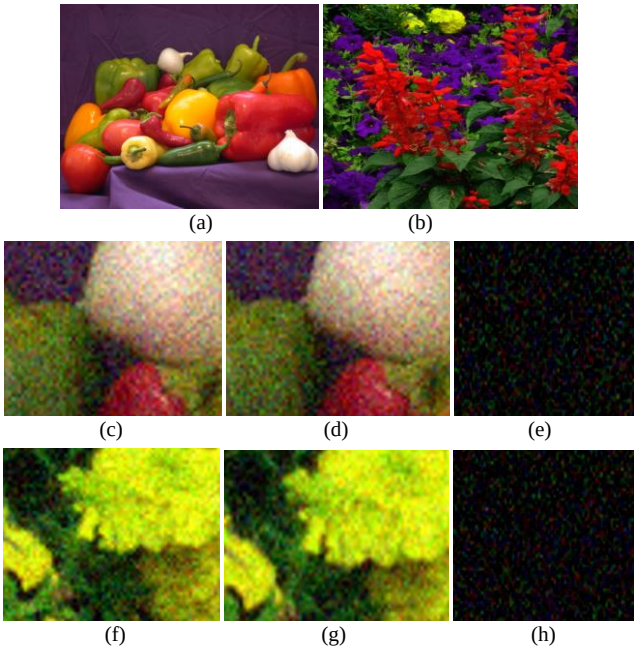

Fig. 3. (a) Original peppers image and (b) image from the Kodak Dataset, (c), (d), (f), and (g) are noisy and enlarged images obtained from (a) and (b), and (e) and (h) are the difference as network inputs.

These two noisy images are created under the assumption that the original image is noise free, and the noise added is purely AWGN. If the noise added to both frames is purely additive, the proposed method gives image and texture independent noise estimation results. The input of our network, which is the difference of two noisy images, is mostly independent from the original picture and characterizes the inherent noise in the image. In most of the methods reviewed, if the image has fine details, the success of the noise estimation decreases as mentioned in [10]. Suppose that the original clean image and AWGN image of size N×M are denoted by Im and μ, respectively. The first noisy image can be denoted by Im + μ1 and the second image by Im + μ2, then the difference is (Im + μ1) – (Im + μ2) yielding (μ1 – μ2). The result is the difference of two AWGNs due to cancelation of Im. Thus, the input to the network becomes irrelevant of the clean image and texture. For gray level images, the input layer has one channel, and for color images, the layer has 3 channels (RGB). The proposed method consists of CNN training and testing steps with the created image database. The depiction of the proposed model with input image is presented in Fig. 4.

The noisy images are created synthetically by adding AWGN with MATLAB's "imnoise" function. In application, creating images synthetically by adding AWGN to noise free images twice means that two noisy pictures of a reference scene are taken consecutively having zero delay in time.

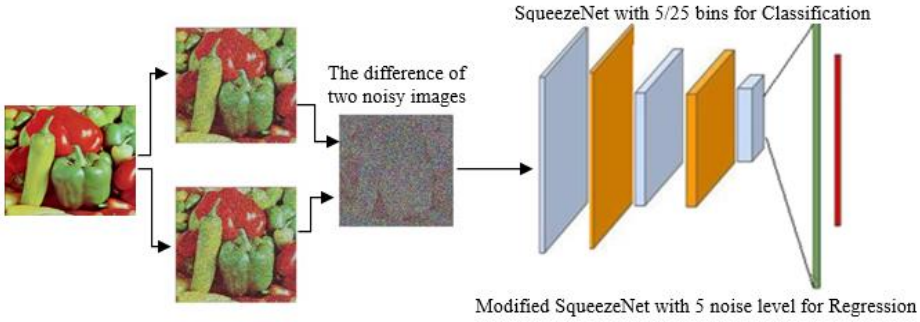

Fig. 4. The proposed method.

IV. EXPERIMENTS

In this study, three experiments were conducted. First is classification of 1 to 25 noise standard deviation levels with 25 bins increasing 1 by 1 with 4,500 samples for each class, second is classification of 5 to 25 noise level with 5 bins increasing 5 by 5 with 2,000 samples for each class, and the last is regression of 5 to 25 noise level increasing 5 by 5 with 2,000 samples for each regressed value. We increased the number of samples for the first classification scheme to obtain a better result.

The classification of the noise level of image was carried out with the standard SqueezeNet model in MATLAB. To implement network training for both classification and regression, the model parameters must be defined. The system solver for the network was selected as the Adam optimization algorithm "adam", and the initial learning rate was chosen to be 0.01. The validation frequency of the training was 50. The learning rate schedule was set as "none" and the learning rate drop factor was 0.1. The learning rate drop period was 10, and the momentum was chosen to be 10. L2Regularization parameter of training was 0.0001 and the gradient threshold method was L2Norm. The gradient threshold and the validation patience were set to infinity. The number of epochs was chosen to be 30, but in 3 epochs, the network was successful in reaching an accuracy rate of 100 %. We limit the number of epochs to 30 to avoid memorizing the noise and the corresponding output. The minimum batch size was 128. The execution environment was Graphics Processing Unit (GPU) for both classification and regression. Since there were 4,500 samples per class in a total of 112,500 samples with 25 bins and 2,000 samples per class in a total of 10,000 samples with 5 bins. Since these operations require too much processing time for an ordinary Central Processing Unit (CPU), the GPU was selected as the training environment. The data were shuffled at every epoch, yielding a more robust learning. As the epoch number increases, the accuracy of the trained network

increases. For the classification, the difference image is classified with 25 noise levels, i.e., it has 25 different levels starting from 1 to 25 increasing 1 by 1. The accuracy rate was 99 %. When the number of classes was reduced to 5 with noise level from 5 to 25 increasing 5 by 5, the classification accuracy rate is 100 %, as in the system, classes increasing one by one until 25. The classification accuracy shows that the system could learn the difference image accurately.

For the implementation of regression, the last layer of the SqueezeNet is replaced with FCL and regression layers. The training was implemented as regression rather than classification. With 2000 samples for each regression value (i.e., σ = 10, 20, 30, 40, 50), a total of 10,000 training samples are used. After the network is trained with 2000 samples (80 % of Dataset) for each noise standard deviation level, we proceed to the testing step with 500 samples for each for validation.

In Fig. 5, the noise level predictions of the trained network were plotted. For smaller regression values, the error is too small both in value and percentage. When the regression value increases, both the regression value and percentage of the error increase. Besides, in classification, the samples are forced to be one of the output classes. In regression, there is also a regression force which makes the output to be one of the regression levels, but this is not as forceful as in the classification process.

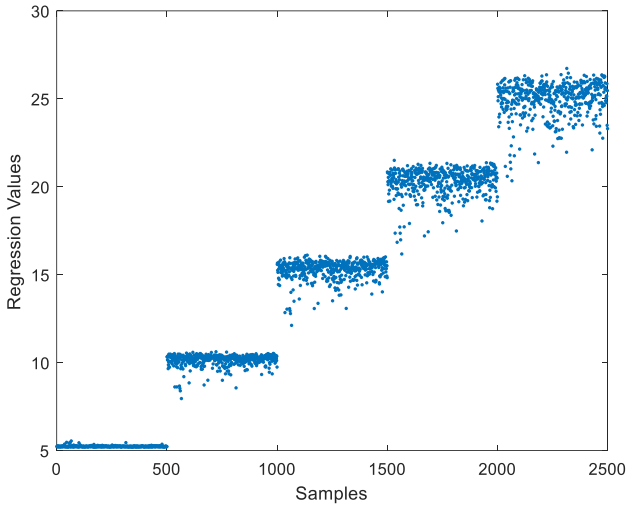

Fig. 5. The regression plot for validation data.

As a performance evaluation, we only used quantitative results. The comparison metric is the Root Mean Squared Error (RMSE) for the estimated noise level as chosen in the compared studies.

The test datasets were Kodak, McMaster, and BSDS500. When comparing the results, other parameters were kept the same. For color images, the three channels are treated independently, and as for the final result, the average of three channels is calculated.

V. RESULTS

In this study, for classification of noise with 5 bins, and regression with 5 noise levels, we used 2000 images for training and 500 images for the network validation per class. In classification of noise with 5 bins (noise level 5-10-15-20-25), the proposed method has an accuracy rate of %100. For classification with 25 bins, we used 4,500 images for training and 1,125 images for validation for each class. The accuracy rate for classification with 25 bins was 99 %. The regression validation results are shown in Fig. 5.

The results of the proposed method are compared with state-of-the-art methods presented by Pyatykh, Hesser, and Zheng [9], Tan, Xiao, Lai, Liu, and Zhang [19], Liu, Tanaka, and Okutomi [20], and Chen, Zhu, and Heng [21] in noise estimation. The source code for these implementations can be downloaded from the given URL site on the Internet. The results of all methods can be found in Table I.

The regression network is trained with 5 levels of AWGN with σ = 5, 10, 15, 20, 25. Other noise levels for regression are not considered since the methods with which we compared our results tested only images with these noise levels. Although McMaster database has heavy fine detailed pictures, the success of the algorithm does not change. The proposed method wins the first place 9 times out of 15.

As shown in Fig. 4, regardless of the input image and its texture, the network input is just the noise. After taking the difference of two noisy images, the input becomes purely the difference of two randomly distributed noise signals. Therefore, after training the network, the result can be found very fast compared to the other methods as seen in Table II.

The result values can be used by other non-blind denoising methods since it successfully estimates the noise level in regression.

TABLE I. ROOT MEAN SQUARED ERROR OF ESTIMATED NOISE LEVEL ON DIFFERENT DATASETS.

Dataset	**Noise Level**	**Pyatkh, Hesser, and Zheng [9]**	**Tan, Xiao, Lai, Liu, and Zhang [19]**	**Liu, Tanaka, and Okutomi [20]**	**Chen, Zhu, and Heng [21]**	**Proposed Method**
Kodak	σ = 5	0.51	**0.15**	0.26	**0.15**	*0.23*

Dataset	Noise Level	Pyatkh, Hesser, and Zheng [9]	Tan, Xiao, Lai, Liu, and Zhang [19]	Liu, Tanaka, and Okutomi [20]	Chen, Zhu, and Heng [21]	Proposed Method
(24 Images)	σ = 10	1.00	0.30	0.49	0.36	**0.27**
	σ = 15	1.46	0.50	0.67	0.55	**0.48**
	σ = 20	1.91	0.71	0.86	0.75	**0.55**
	σ = 25	2.31	0.96	1.07	0.95	**0.66**
McMaster (18 Images)	σ = 5	0.17	**0.11**	0.18	0.28	0.22
	σ = 10	0.52	0.32	0.46	0.34	**0.22**
	σ = 15	0.87	0.62	0.76	0.67	**0.40**
	σ = 20	1.11	0.94	1.09	1.03	**0.51**
	σ = 25	1.35	1.28	1.42	1.37	**0.71**
BSD500 (500 Images)	σ = 5	1.12	0.11	1.31	**0.08**	*0.23*
	σ = 10	0.80	0.26	0.42	**0.22**	0.35
	σ = 15	1.25	0.47	0.61	**0.41**	0.61
	σ = 20	1.71	0.72	0.85	**0.64**	0.80
	σ = 25	2.17	1.00	1.11	0.91	**0.90**

TABLE II. NOISE ESTIMATION EXECUTION TIMES IN SECONDS.

Database	Pyatykh, Hesser, and Zheng [9] (CPU)	Tan, Xiao, Lai, Liu, and Zhang [19] (CPU/GPU)	Liu, Tanaka, and Okutomi [20] (CPU)	Chen, Zhu, and Heng [21] (CPU)	Proposed Method
McMaster	1.75	5.31/1.16	2.20	0.27	0.01

VI. CONCLUSIONS

In this paper, we propose a Gaussian noise estimation method that outperforms the state-of-the-art noise estimation methods. The output of this architecture can be used by non-blind denoising methods. The input of the network is purely the difference of the noise added to the original noise free images. Regardless of the original image, the system can learn and perform classification and regression. For regression model, the regression values for validation data are plotted in Fig. 5 and the comparison of the results is given in Table I. The results support that the CNN can learn the difference noise well. This is a major advantage, since all of the noise estimation methods work inefficiently when the original image has details and fine texture. The fine texture complicates the method of estimating the noise level. As can be seen from the results, the proposed method is independent of image and texture, i.e., it works with the similar accuracy for all of the images as long as the noisy images have AWGN. Compared to other state-of-the-art noise estimation methods, our proposed method works better most of the time.

CONFLICTS OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

[1] N. Wiener, *Extrapolation, Interpolation, and Smoothing of Stationary Time Series: With Engineering Applications*, 1st ed. Cambridge, Technology Press of Massachusetts Institute of Technology, 1949. DOI: 10.7551/mitpress/2946.001.0001.

[2] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising", in *Proc. of 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 2005, pp. 60–65, vol. 2. DOI: 10.1109/CVPR.2005.38.

[3] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising with block-matching and 3D filtering", *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 6064, pp. 606414-1–606414-12, 2006. DOI: 10.1117/12.643267.

[4] H. C. Burger, C. J. Schuler, and S. Harmeling, "Image denoising: Can plain neural networks compete with BM3D?", in *Proc. of 2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2392–2399. DOI: 10.1109/CVPR.2012.6247952.

[5] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising", *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017. DOI: 10.1109/TIP.2017.2662206.

[6] P. Jiang and J.-z. Zhang, "Fast and reliable noise level estimation based on local statistic", *Pattern Recognition Letters*, vol. 78, pp. 8–13, 2016. DOI: 10.1016/j.patrec.2016.03.026.

[7] C. Liu, W. T. Freeman, R. Szeliski, and S. B. Kang, "Noise estimation from a single image", in *Proc. of* 2006 *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, 2006, pp. 901–908. DOI: 10.1109/CVPR.2006.207.

[8] H. Yue, J. Liu, J. Yang, T. Nguyen, and C. Hou, "Image noise estimation and removal considering the Bayer pattern of noise variance", in *Proc. of 2017 IEEE International Conference on Image Processing*, 2017, pp. 2976–2980. DOI: 10.1109/ICIP.2017.8296828.

[9] S. Pyatykh, J. Hesser, and L. Zheng, "Image noise level estimation by principal component analysis", *IEEE Transactions on Image Processing*, vol. 22, no. 2, pp. 687–699, Feb. 2013. DOI: 10.1109/TIP.2012.2221728.

[10] S. K. Abramov, V. V. Lukin, B. Vozel, K. Chehdi, and J. T. Astola, "Segmentation-based method for blind evaluation of noise variance in images", *Journal of Applied Remote Sensing*, vol. 2, no. 1, pp. 1–16, Aug. 2008. DOI: 10.1117/1.2977788.

[11] D.-H. Shin, R.-H. Park, S. Yang, and J.-H. Jung, "Block-based noise estimation using adaptive Gaussian filtering", *IEEE Transactions on Consumer Electronics*, vol. 51, no. 1, pp. 218–226, Feb. 2005. DOI: 10.1109/TCE.2005.1405723.

[12] A. Amer, A. Mitiche, and E. Dubois, "Reliable and fast structure-oriented video noise estimation", in *Proc. of International Conference on Image Processing*, 2002, pp. I–I. DOI: 10.1109/ICIP.2002.1038156.

[13] C.-H. Wu and H.-H. Chang, "Superpixel-based image noise variance estimation with local statistical assessment", *EURASIP Journal on Image and Video Processing*, vol. 2015, art. no. 38, 2015. DOI: 10.1186/s13640-015-0093-2.

[14] W. Liu, "Additive White Gaussian noise level estimation based on block SVD", in *Proc. of the 2014 IEEE Workshop on Electronics, Computer and Applications*, 2014, pp. 960–963. DOI: 10.1109/IWECA.2014.6845781.

[15] S.-M. Yang and S.-C. Tai, "Fast and reliable image-noise estimation using a hybrid approach", *Journal of Electronic Imaging*, vol. 19, no. 3, pp. 033007-1–033007-15, Jul. 2010. DOI: 10.1117/1.3476329.

[16] S. Gai, G. Yang, M. Wan, and L. Wang, "Hidden Markov tree model of images using quaternion wavelet transform", *Computers & Electrical Engineering*, vol. 40, no. 3, pp. 819–832, 2014. DOI:

10.1016/j.compeleceng.2014.02.009.

[17] M. Hashemi and S. Beheshti, "Adaptive noise variance estimation in BayesShrink", *IEEE Signal Processing Letters*, vol. 17, no. 1, pp. 12–15, Jan. 2010. DOI: 10.1109/LSP.2009.2030856.

[18] 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", in *Proc. of TENCON 2017 - 2017 IEEE Region 10 Conference*, 2017, pp. 2447–2450. DOI: 10.1109/TENCON.2017.8228272.

[19] H. Tan, H. Xiao, S. Lai, Y. Liu, and M. Zhang, "Pixelwise estimation of signal-dependent image noise using deep residual learning", *Computational Intelligence and Neuroscience*, vol. 2019, art. ID 4970508, 2019. DOI: 10.1155/2019/4970508.

[20] X. Liu, M. Tanaka, and M. Okutomi, "Single-image noise level estimation for blind denoising", *IEEE Transactions on Image Processing*, vol. 22, no. 12, pp. 5226–5237, Dec. 2013. DOI: 10.1109/TIP.2013.2283400.

[21] G. Chen, F. Zhu, and P. A. Heng, "An efficient statistical method for image noise level estimation", in *Proc. of 2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 477–485. DOI: 10.1109/ICCV.2015.62.

[22] Z. Wang and G. Yuan, "Image noise level estimation by neural networks", in *Proc. of the 2015 International Conference on Materials Engineering and Information Technology Applications*, 2015, pp. 692–697. DOI: 10.2991/meita-15.2015.126.

[23] F. N. Landola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size", *arXiv*, vol. abs/1602.07360. DOI: 10.48550/arXiv.1602.07360.

[24] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation", in *Proc. of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3431–3440. DOI: 10.1109/CVPR.2015.7298965.