







Review

# A Review of Neural Network-Based Image Noise Processing Methods

Anton A. Volkov , Alexander V. Kozlov, Pavel A. Cheremkhin \* , Dmitry A. Rymov , Anna V. Shifrina ,  
Rostislav S. Starikov, Vsevolod A. Nebavskiy , Elizaveta K. Petrova, Evgenii Yu. Zlokazov   
and Vladislav G. Rodin

Laser Physics Department, Institute for Laser and Plasma Technologies, National Research Nuclear University MEPhI (Moscow Engineering Physics Institute), Kashirskoe Shosse 31, 115409 Moscow, Russia

\* Correspondence: cheremhinpavel@mail.ru

## Abstract

This review explores the current landscape of neural network-based methods for digital image noise processing. Digital cameras have become ubiquitous in fields like forensics and medical diagnostics, and image noise remains a critical factor for ensuring image quality. Traditional noise suppression techniques are often limited by extensive parameter selection and inefficient handling of complex data. In contrast, neural networks, particularly convolutional neural networks, autoencoders, and generative adversarial networks, have shown significant promise for noise estimation, suppression, and analysis. These networks can handle complex noise patterns, leverage context-specific data, and adapt to evolving conditions with minimal manual intervention. This paper describes the basics of camera and image noise components and existing techniques for their evaluation. Main neural network-based methods for noise estimation are briefly presented. This paper discusses neural network application for noise suppression, classification, image source identification, and the extraction of unique camera fingerprints through photo response non-uniformity. Additionally, it highlights the challenges of generating reliable training datasets and separating image noise from photosensor noise, which remains a fundamental issue.

**Keywords:** camera noise; denoising; noise estimation; convolutional neural networks; generative adversarial networks; image noise; deep learning; camera source identification; photo-response non-uniformity; synthetic image



Academic Editor: Zhe-Ming Lu

Received: 19 August 2025

Revised: 28 September 2025

Accepted: 1 October 2025

Published: 2 October 2025

**Citation:** Volkov, A.A.; Kozlov, A.V.; Cheremkhin, P.A.; Rymov, D.A.; Shifrina, A.V.; Starikov, R.S.; Nebavskiy, V.A.; Petrova, E.K.; Zlokazov, E.Y.; Rodin, V.G. A Review of Neural Network-Based Image Noise Processing Methods. *Sensors* **2025**, *25*, 6088. <https://doi.org/10.3390/s25196088>

**Copyright:** © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

With the rapid development of digital and computer technologies, digital cameras have become an indispensable tool for various applications [1]. The precision of recorded images can be critical in fields such as forensics [2], copyright protection [3], and medical diagnostics [4]. One of the most important factors affecting the quality of the recorded images is digital camera noise [5]. Despite the fact that this noise often cannot be perceived by the human eye, it can significantly degrade the quality of images, and thus, the accuracy of their analysis. Therefore, noise analysis, evaluation, and suppression are crucial tasks in digital image processing.

Traditional methods for estimating and suppressing noise in images have a number of limitations, such as the need for parameter selection and low efficiency when dealing with large amounts of data or high noise levels [6]. With the advancements in neural network (NN) technology, NN-based methods are emerging that can be more efficient than

traditional approaches. Neural networks, including convolutional neural networks (CNN), are already showing good results in extracting noise parameters from images, as well as in extracting unique noise characteristics such as pixel's photo response non-uniformity (PRNU), which can be important, for example, for image source identification tasks [2].

The adaptability of NN processing allows us to take into account the peculiarities of signal-to-noise ratios in different images, handle heterogeneous data, and continuously improve the NN by adding new training samples. This is especially important as camera resolutions increase and signal distortion models become more complex, including both electronic noise and compression artifacts. In addition, NN algorithms can more effectively take into account additional contextual data, such as meta-information about camera settings or shooting conditions.

This paper discusses current NN-based approaches for noise estimation, suppression, classification, camera characterization, and image source identification. Special attention is paid to CNN-based extraction of camera noise footprints. Advantages and limitations of NN-based methods over traditional algorithms are analyzed.

The paper is structured as follows. In Section 2, camera noise is defined and traditional methods of its estimation are presented. Section 3 describes the main features of NN-based approaches. In Section 4, the application of NN-based methods for noise estimation, suppression, camera identification, etc., are described. The results are discussed in Section 5 and are further evaluated in Section 6.

## 2. Digital Camera Noise

### 2.1. Types of Camera Photosensor Noise

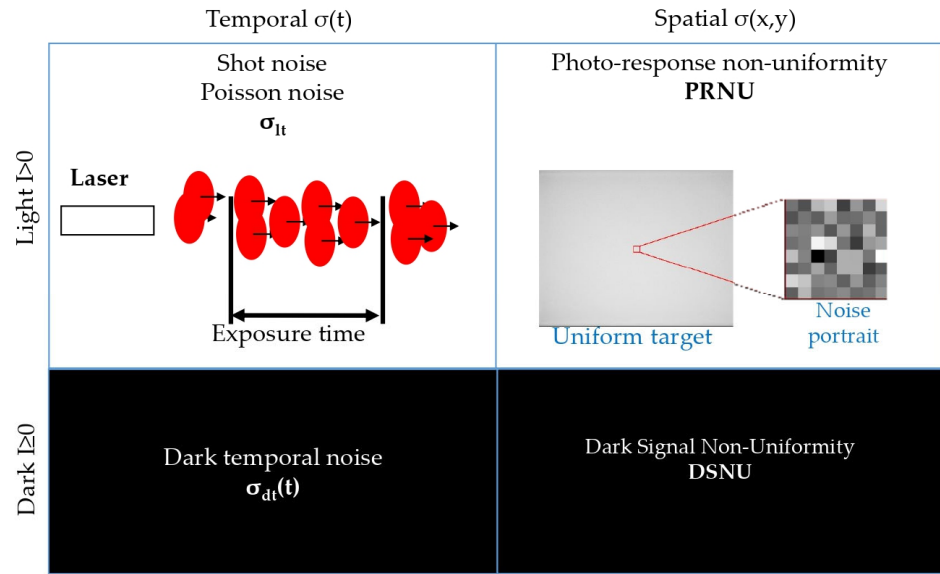
Digital camera noise refers to the distortions introduced into the image during the registration process. These distortions occur due to the imperfections in the camera photosensor, its internal processes, and the physical basis of the registration process itself. It is necessary to distinguish the concepts of the camera noise and noise of the registered image. Image noise is defined as distortions of the already registered image. In this case, digital camera noise is one of the components of image noise. Additionally, image noise depends on such factors as changes in illumination during exposure, changes in camera position, variations in quantum efficiency for different light wavelengths, and others. These factors and the imperfections of current noise estimation algorithms (e.g., from a single image [7]) can lead to other noise parameters being more impactful than camera noise.

Nowadays, various standards are used to measure and characterize digital camera noise. One widely used standard was proposed by the European Machine Vision Association (EMVA) known as the EMVA 1288 standard [8]. This standard describes all noise characteristics of digital photosensors using four parameters. Two of the parameters describe dark noise. They characterize the noise present without external illumination. The other two describe light noise. They depend on the parameters of external illumination of the photosensor. Each type is further separated according to the axis along which the signal is fluctuating. Noise arising from signal fluctuations in time is called temporal, and noise arising from differences in signal from pixel to pixel is called spatial. Thus, camera noise is categorized into the following four components:

- (1) Dark temporal noise.
- (2) Dark spatial noise.
- (3) Light temporal noise.
- (4) Light spatial noise.

The main characteristics of each type of noise are shown in Figure 1. All noise components are expressed in digital number (DN)—the value of the digital signal at the

camera output or in the number of generated charges (electrons). To simplify the equation representations, we will note the noise components in DN.



**Figure 1.** Diagram of noise characteristics according to the EMVA 1288 standard.

Dark temporal noise  $\sigma_{dt}$  is the temporal signal change in the absence of light on the photosensor. Dark temporal noise is caused by dark currents, random charge generation processes in the pixels, etc. Dark spatial noise  $\sigma_{ds}$  is the Dark Signal Non-Uniformity (DSNU). DSNU characterizes the sensitivity of the photosensor to dark currents and differences in the probability of dark charge occurrence. Light temporal noise  $\sigma_{lt}$  is the shot photon noise. It represents the difference in the number of photons hitting the pixels, even when the average illumination is unchanged. Light temporal noise has a Poisson noise statistic representation [9]. Another factor connecting the light temporal noise and the signal  $S$  is the overall system gain of conversion gain constant  $K$  (DN to electrons conversion coefficient) that depends on the bit depth of the camera. The light temporal noise can be written in DN as follows:

$$\sigma_{lt\_DN} = \sqrt{S_{DN} \cdot K_{DN/e}} \quad (1)$$

where  $\sigma_{lt\_DN}$  is light temporal noise (in DNs),  $S_{DN}$  is the signal value (in DNs), and  $K_{DN/e}$  is single value of the constant system gain (in units of DN to electrons ratio). The total temporal noise is as follows:

$$\sigma_{t\_DN} = \sqrt{\sigma_{dt\_DN}^2 + \sigma_{lt\_DN}^2} = \sqrt{\sigma_{dt\_DN}^2 + S_{DN} \cdot K_{DN/e}} \quad (2)$$

where  $\sigma_{t\_DN}$  is total temporal noise (in DNs), and  $\sigma_{dt\_DN}$  is dark temporal noise (in DNs).

Light spatial noise  $\sigma_{ls}$  is characterized by the PRNU value. It is mainly caused by the variations in the photosensitive area of the different pixels. PRNU is usually described as fractions of the signal in a pixel. The light temporal noise in DN can be written as follows:

$$\sigma_{ls\_DN} = PRNU_{rel.un.} \cdot S_{DN} \quad (3)$$

where  $\sigma_{ls\_DN}$  is the light spatial noise (in DNs), and  $PRNU_{rel.un.}$  is the photo response non-uniformity value (in dimensionless relative units of the signal). The total spatial noise is as follows:

$$\sigma_{s\_DN} = \sqrt{\sigma_{ds\_DN}^2 + \sigma_{ls\_DN}^2} = \sqrt{(DSNU_{DN})^2 + (PRNU_{rel.um.} \cdot S_{DN})^2} \quad (4)$$

where  $\sigma_{s\_DN}$  is total spatial noise (in DN),  $\sigma_{dt\_DN}$  is dark temporal noise (in DN), and  $DSNU_{DN}$  is dark signal non-uniformity (in DN).

DSNU and PRNU represent the average of spatial inhomogeneities across the entire area of the photosensor. At the same time, each pixel has its own PRNU and DSNU values, which give us the matrices for these two types of spatial noise. These matrices are unique for each photosensor and are sometimes referred to as photosensor “fingerprints”. They can be used for camera identification, i.e., for determining which camera the image was taken on or whether different images were taken by the same camera [10]. The system gain  $K$  can also be represented as a matrix  $K(i,j)$  [11]. The EMVA 1288 standard recommends that  $K$  be represented as a single value for the entire photosensor. In order to eliminate contradictions, we will follow the Standard recommendations. In our equations  $K$ , PRNU and DSNU are the average values for the whole photosensor, i.e., they are single-value (not matrix).

In addition to the four types of noise mentioned above, the quantization noise, characteristic to the digital systems, is sometimes considered separately [12]. It is digital systems specific and equal to the following (in DN):

$$\sigma_{q\_DN} = 0.288 \text{ DN}. \quad (5)$$

Therefore, the total noise of a camera photosensor can be written as follows:

$$\sigma_{full} = \sqrt{\sigma_t^2 + \sigma_s^2} = \sqrt{\sigma_{lt}^2 + \sigma_{dt}^2 + \sigma_{ls}^2 + \sigma_{ds}^2} \quad (6)$$

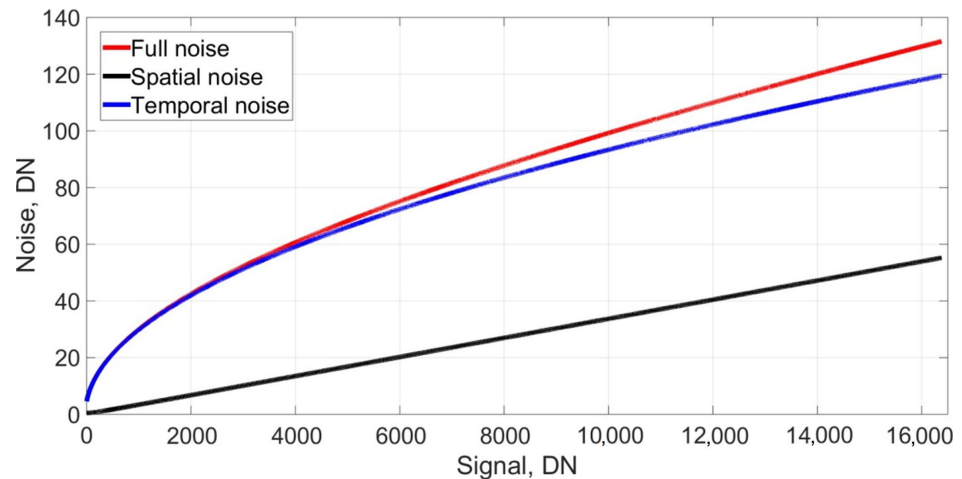
In DN, the total noise (in DN units) can be written as follows:

$$\sigma_{full\_DN} = \sqrt{\sigma_{dt\_DN}^2 + S_{DN} \cdot K_{DN/e} + (DSNU_{DN})^2 + (PRNU_{rel.um.} \cdot S_{DN})^2} \quad (7)$$

Examples of the noise parameters for three cameras are given in Table 1. The cameras have different application areas and are machine vision PixeLink PL-B781F (PixeLink, Ottawa, ON, Canada), microscope Retiga R6 (QImaging, Surrey, BC, Canada), and mirrorless Canon EOS M100 (Canon, Oita, Japan) cameras [13]. Dependences of total, temporal, and spatial noise vs. signal for the camera Retiga R6 are shown in Figure 2.

**Table 1.** Noise and technical parameters of several digital cameras.

Parameters	Canon EOS M100	PixeLink PL-B781F	Retiga R6
Pixel size, $\mu\text{m}$	3.7	3.5	4.6
Full resolution, MP	24	6.6	5.9
Sensor	CMOS	CMOS	CCD
Type	Consumer	Machine vision	Microscopy
Bit depth, bit	13.5	10	14
$\sigma_{dt}$ , DN	$2.479 \pm 0.004$	$0.351 \pm 0.004$	$4.46 \pm 0.01$
$K$ , DN/e	$0.781 \pm 0.006$	$0.093 \pm 0.005$	$0.84 \pm 0.03$
DSNU, DN	$0.191 \pm 0.005$	$0.66 \pm 0.02$	$0.50 \pm 0.01$
PRNU, relative units	$0.0092 \pm 0.0003$	$0.0075 \pm 0.0002$	$0.0033 \pm 0.0001$

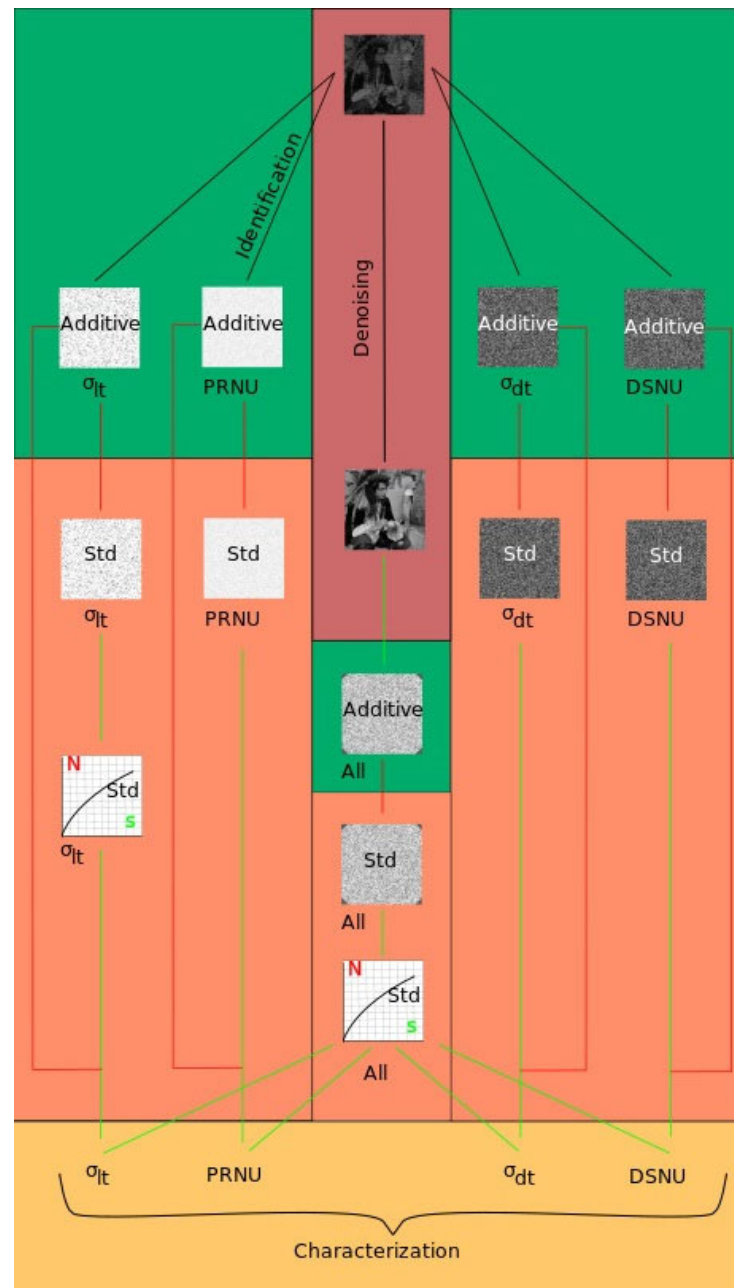


**Figure 2.** Dependencies of spatial, temporal, and total noise vs. the signal level for the camera Retiga R6.

Simplified noise models are also commonly used. They are named after their respective distributions: Gaussian [14], Gaussian–Gaussian [15], Poisson–Gaussian [16] and Poisson model [17]. The Gaussian–Gaussian model uses the sum of two Gaussian functions to describe the distribution of noise across the photosensor. The Poisson–Gaussian model uses the sum of the Poisson and Gaussian distributions. These models are actively used for noise modeling in generated images [18,19], single image noise estimation [19–22], etc. Some analogy can be found between the EMVA 1288 standard and Gaussian–Gaussian and Poisson–Gaussian models. The light temporal noise in the standard and Poisson–Gaussian model is described by a Poisson distribution. In the Gaussian–Gaussian model, it is one of the Gaussian distributions, which can be used instead of a Poisson distribution at high signal levels. In the Poisson–Gaussian model, the remaining noise (dark and light spatial noise) is described by a Gaussian distribution, and in the Gaussian–Gaussian model it is a second Gaussian. In the Gaussian and Poisson models, all noise is considered as one Gaussian or Poisson distribution. It can be seen that in the Gaussian, Gaussian–Gaussian, Poisson–Gaussian and Poisson models, there is no separation into four types of noise. However, in many cases it is important to have information about the individual noise parameters.

## 2.2. Interconnection of Applications

Digital camera noise information can be used for various applications: device selection, camera identification, characterization, noise reduction, etc. To demonstrate the noise representation, a diagram is presented in Figure 3. The green boxes show additive noise matrices (matrices of noises). Values in each pixel are added to the ideal signal. Additive noise values in the matrices can be positive, negative or zero. The orange boxes show noise standard deviation (Std) values (matrices of noise Std). Values of the noise Std in the matrices are non-negative. Also, dependency of the light temporal noise vs. the signal that is the 1D matrix with Std values is located here. The yellow box shows single averaged PRNU, DSNU, dark temporal, and light temporal noise values. The red box indicates the registered image that can be used for various tasks and noiseless image after ideal denoising. Green arrows indicate transitions between tasks that can be performed both in direct and inverse directions. Red arrows indicate one-way transitions and transformations.

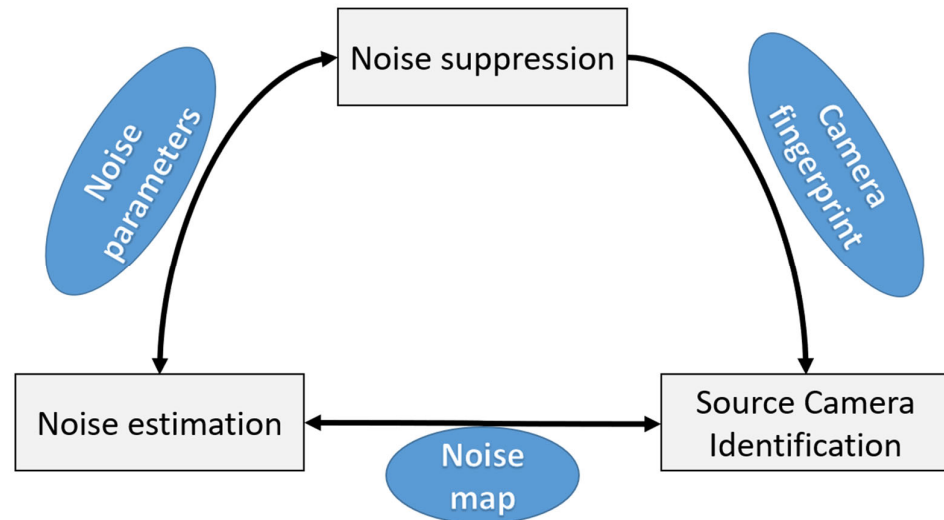


**Figure 3.** Diagram of digital camera noise data representation.

It can be seen that the most preferred option is to obtain all four noise matrices: PRNU( $i,j$ ), light temporal  $\sigma_{lt}(i,j)$ , dark temporal  $\sigma_{dt}(i,j)$ , and DSNU( $i,j$ ). This is the most complete and informative form of noise estimation. It can be successfully used for all applications: photosensor characterization, camera selection, source identification (based on the estimation of the PRNU( $i,j$ ) matrix), and noise suppression (based on the compensation of the noise matrices). However, the problem of obtaining all four matrices from a single image is extremely difficult and currently unsolved. So, simplified options are usually considered. For example, for camera identification, the most important part is the PRNU( $i,j$ ) matrix estimation. For noise suppression, the noiseless image can be used; however, obtaining the noiseless image is also a difficult task which has not yet been solved, either. Various filtration methods require some image noise information. In some methods, pre-filtering is used, estimating noise by finding the standard deviations of the full noise at

all signal levels. This allows us to estimate the camera noise parameters, albeit with low accuracy. After pre-filtering, the noise information is used for the filtering method.

Another option for noise estimation from an image is the direct measurement of the four values characterizing noise: PRNU, DSNU,  $K$ , and dark temporal noise [13]. This can be used to obtain the correlation between signal and noise and standard deviation matrices in a given image. This approach is suitable for both camera characterization and noise suppression, and other tasks (see Figure 4).



**Figure 4.** Interconnectivity between different practical applications and camera and noise parameters.

### 2.3. Noise Estimation for Characterization, Denoising, and Identification

The EMVA 1288 [8] standard is a robust algorithm for accurate estimation of all noise parameters as well as for radiometric characterization. However, it is challenging to implement in real systems [23]. A special uniform scene is required that fulfills the following criteria:

$$\frac{E_{max} - E_{min}}{\mu_E} \cdot 100 < 3\% \quad (8)$$

where  $E_{max}$ ,  $\mu_E$ , and  $E_{min}$  are the maximum, average, and minimum signal in the scene. It is recommended to use a light source with the full width at half maximum, less than 10 nm. Therefore, it is required to assemble a special setup to provide the necessary uniformity of the recorded scene. It is recommended to register at least 50 sets of images (at least two images each). The sets should be registered such that the average frame signal level is distributed evenly from the minimum to the maximum camera signal. Processing of the captured images starts with the calculation of dark temporal and dark spatial noise:

$$\sigma_{DTN} = \sqrt{\frac{\sum_{i,j} (D_1(i,j) - D_2(i,j))^2}{2MN}} \quad (9)$$

$$DSNU = \sqrt{\frac{\sum_{i,j} (\overline{D(i,j)} - \overline{D})^2}{(MN-1)} - \sigma_{DTN}^2} \quad (10)$$

$$\overline{D(i,j)} = \frac{D_1(i,j) + D_2(i,j)}{2} \quad (11)$$

$$\overline{D} = \frac{\sum_{i,j} \overline{D(i,j)}}{MN} \quad (12)$$

where  $D_{1,2}(i,j)$ —dark images taken without external illumination,  $M, N$ —their dimensions,  $i,j$ —row and column coordinates.

Next, the light temporal noise and the PRNU value can be calculated:

$$\sigma_{LTN} = \sqrt{\frac{\sum_{i,j}(L_1(i,j) - L_2(i,j))^2}{2MN} - \sigma_{DTN}^2} \quad (13)$$

$$PRNU = \frac{\sqrt{\frac{\sum_{i,j}(\overline{L(i,j)} - \overline{L})^2}{(MN-1)} - \sigma_{LTN}^2 - \sigma_{DTN}^2 - DSNU^2}}{\overline{L} - \overline{D}} \quad (14)$$

$$\overline{L(i,j)} = \frac{L_1(i,j) + L_2(i,j)}{2} \quad (15)$$

$$\overline{L} = \frac{\sum_{i,j} \overline{L(i,j)}}{MN} \quad (16)$$

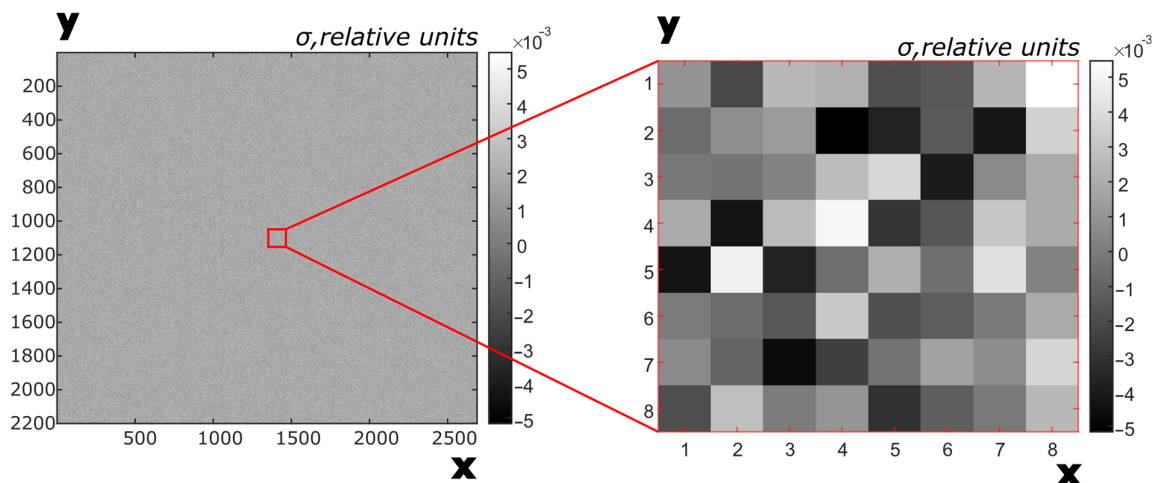
where  $L_{1,2}(i,j)$  are light images taken with external illumination.

Within the framework of the EMVA 1288 standard,  $DSNU(i,j)$  and  $PRNU(i,j)$  matrices can be measured. This requires at least 100 dark images and at least 100 light images to be registered at the average signal level of the photosensor. This number of shots is necessary to reduce the impact of the temporal noise. After that, the  $DSNU(i,j)$  and  $PRNU(i,j)$  matrices can be obtained:

$$DSNU(i,j) = \overline{D_N(i,j)} - \overline{D}, \quad \overline{D} = \frac{\sum_{i,j} \overline{D_N(i,j)}}{MN}, \quad \overline{D_N(i,j)} = \frac{\sum_{k=1}^N D_k}{N}, \quad (17)$$

$$PRNU(i,j) = \frac{\overline{L_N(i,j)} - \overline{D_N(i,j)}}{\overline{L}}, \quad \overline{L} = \frac{\sum_{i,j} \overline{L_N(i,j)}}{MN}, \quad \overline{L_N(i,j)} = \frac{\sum_{k=1}^N L_k}{N} \quad (18)$$

where  $N$  is the number of images, and  $k$  is the image index. An example of a  $PRNU(i,j)$  matrix is shown in Figure 5.



**Figure 5.** Example of a photo response non-uniformity of photosensor (PRNU matrix).

Since the EMVA 1288 standard is very labor-intensive, other approaches that require a smaller number of images have been developed. Methods for fast and accurate noise estimation are based on automatic scene segmentation [13]. Segmentation involves sorting signals and their corresponding deviations between images. Automatic segmentation has been successfully applied to inhomogeneous [24], homogeneous [25], and bandpass [26] scenes. Segmentation of an inhomogeneous [24] scene allows for estimation of temporal

noise from two shots, but it cannot correctly estimate spatial noise. The homogeneous scene segmentation [25] gives an accurate estimate of all four noise components over four shots: two homogeneous and two dark frames. Strip scene segmentation [26] estimates all four noise components from two shots with four stripes. The noise estimates obtained by these methods can be used for every application: characterization, camera comparison, identification, noise reduction, etc.

A variety of methods for noise suppression have been developed. They estimate the noise from a single image, which then will be filtered. These methods can be separated into three main groups: filtration-, wavelet transformation-, and patch-based. Filtration-based methods are the least complex [27–30]. By subtracting the filtered image from the image and removing the edges in the resulting image, noise can be estimated. Considering the challenges of parameter selection for both steps of the algorithm, obtaining a reliable filtration-based noise estimation method is an extremely difficult task.

A more sophisticated approach is based on wavelet transformation. The resulting wavelet coefficients are used for noise estimation [31–35]. However, the residual image edges degrade the noise estimation quality. To reduce their impact, in [33] edges were removed, and in [35] an upper signal threshold was implemented.

Currently the most promising approach is based on image patches [36–43]. Each image is divided into patches. Signal fluctuations in each patch are assumed to only be caused by noise. In [36,37], threshold selection is applied to select the most uniform patches. In [39], the eigenvalues of the covariance matrix in redundant dimensions were calculated to estimate the noise.

For camera identification [44], the two main goals are to determine whether images were taken by a single camera and whether an image was taken by a specific camera. Using smoothing filtration, the high frequency component of the image is identified. Dividing it by the original image allows us to obtain the relative deviation magnitudes (image portrait). To identify whether images were taken using a single camera, the noise estimates from different images are compared (e.g., using the correlation coefficient). Although noise estimation is not particularly accurate, it is usually enough to reliably determine whether the images were taken with the same camera.

To determine whether an image was taken by a specific camera, it is necessary to measure its spatial noise characteristics. For this, hundreds of homogeneous light and dark images are recorded. The light images are averaged (to suppress the temporal noise), and the averaged dark image is then subtracted and divided by the average signal. This allows us to obtain the PRNU( $i,j$ ) matrix. The image portrait is then numerically compared to this noise portrait. This approach has been shown to be effective even in the case of JPG compression, which indicates its reliability and efficiency.

### 3. Fundamentals of Neural Network-Based Image Processing

#### 3.1. Training of Neural Networks

NNs require training before they can be used. Training is a process of nonlinear optimization of the NN's parameters. The difference between the NN's output and the expected result is calculated using a loss function [45]. For noise estimation and processing, both supervised and unsupervised training can be applied [1].

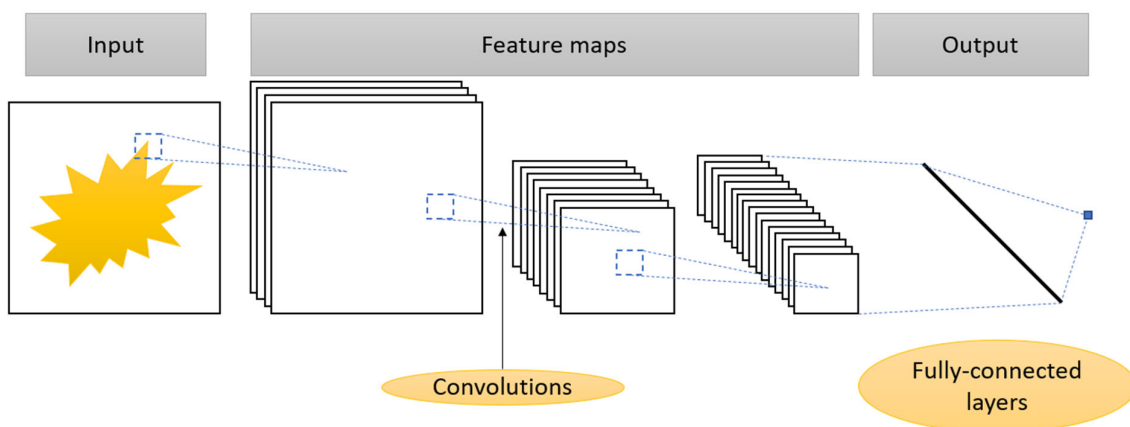
The training datasets can be separated into real and synthetic datasets. Quality of training datasets for NN processing of noise depends on the variety of models used to generate data, noise levels, dataset size, and the resolution of individual images. Real datasets contain images captured with pre-calibrated photosensors. The calibration process and the reference noise extraction are typically very resource- and time-intensive. This can

limit the size of commonly used datasets such as DID [46], SIDD [47], DND [48], SID [49], ELD [50], BSD300 [51], McMaster [52], BSD500 [53], Nam [54], and others.

The lack of available real training data can partially be addressed by numerical noise models, including NN-based [55–66] noise modeling techniques. Synthetic datasets are usually generated by adding noise to noise-free images, which allows us to precisely control all aspects of the noise in each image. This helps create optimal numerical models for rapid data generation that can be used for NN training. However, the properties and limitations of numerical noise models can also decrease the training efficiency.

### 3.2. Convolutional Neural Networks

Convolutional neural networks are the most common type of NN for image processing and noise suppression applications [1]. Convolutional neural networks are a class of deep neural networks inspired by the structure of the visual cortex of the brain [62]. CNNs outperform fully connected NNs in processing data with topological structure, such as images, video, audio, and other types of multidimensional signals. A typical structure of convolutional neural networks is shown in Figure 6.



**Figure 6.** Example of a convolutional neural network architecture.

In contrast to fully connected NNs, where each neuron in one layer is connected to all neurons in the previous layer, in CNNs each neuron is connected only to a small local area in the previous layer—the so-called local receptive field [63]. This structure reflects the biological organization of the visual cortex, where neurons respond to stimuli in a limited part of the visual field [62]. This reduces the number of training parameters and makes the NN output dependent on the spatially localized features such as edges, shapes or textures. Another feature of CNNs is the use of identical parameters throughout the input layer to compute the activation function in a single receptive field. These common weights form the convolution kernel, which serves as a filter for extracting certain types of local features regardless of their position in the image. This ensures translational invariance, which allows us to form a map of feature distributions across the image [63,64].

The basic structure of a CNN architecture includes a sequence of alternating convolutional and pooling layers. The input layer accepts multidimensional data, such as a “width × height × number of channels” tensors for RGB images. The output of the convolution layer can be expressed as follows:

$$y(i, j) = \sum_m \sum_n x(i + m, j + n) \cdot k(m, n) \quad (19)$$

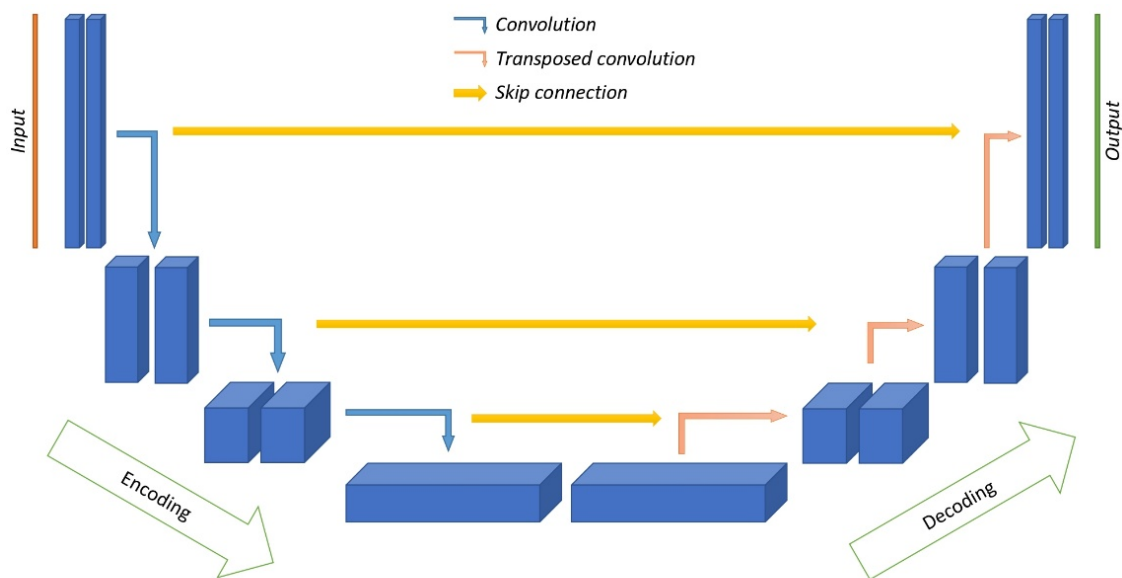
where  $x(i, j)$  is the input image,  $k(m, n)$  is the convolution kernel, and  $i, j$  and  $m, n$  are the indices of the output matrix and the convolution kernel, respectively. Each kernel forms its

own feature map and several kernels form the output tensor with the corresponding number of channels. The size of the output feature map is determined by the input parameters, the kernel size, the convolution step, and the edge processing (padding) [63,65].

Rectified Linear Unit (ReLU,  $f(x) = \max(0, x)$ ) is the most commonly used activation function in CNNs [66]. ReLU has low computational complexity, is effective against vanishing gradients, and can be biologically interpreted.

The current development of CNNs, such as VGG [65], ResNet [67], and Inception [68], is built on improving the depth and organization of layers, introducing attention mechanisms, and optimizing information flow while maintaining the basic principles.

Currently, the most commonly used architecture for image processing [69] is the U-net architecture, developed in 2015 for segmentation of biomedical images [70]. The main elements are shown in Figure 7.



**Figure 7.** The U-Net architecture.

The U-Net is named after its characteristic U-shape and consists of two symmetric parts: the encoder and the decoder. The encoder is responsible for feature extraction and compression of spatial information. The encoder forms the hierarchy of features. The initial layers extract simple features such as edges, corners, and homogeneous regions. These feature maps have high spatial resolution but contain relatively few channels. In the middle layers, simple features are combined into more complex features such as geometric shapes, parts of objects, and characteristic textures. Here the resolution decreases and the number of channels increase. In the last layers, high-level, semantic representations are formed—complete objects and their characteristic combinations. The resulting feature maps have minimal spatial resolution but a large number of channels, reflecting the abstract properties of the input data [63,65]. The feature extraction is performed by the encoder blocks which include two consecutive convolution operations each and a ReLU activation function.

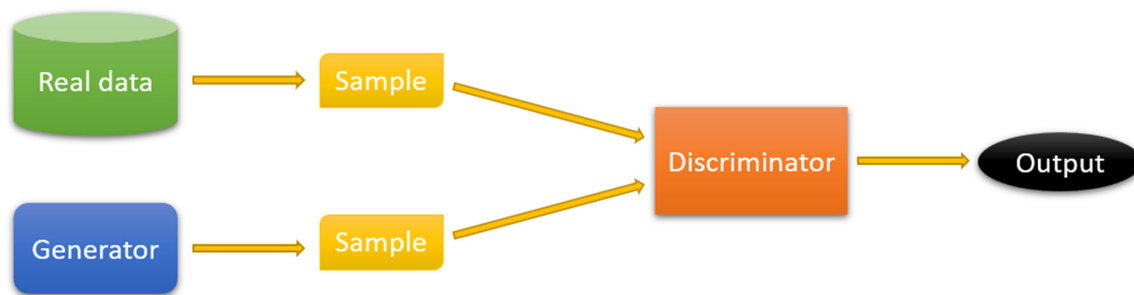
The convolution operation is followed by a pooling operation, which performs spatial subsampling, i.e., reducing the size of the feature map. The most commonly used pooling layers are max pooling, which selects the most pronounced features, and average pooling, which provides smoothing [64]. Pooling reduces computational complexity, reduces the risk of overtraining, and increases the invariance of the network to scale and affine transformations of the input data. Pooling has no trainable parameters [64]. The purpose of the

compression branch is to reduce the spatial resolution of feature maps and increase the number of channels to extract abstract high-level features.

The decoder performs the opposite operation—the resolution of feature maps is increased in order to restore accurate localization of objects. Here, at each decoder block the size of the feature maps increases, while the number of channels is reduced. The output of each decoder block is also concatenated with the corresponding feature maps from the encoder. These skip-connections allow the integration of high-level semantic information with high spatial resolution recovered from the initial layers of the encoder. The skip-connections not only compensate for the loss of spatial information due to pooling but also provide integration of contextual and localized information. As a result, pixel classification relies not only on the properties of the full scene but also on localized features.

### 3.3. Generative-Adversarial Network Architecture

Generative-adversarial networks (GANs) are a class of neural network algorithms based on the training of two CNNs at the same time. The GAN model implements an adversarial model between two NNs, a generator and a discriminator. The purpose of the generator is to create synthetic samples that mimic real data. In turn, the discriminator has to distinguish between the real samples and samples created by the generator. An example of a GAN model is shown in Figure 8.



**Figure 8.** Example of a GAN model.

The training of a GAN model is interpreted as an optimization problem with a min-max loss function, where the discriminator seeks to maximize the probability of correct classification and the generator seeks to minimize the probability of the discriminator detecting synthetic data. This process involves updating the parameters of each network in turn. First, the generator's parameters are assumed to be unchanged, and the discriminator is trained to distinguish between real and synthetic data based on batches containing both types of samples. The generator then optimizes its parameters to overcome the current discriminator parameters. It is critical to maintain balance between the generator and the discriminator. If one part significantly outperforms the other, the overall performance of the model can degrade.

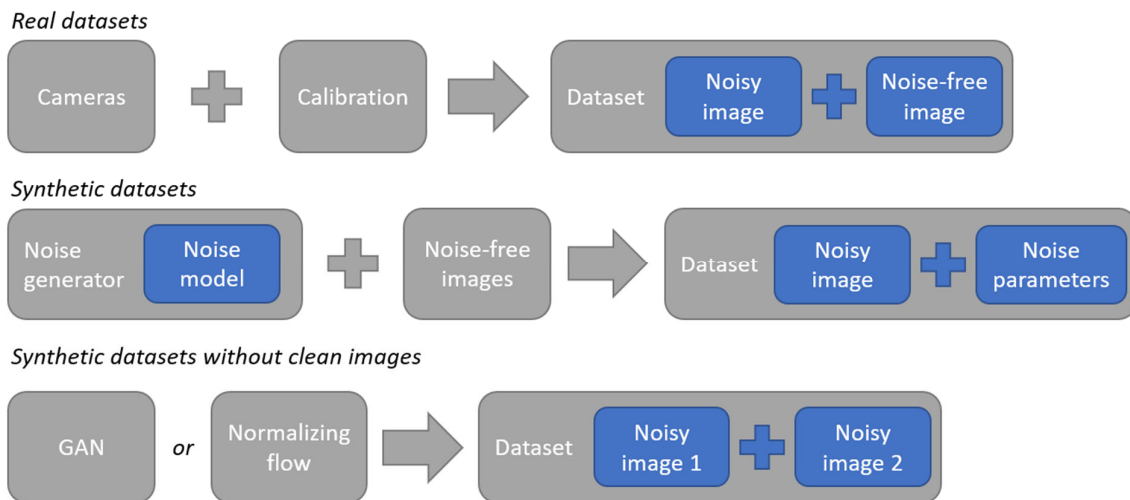
In the context of NN noise processing, there are GAN models for synthesizing noisy images with spatial noise correlation [71,72], as well as models for noise suppression [73].

## 4. Practical Applications of Neural Network Methods

### 4.1. Synthesis of Datasets and Noise Modeling

Synthetic datasets for NN training can save time and resources on preparing real datasets of noisy and noise-free images [74] (see Figure 9). At the same time, synthetic datasets are often used for noise estimation [55]. This is demonstrated by the compact NoiseFlow model, which has fewer than 2500 parameters. It is one of the first NN-based models for camera sensor noise estimation [56]. NoiseFlow uses conditional normalizing

flow to model complex signal-dependent and gain-dependent noise beyond parametric representations. The model was trained and evaluated on the SIDD dataset [47], which contains approximately 30,000 thousand raw-RGB image pairs from five smartphone cameras, captured under various ISO exposure levels (50–10,000) and lighting conditions (low, normal, and high brightness; 3200 K, 4400 K, and 5500 K color temperature). The dataset covers a range of noise levels, with the standard deviation of Gaussian noise of approximately 1.62 to 23.5. On this dataset, NoiseFlow achieved a negative log-likelihood (NLL) of  $-3.521$  nats/pixel on the test set. This represents a 0.42 nats/pixel improvement (or a 51.6% increase in likelihood) over the camera-calibrated noise level functions (NLF) and a 0.69 nats/pixel improvement (a 99.4% increase in likelihood) over a homoscedastic additive white Gaussian noise (AWGN) model. Furthermore, noise samples generated by the NoiseFlow exhibited a significantly lower marginal Kullback–Leibler (KL) divergence (0.008) from the real noise compared to the camera NLF (0.052) and AWGN (0.394) base-lines. In similar noise levels (e.g., medium to high noise conditions with standard deviation values between 4.79 and 23.5), the model’s superior performance in both density estimation and noise synthesis demonstrates its ability to capture the complex, non-Gaussian characteristics of real photosensor noise more accurately than traditional parametric models. However, the model’s disadvantages include its primary design for raw-RGB sensor data, making it less suitable for processed sRGB images where the noise distribution is altered by the camera’s image signal processor. Additionally, the computational complexity of inverting normalizing flows for sampling could pose challenges in memory- or latency-constrained scenarios.



**Figure 9.** Different types of noise datasets that can be used for neural network training.

The concept of NN-based noise generators has evolved with the development of the CANGAN framework, which takes into account camera features by training on real photosensor noise [57]. The CANGAN framework consists of two parts: the U-Net [70]-based Noise-Generator Network and the Camera-Encoder Network. The Camera-Encoder Network is used to extract a camera-specific latent vector from a noisy image, while the Noise-Generator Network can mimic the camera noise based on a clean image and camera parameters. This allows the model to generate noise for a particular camera sensor. The proposed method was trained and evaluated on approximately 24,000 raw-RGB image pairs from the SIDD dataset [47], captured under various ISO exposure levels (from 50 to 10,000) and lighting conditions (low and normal illumination). Quantitative evaluation demonstrated that CANGAN significantly outperforms existing models, achieving a KL divergence of 0.00159, which represents a 74.7% improvement over the Poisson–Gaussian

model (0.01006) and a 72.0% improvement over the NoiseFlow model (0.00912) [56]. Similarly to NoiseFlow [56], CANGAN's lower KL divergence indicates a superior ability to capture the complex, camera-specific noise distributions present in real data. When the CANGAN was used to generate training data for the DnCNN [75] denoiser, it achieved peak signal-to-noise ratio (PSNR) [76] of 48.71 dB and structural similarity index measure (SSIM) [77,78] of 0.992, outperforming the same denoiser trained on NoiseFlow-generated data (48.52 dB, 0.991) and real data (48.30 dB, 0.989). On the other hand, an important disadvantage of this approach is its reliance on a camera encoder that requires a sample noisy image from the target camera in order to extract the latent vector, which may limit its applicability in scenarios when a sample cannot be obtained. Furthermore, the adversarial training framework and the U-Net [70] architecture of the generator introduce significant computational complexity and memory demands compared to simpler parametric models.

Subsequently, a contrast training framework was proposed to estimate fine-grain camera noise parameters, such as the variance of light shot noise or readout noise, from a single raw image [58]. This ResNet-based [67] framework employs a contrastive learning strategy to train a noise estimation model. It is trained on a dataset of carefully calibrated camera sensors (including Canon EOS 5D4, Nikon D850, Sony RX100VI, and HUAWEI P40 Pro), where noise parameters for each image are represented as a four-tuple  $(K, \sigma, \mu_c, \sigma_r)$ , encompassing overall system gain  $K$ , readout noise standard deviation  $\sigma$ , mean readout noise (or color bias)  $\mu_c$ , and row noise standard deviation  $\sigma_r$ . The model uses contrastive loss to maximize agreement between augmented views derived from the same noise parameters. This enables it to separate and accurately estimate the components of a complex, physics-based noise model that includes signal-dependent Poisson shot noise and multiple signal-independent Gaussian noise sources. The estimated noise parameters are used to define a joint distribution as a base for sampling and generation of realistic training data. This generated data, when used to train a denoiser, achieved a competitive PSNR of 50.13 dB and SSIM of 0.9891 on the SIDD benchmark, outperforming methods based on AWGN (47.55 dB, 0.9698), Poisson–Gaussian (49.91 dB, 0.9896), NoiseFlow (44.96 dB, 0.9517) [56], and CANGAN (49.19 dB, 0.9879) [57], and approaching the performance of training with real paired data (50.10 dB, 0.9902). The proposed fine-grained model's superior parameter estimation leads to synthetic noise that closely matches the real noise distribution. This is quantitatively reflected in the lower KL divergence achieved by the synthesis pipeline (average of 0.0211 across five SIDD cameras) compared to AWGN (0.7544), Poisson–Gaussian (0.0467), NoiseFlow (0.0590) [56], and CANGAN (0.0220) [57]. However, the current restriction to Bayer color filter arrays (CFAs) limits the approach, making it unsuitable for sensors with alternative CFAs like RYYB, RGBC, RGBW, RWB or X-Trans. Furthermore, while the original paper does not provide explicit computational metrics, the complexity of the contrastive learning framework and the deep ResNet [67] backbone suggests the associated computational costs for training are expected to be high.

In [59], an unsupervised GAN training framework C2N for noise modeling was proposed. The C2N does not require the preparation of the training dataset of noisy and noise-free images. Instead, it trains using unpaired sets of clean images and real noisy images. This approach explicitly separates the noise generation process into dedicated modules for signal-dependent and signal-independent components, further enhanced by convolutional transformations to model spatial correlations inherent in real camera noise. The framework was trained and evaluated on real-world benchmarks including SIDD [47] and DND [48] datasets. Quantitative results demonstrate that a denoiser trained with data generated by the C2N significantly outperforms other unsupervised methods, achieving a PSNR of 33.76 dB and SSIM of 0.901 on the SIDD benchmark with a DnCNN [75] backbone, and 35.35 dB/0.937 with a larger DIDN [79] backbone. This represents an

improvement of over 1 dB in PSNR compared to previous generation methods like UIDNet (32.48 dB) [80]. In the complex real-world noise conditions present in the SIDD [47] and DND [48] benchmarks, which include signal-dependent (e.g., photon shot noise) and signal-independent components with spatial correlations, the C2N's modular design enables it to more accurately capture the true noise distribution. This is quantitatively supported by its lower KL divergence (0.1638) compared to AWGN (0.1746) when comparing generated noise to the ground truth. A key disadvantage of this approach is the framework's inherent instability during adversarial training, which can be partially mitigated by the proposed stabilizing loss term to prevent color shifting. In addition, while the C2N eliminates the need for paired data, its performance is still contingent on having access to a representative set of unpaired clean and noisy images from the target domain. Finally, the computational complexity of the framework's architecture is not explicitly specified and heavily depends on the backbone denoiser used.

The emergence of unsupervised NN models has led to the creation of Noise2NoiseFlow [60] based on the Noise2Noise model [61] with integrated normalizing flow. The Noise2NoiseFlow training uses pairs consisting of two independent noisy images of the same scene, instead of the traditional pairs of noisy and noise-free images, thereby eliminating the need for clean ground truth data. The framework was evaluated on the SIDD dataset [47], utilizing approximately 500,000 patches  $32 \times 32$  pixels each, with ISO exposure levels from 100 to 3200 with a 70/30 train/test split. The jointly trained model achieved NLL of  $-3.501$  nats/dim and a KL divergence of 0.0265 on the test set. This makes the Noise2NoiseFlow performance on par with the fully supervised NoiseFlow model (NLL:  $-3.502$ , KL: 0.0267) and significantly superior to statistical baselines like the camera NLF (NLL:  $-3.282$ , KL: 0.0578) and AWGN (NLL:  $-2.874$ , KL: 0.4815). Furthermore, the integrated DnCNN [75] denoiser achieved a PSNR of 52.80 dB and SSIM of 0.984, outperforming a DnCNN trained with standard Noise2Noise (51.57 dB) and even a supervised DnCNN trained on real clean/noisy pairs (51.32 dB). Despite its advantages, the method requires precisely aligned pairs of noisy images from the same scene. This requirement poses an additional challenge for practical applications.

The DCD-Net [81] was proposed in order to address the lack of quality training data. DCD-Net is a system with an iterative "denoise—corrupt (add synthetic noise)—denoise" algorithm. For each input image, the system iteratively suppresses the noise, estimates the noise characteristics, and adds the noise back. Repeating this cycle, the system can train only on noisy images, without any noise-free samples. The proposed method was evaluated on both synthetic and real-world benchmarks. On synthetic sRGB datasets (Kodak, BSD300 [51], Set14 [82]) with Gaussian noise, DCD-Net achieved a PSNR/SSIM of 32.27/0.881, 31.01/0.876, and 31.29/0.862, respectively, closely matching the performance of the strongly supervised Noise2Noise [61] baseline. For real-world denoising, on the SIDD [47] raw-RGB validation set, it achieved a PSNR of 51.40 dB and a SSIM of 0.992, surpassing other self-supervised methods, including Blind2Unblind [83] which requires 17 times more computational resources. The method assumes the noise can be modeled by a Poisson–Gaussian distribution and employs a dedicated noise estimation network trained with a patch variance loss to predict a pixel-wise noise level map. However, the framework's iterative nature increases its computational cost up to three times when compared to a standard U-Net architecture [70]. Furthermore, its performance is contingent on the accuracy of the initial blind-spot network used for bootstrapping and may degrade if the underlying noise deviates significantly from the assumed heteroscedastic Gaussian model.

In [74], further improvements to contrast training approach were made, resulting in a noise estimation and generation framework that does not require pre-calibrated sensors. Similarly to [58], the framework is based on a fine-grained physics-based noise model that

includes shot noise, readout noise, row noise, and color bias. The proposed framework was trained using a specialized data augmentation strategy that generates diverse positive and negative samples by selectively varying individual noise parameters while ensuring distinct scene content. It achieved state-of-the-art noise parameter estimation performance. This approach enables the model to focus accurately on different noise components, enhancing its discriminative capability and robustness. The method achieves a mean absolute error of 0.1598 for color bias  $\mu_c$  and 0.8621 for row noise variance  $\sigma_r$  on the SIDD [49] dataset. On the SIDD dataset [47], the synthesized noise achieved an average KL divergence of 0.0198 across five smartphone cameras, outperforming statistical models (0.0285 for AWGN and 0.0247 for Poisson–Gaussian) and learning-based methods (0.0223 for NoiseFlow [56] and 0.0215 for CANGAN [57]) that require in-camera parameters or paired training data. Using this pipeline to generate training data for the denoiser, the U-Net [70] reached a PSNR of 51.40 dB on the SIDD [47] validation set, exceeding methods trained on other synthetic data such as AWGN (49.01 dB), Poisson–Gaussian (49.93 dB), NoiseFlow [56] (50.54 dB), CANGAN [57] (50.12 dB), and SINE [39] (50.77 dB), and matching models trained on real paired data (51.42 dB). The superior PSNR results confirm that training with more realistic, fine-grained noise synthesis improves denoising performance across different noise levels, particularly in challenging real-world conditions. However, the model’s complexity is substantial at 6.10 GFLOPs with 12.34M parameters. Similarly to the previous model [58], the proposed pipeline is currently limited in applicability and is not designed to accommodate CFA patterns other than the Bayer CFA.

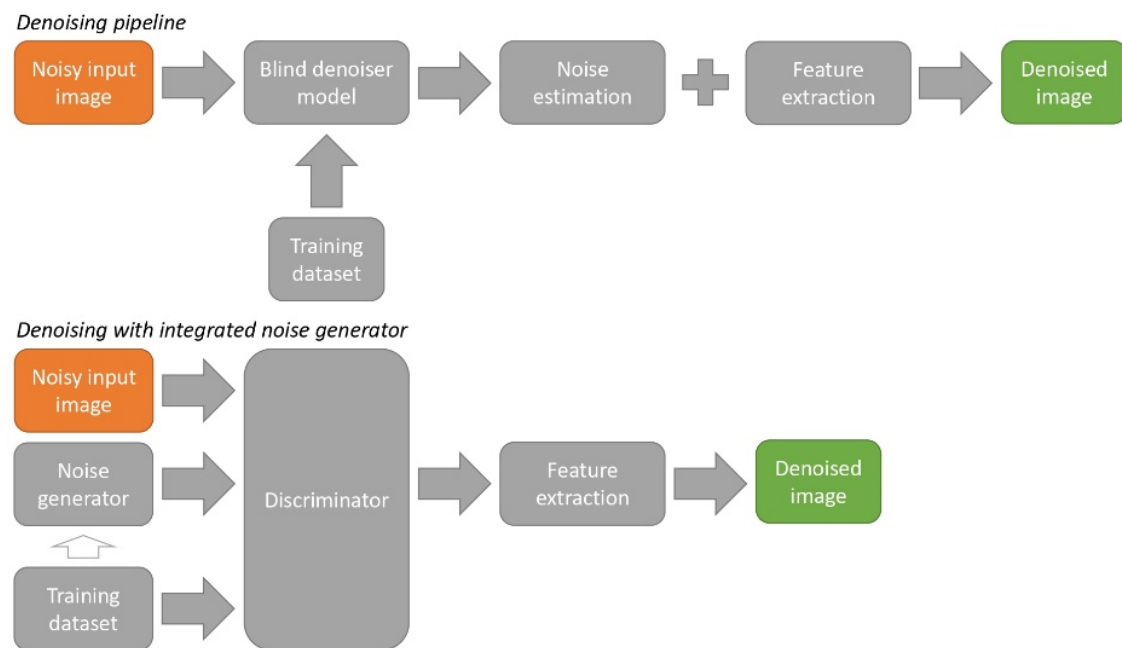
Table 2 presents the summary of methods discussed in this section.

**Table 2.** Noise synthesis and modeling neural network architectures comparison.

Architecture	Numerical Results	Training Conditions and Datasets	Notes
NoiseFlow [56]	NLL: $-3.521$ nats/pixel KL: 0.008	SIDD [47] with $\sim 30,000$ raw–RGB image pairs, 5 smartphone cameras, ISO 50–10,000	Conditional normalizing flow for complex signal-dependent noise modeling, $<2500$ parameters
CANGAN [57]	KL: 0.00159, DnCNN [75]: PSNR: 48.71 dB, SSIM: 0.992	SIDD [47], $\sim 24,000$ raw–RGB image pairs	U-Net [70] based noise generator + camera encoding network for camera-specific noise
ResNet-based frameworks [67,74]	KL: 0.0211 DnCNN [75]/U-Net [70]: PSNR: 50.13/51.40 dB, SSIM: 0.9891/–	SIDD [47], SID [49] Canon EOS 5D4, Nikon D850, Sony RX100VI, HUAWEI P40 Pro	Contrastive learning for fine-grain noise parameter estimation with 4-tuple model
C2N [59]	KL: 0.1638 DnCNN [75]/DIDN [79]: PSNR: 33.76/35.35 dB, SSIM: 0.901/0.937	SIDD [47], DND [48], unpaired clean and noisy images	Unsupervised GAN for noise modeling without paired data
Noise2NoiseFlow [60]	NLL: $-3.501$ nats/dim, KL: 0.0265, DnCNN [75]: PSNR: 52.80 dB, SSIM: 0.984	SIDD [47], $\sim 500,000$ patches $32 \times 32$ , ISO 100–3200	Combines Noise2Noise [61] with normalizing flow, eliminates need for clean ground truth
DCD-Net [81]	PSNR: up to 51.40 dB, SSIM: up to 0.992	Kodak, BSD300 [51], Set14 [82], SIDD [47] raw–RGB validation	Iterative training «denoise-corrupt-denoise» on noisy images only, denoising enhancement

#### 4.2. Denoising

NN-based noise suppression methods explicitly or implicitly determine noise properties during operation, often taking into account the signal level (see Figure 10). In this case, the noise estimation can be obtained both for individual pixels and for the whole image. NNs explicitly estimating the noise parameters can extract several spatial pixel features [84]. Typical examples of noise-aware CNNs with explicit noise estimation are DnCNN [75] and ADNet [85], modifications of the U-net architecture: DANet [70,86]. Accurate estimators that take into account the nature of the noise can improve the efficiency of noise suppression algorithms [87]. However, image noise is usually not separated from camera noise in noise suppression applications.



**Figure 10.** Noise suppression pipeline using neural networks.

One of the first NNs to perform signal-dependent noise estimation for each pixel is DRNE [88]. Building on earlier studies, the authors assume the statistical noise distribution to be Gaussian with variance

$$\sigma^2 = \sigma_r^2 + \sigma_s y_p \quad (20)$$

where  $\sigma_r$  is readout noise,  $\sigma_s$  is shot light noise, and  $y_p$  is the true illumination intensity. This model explicitly captures the signal-dependent nature of noise, where noise intensity increases with scene brightness. The DRNE is a deep convolutional network consisting of 16 layers with 64 channels each, carefully designed without pooling or interpolation layers to preserve image details. The DRNE takes noisy RGB images as input and outputs a pixelwise noise variance map reflecting the spatially varying, signal-dependent noise levels. Quantitative evaluations were conducted on three image datasets—Kodak (24 images), McMaster [52] (18 images), BSD500 [53] (500 images)—using both homogeneous and inhomogeneous synthetic Gaussian noise.

For homogeneous noise with fixed variance, the DRNE achieved average noise estimation error competitive with the state-of-the-art SINE method [39]. For example, on the Kodak dataset at noise level  $\sigma = 5$ , the DRNE achieved an average error of 0.08 dB compared to the SINE's 0.09 dB. On BSD500 at  $\sigma = 0$ , the DRNE scored 0.32 dB average error vs. the SINE's 0.54 dB. These results demonstrate that the DRNE performs on par with leading scalar estimation methods even under the simplified homogeneous noise

assumption. For inhomogeneous noise, where images were synthetically divided into four parts with different noise levels, the DRNE produced accurate pixelwise noise maps closely matching the ground truth spatial patterns. These maps revealed expected signal-dependent characteristics, such as lower noise estimates in flat image regions that contain less high-frequency content. When incorporated into deep denoising networks like the FFDNet [89], the DRNE's pixelwise noise maps enabled significant gains in denoising performance for realistic, signal-dependent noise. For instance, on the Kodak dataset under the Poisson–Gaussian noise model, the DRNE combined with the FFDNet achieved a PSNR of 33.68 dB, surpassing the SINE method with 33.24 dB, while reducing visible artifacts in denoised images and improving visual quality. This performance gain highlights the practical benefit of using pixelwise noise estimation for denoising under more realistic, signal-dependent noise conditions. However, the DRNE's computational demands are relatively high. Processing a  $500 \times 500$  image took about 5.31 s on a CPU and 1.16 s on a GPU, which is considerably slower than the fastest scalar noise estimation method SINE that runs in 0.27 s on CPU. This makes the DRNE less favorable for applications requiring real-time or low-latency noise estimation. Additionally, the DRNE was trained solely on synthetic homogeneous noise with  $\sigma$  uniformly sampled in, which may limit its ability to generalize to more complex real-world noise that deviates from the Poisson–Gaussian assumption or exhibits spatial correlations absent from the training data. This could reduce performance on noisy images with sensor-specific or correlated noise patterns not well captured by the training distribution.

Denoising efficiency and noise estimation accuracy can be enhanced by application of frequency–domain attention mechanisms [90]. The FADNet builds upon a conventional encoder–decoder backbone by interleaving a sequence of Adaptive Instance Residual Modules (AIRMs) that fuse spatial features with multi-scale frequency cues. Additionally, input images are preprocessed via a CNN noise map extractor. The noise map extractor and frequency attention mechanism allow for better preservation of high-frequency details in the spatial spectrum. This also includes spectral noise components. The FADNet attains superior efficiency compared to contemporary denoisers. On  $512 \times 512$  test images processed on the GPU, the FADNet requires on average 0.20 s per image to achieve 41.36 dB PSNR, whereas other methods like the CycleISP [91] need approximately 0.40 s for 40.85 dB PSNR, and the MPRNet [92] takes about 1.00 s for 40.01 dB PSNR. In terms of model size and memory footprint, FADNet comprises approximately 22 million parameters ( $\approx 88$  MB model file) and peaks at  $\sim 3.0$  GB of GPU memory during inference—roughly half the parameter count of the MPRNet (51 M,  $\approx 204$  MB) and 15% fewer parameters than the Restormer [93] (26 M,  $\approx 104$  MB). Computationally, the FADNet performs on the order of 150 GFLOPs per forward pass, compared to  $\sim 300$  GFLOPs for the Restormer and  $\sim 500$  GFLOPs for MPRNet. However, in terms of memory and computational efficiency the FADNet is still orders of magnitude more demanding than traditional non-learning-based methods which require  $\sim 0.2$  GFLOPs and less than 100 KB. The evaluation was conducted on 1200 randomly cropped  $512 \times 512$  patches drawn from the Nam [54] and SIDD [47] benchmarks with noise levels typically quantified by standard deviation estimates ranging from approximately  $\sigma = 5$  to 50 in linearized RAW space.

In addition to noisy data generation for NN training, GAN models can be used to create reference noise signals for noise suppression systems. This led to the development of a three-component noise suppression GAN model. It consists of a generator NN, a noise suppression NN based on a dual-path U-Net [70] architecture, and a discriminator NN [94]. On the SIDD dataset [47], the proposed GAN model achieved a PSNR of 39.29 dB and a SSIM of 0.915, significantly outperforming the traditional BM3D [95] method (25.67 dB PSNR, 0.687 SSIM) and learning-based TridentGAN [96] (38.40 dB PSNR, 0.904 SSIM). On the other

hand, the proposed architecture exhibits significant computational demands, with the denoiser alone comprising approximately 15.6 million parameters and requiring 68.9 GFLOPs per forward pass, rendering it less suitable for deployment on resource-constrained devices. In contrast, the traditional BM3D [95] algorithm operates with minimal memory requirements of only 71 MB for  $512 \times 512$  images and effectively zero learnable parameters, while the TridentGAN [96] achieves comparable denoising performance (38.40 dB PSNR, 0.904 SSIM) with 0.047 s inference time compared to the proposed method's 0.051 s. Furthermore, the model's effectiveness is contingent upon the noise characteristics; its stability may degrade with structured or non-stationary noise patterns, such as low-frequency noise, due to the fixed dilation rates and the reliance on local receptive fields.

Approaches based on direct image reconstruction as well as on denoising with a known noise level are also being developed. Examples of such NN models are RED-Net [97], RBDN [98], MemNet [99], DHDN [100], NNs based on the U-net modifications [101], and others [102–104]. These methods have gained attention for camera noise estimation for the purpose of image source identification.

Table 3 presents the summary of methods discussed in this section.

**Table 3.** Denoising neural network architecture performance comparison.

Architecture	Numerical Results	Training Conditions and Datasets	Notes
DRNE [88]	FFDNet [89]: PSNR: 33.68 dB Average error (noise estimation): up to 0.32 dB	Kodak, McMaster [52], BSD500 [53], synthetic Gaussian noise	16-layer CNN for pixelwise noise variance mapping, signal-dependent noise estimation
FADNet [90]	PSNR: 41.36 dB	Nam [54], SIDD [47], 1200 random $512 \times 512$ patches	Frequency-domain attention mechanism with encoder-decoder, 22 M parameters, ~150 GFLOPs
GAN-based denoiser [94]	PSNR: 39.29 dB, SSIM: 0.915	SIDD [47]	Three-component: generator + dual-path U-Net [70] denoiser + discriminator, 15.6 M parameters, 68.9 GFLOPs

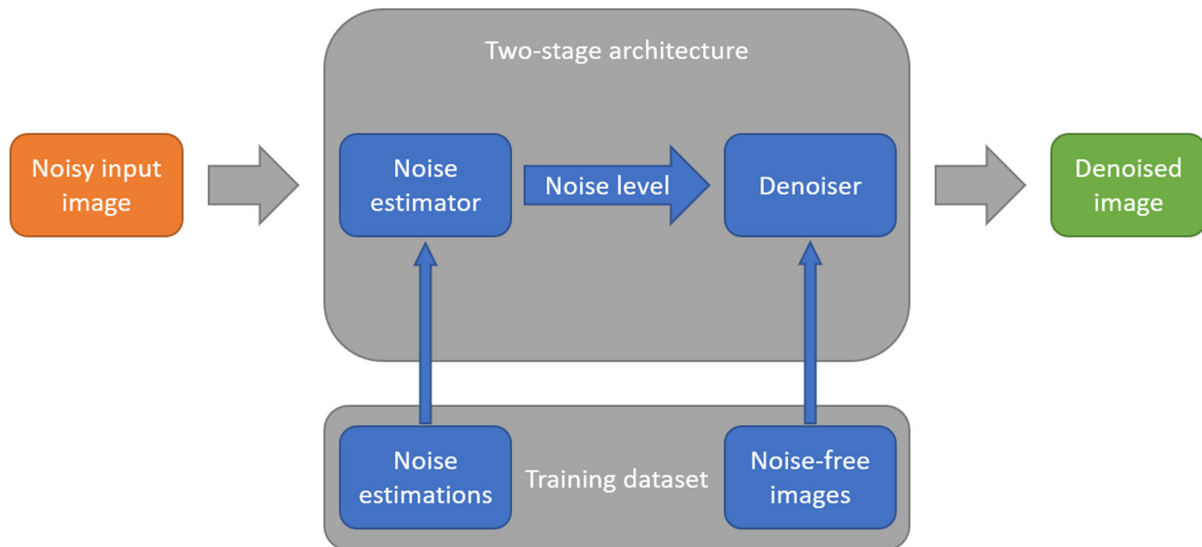
#### 4.3. Noise Estimation for Denoising

Camera and image noise estimation is closely related to denoising. However, NN models for these applications often require mutually exclusive modifications. Therefore, two-component systems are being developed that separate noise estimation and suppression between two NNs. An example of such a system is shown in Figure 11. These systems typically estimate image noise rather than the photosensor noise.

Noise suppression with signal-dependent noise estimation is at the core of the CBDNet model [105]. The CBDNet structure includes two subnetworks: a shallow five-layer fully convolutional network for noise estimation and a modified 16-layer U-Net [70] for non-blind denoising. The model is trained using a combined dataset of synthetic images, generated with a sophisticated noise model, and real-world noisy photographs. The synthetic noise is generated using a Poisson–Gaussian distribution that incorporates in-camera processing pipeline artifacts, including demosaicing, camera response function application, and JPEG compression. This Poisson–Gaussian distribution is approximated as a heteroscedastic Gaussian with variance

$$\sigma^2(L) = L \cdot \sigma_s^2 + \sigma_c^2 \quad (21)$$

where  $L$  is irradiance of the pixel,  $\sigma_s$  is signal-dependent noise variance (photon shot noise), and  $\sigma_c$  is signal-independent noise variance (read noise). During training, the noise level parameters  $\sigma_s$  and  $\sigma_c$  were uniformly sampled from the ranges  $[0, 0.16]$  and  $[0, 0.06]$ , respectively, to cover a wide spectrum of realistic noise intensities. For the JPEG compression scenario, the quality factor was sampled from  $[60, 100]$ , representing moderate to high quality and correspondingly low to moderate compression artifacts.



**Figure 11.** Noise suppression pipeline with two-component structure.

The noise estimation subnetwork is trained with an asymmetric loss function and a total variation regularizer that decreases the probability of the noise underestimation during training. This approach significantly outperforms methods reliant on simplistic AWGN assumptions. On the DND dataset [48], which contains real-world sRGB images with complex, signal-dependent noise, the CBDNet achieved a PSNR of 38.06 dB and a SSIM of 0.9421, surpassing contemporary methods like the MCWNNM [106] (37.38 dB/0.9294) and the TWSC [107] (37.94 dB/0.9403). On the Nam dataset [54], which consists of JPEG-compressed images, a CBDNet variant incorporating JPEG compression achieved a PSNR of 41.31 dB and a SSIM of 0.9784. However, the model's performance is contingent on the fidelity of its noise model to the target real-world noise; performance may degrade on images with noise characteristics not well-represented by the Poisson–Gaussian model, such as structured noise patterns.

Further development of the CBDNet architecture, which itself was influenced by the FFDNet [89], led to the NERNet model [108]. The NERNet retains the two-subnetwork structure for blind denoising but introduces significant architectural innovations to enhance performance. The noise estimation subnetwork utilizes a symmetric dilated convolution block and a pyramid feature fusion block to create a receptive field pyramid, enabling more accurate estimation of spatially variant noise levels. The noise suppression subnetwork, based on the U-Net [70], is enhanced with dense blocks and a novel Dilation Selective (DS) block incorporating an attention mechanism. This DS block adaptively fuses features from parallel convolutional layers, using a mechanism that combines local (Gram matrix-based) and global (global average pooling) attention to weigh the contributions of each branch. For synthetic noise evaluation, the model was trained and tested on standard noise levels from  $\sigma = 15$  to 50. On the SIDD dataset [47], the NERNet achieved a PSNR of 37.97 dB and a SSIM of 0.942, surpassing CBDNet's 33.28 dB and 0.868. On the Nam [54] dataset, it achieved 40.10 dB PSNR, outperforming CBDNet's 39.01 dB. Furthermore, on synthetic

noise benchmarks (BSD68 with  $\sigma = 50$ ), the NERNet reached 28.12 dB PSNR, exceeding the FFDNet [89] (27.96 dB) and the DnCNN [75] (27.92 dB).

The two-network architecture has advanced significantly with the development of the FBI-Denoiser network [109]. The FBI-Denoiser integrates two specialized neural modules: the PGE-Net, a compact convolutional regressor designed for the rapid estimation of Poisson–Gaussian noise parameters, and the FBI-Net, an efficient blind-spot network based on a modified U-Net architecture [70] that is trained exclusively on single noisy images. A critical preprocessing step involves applying the Generalized Anscombe Transformation (GAT) [110] to the input noisy image using the parameters estimated by the PGE-Net, stabilizing the noise to approximate Gaussian distribution with unit variance. This preprocessing enables the subsequent FBI-Net to perform denoising effectively in the transformed domain. The PGE-Net was trained and evaluated on images corrupted with a range of Poisson–Gaussian noise levels, with parameters  $(\alpha, \sigma)$  covering very low to moderate noise intensities  $((0.01, 0.0002), (0.01, 0.001), (0.05, 0.0002), \text{ and } (0.05, 0.002))$ . The PGE-Net achieved significant speedup in noise parameter estimation compared to conventional methods, reducing the estimation time from seconds (3.123 s for Foi’s method [20] and 1.084 s for the Liu method [111] on CPU) to 0.002 s per image on the GPU when tested on BSD68 [112] (grayscale) and FiveK [113] (raw–RGB) datasets across multiple Poisson–Gaussian noise levels. This represents an approximately  $1560\times$  speedup compared to Foi’s method and a  $540\times$  speedup compared to the Liu method. The overall FBI-Denoiser pipeline—which combines fast estimation (PGE-Net) with the efficient FBI-Net blind-spot network—was evaluated on  $512 \times 512$  real-world noisy images from the FMD [114] (microscopy), the SIDD [47] (raw–RGB), and the DND [48] (sRGB) datasets. On the DND benchmark, the FBI-Denoiser achieved PSNR of 48.02 dB and SSIM of 0.9797, demonstrating state-of-the-art performance among methods trained only on single noisy images. Under these conditions, total end-to-end inference time improves 9.5-fold, from 2.00 s per image for the BP-AIDE to 0.21 s per image for the FBI-Denoiser, while maintaining or surpassing denoising performance. Additionally, the FBI-Net uses only 340,000 parameters and 2512 MB GPU memory compared to competing blind-spot networks like the D-BSN [115] (6,612,000 parameters, 4231 MB) and the FC-AIDE [116] (754,000 parameters, 2581 MB).

The two-network models are not limited to pixel-by-pixel noise mapping. In [117], a method for global statistical noise estimation based on the PCANet [118] and the ResNet101 [119] NNs was proposed. The PCANet is used as a fragmenter defining image segments, and the ResNet101 serves as a classifier for these segments. For each segment, the  $\sigma$  parameter is estimated in the chi-square distribution. This method was evaluated on 100 randomly selected images from the BSD300 dataset [51], with AGWN standard deviations levels ranging from  $\sigma = 5$  to 40. The proposed hybrid network achieved a mean estimation error of noise standard deviation as low as 0.22, outperforming traditional methods such as [28] (0.28) and [37] (0.26). The network architecture demonstrated flat patches selecting accuracy reaching approximately 92% after training on a dataset of 1,165,600 patches.

Dual-stage denoising architectures inherently balance increased computational demands against enhanced output quality. By decoupling noise estimation from noise suppression into two dedicated neural networks, these systems incur substantial overhead compared to single-stage designs. Empirical studies indicate that isolating the noise-parameter estimation and suppression functions approximately doubles the required floating-point operations: for example, the dual-CNN model DudeNet [120] demands roughly 1.87 GFLOPs vs. 0.94 GFLOPs for the single-stage DnCNN, while delivering comparable denoising efficacy. Similarly, the TSP-RDANet [121] framework employs five residual dynamic attention modules (RDAMs) in its noise-estimation stage and five high-dimensional RDAMs in its

suppression stage to achieve a measured compromise between performance and complexity. Memory requirements also escalate: dual-network systems typically possess 25–30% more parameters than their single-stage counterparts [120]. However, advanced network designs can mitigate this burden. For instance, dual-network denoisers like the DCANet [122] and the DCBDNet [123] have been shown to address the issue of increased computational costs and memory consumption growth. The DCANet incorporates a spatial–channel attention module, while the DCBDNet relies on inter-layer skip-connections. While performing on par with state-of-the-art models (e.g., the DRUNet [124]), these models maintain parameter counts around 1 M—far below 32.6 M in the DRUNet architecture. Their efficiency is further reflected in low computational costs, as of 24.38 GFLOPs for a  $256 \times 256$  grayscale input, and fast inference of 0.183 s on a  $1024 \times 1024$  grayscale image.

Several approaches have been suggested to address explicit noise estimation without obligatory denoising stage. A system based on the DRNE [85] was proposed specifically for noise classification and parameter estimation [125]. For this application, the NN training was augmented with EXIF metadata, categorized into minimal (camera gain, exposure time, sensor temperature) and full metadata sets (additionally including parameters such as dark signal figure of merit, full well capacity, pixel clock rate, sense node gain, sense node reset factor, sensor pixel size, sensor type, and thermal white noise). The noise levels were evaluated across a controlled intensity spectrum, with synthetic noise simulating realistic operating conditions, including low-light and high-gain scenarios. Experimental validation across six datasets demonstrated the method’s effectiveness, with the full metadata model variant achieving RMS errors as low as 0.09 DN for photon-shot noise, 0.35 DN for dark-current shot noise, and 0.47 DN for readout noise on synthetic data, compared to 0.75–1.05 DN RMS errors for the minimal-metadata and metadata-free variants. On real-world noise from a Sony ICX285 CCD and an EV76C661 CMOS sensor, a full metadata model yielded bias and standard deviation below 0.15 DN and 0.45 DN, respectively, for all noise sources, outperforming PGE-Net [109] (bias up to 1.74 DN, Std up to 3.02 DN) and principal component analysis [39] (total-noise RMS of 1.07 DN on synthetic data). In downstream denoising, a full metadata model combined with BM3D [95] achieved PSNRs up to 43.74 dB and SSIM of 0.9839 on real-world scenes, surpassing DRNE [88] + BM3D [95] (43.01 dB/0.9803) and Blind2Unblind [83] (44.10 dB/0.9616) under identical testing conditions (realistic environments, low light, high gain). However, the method’s effectiveness is contingent upon the availability and accuracy of comprehensive camera metadata; performance degrades with consumer-grade cameras that provide limited or inaccurate parameters. Additionally, the model exhibits limitations in scenarios with extreme under- or overexposure and with sensors featuring full well capacities below 24k electrons, due to underrepresented training data. The computational cost remains low at 1.3 ms per inference on a GPU, but the memory requirement for storing extensive metadata may impose practical constraints in resource-constrained environments.

Finally, a conditional denoising transformer (Condformer) was introduced alongside a Locally Noise Prior Estimation (LoNPE) framework for explicit noise-prior embedding [126]. LoNPE estimates Poisson–Gaussian noise parameters from a single raw noisy image. Across the Urban100 dataset [127] with different synthetic Poisson–Gaussian noise levels ( $\alpha, \sigma$ )—ranging from medium (0.05, 0.02) to high (0.2, 0.1)—the LoNPE achieved RMS error of 0.020–0.023 and coefficient of variation below 0.031, outperforming the Makitalo–Foi estimator [128] (RMS 0.041–0.089) with a  $300\times$  speedup (0.17 s vs. 56 s per  $512 \times 512$  image). The LoNPE can be simplified and accelerated. A lightweight LoNPE variant runs in 0.01 s with RMS of 0.020–0.025. The Condformer embeds the estimated noise prior into channel-wise self-attention in a Restormer-style [93] U-shaped encoder–decoder with a conditional self-attention module. This results in PSNR and SSIM value

increases of 0.34 dB and 0.0018 respectively over state-of-the-art methods (Restormer [93], Uformer [129], MambaR [130]) on synthetic and realistic benchmarks (Kodak, BSD68 [112], Urban100 [127], SIDD [47]). However, the Condformer’s eight-layer latent module and 27 M parameters incur elevated FLOPs (565 G) and GPU memory (3.8 GB) per  $512 \times 512$  image, leading to 0.37 s inference time—substantially higher than CNN-based denoisers—making it impractical for resource-constrained or real-time applications and scenarios with limited GPU memory.

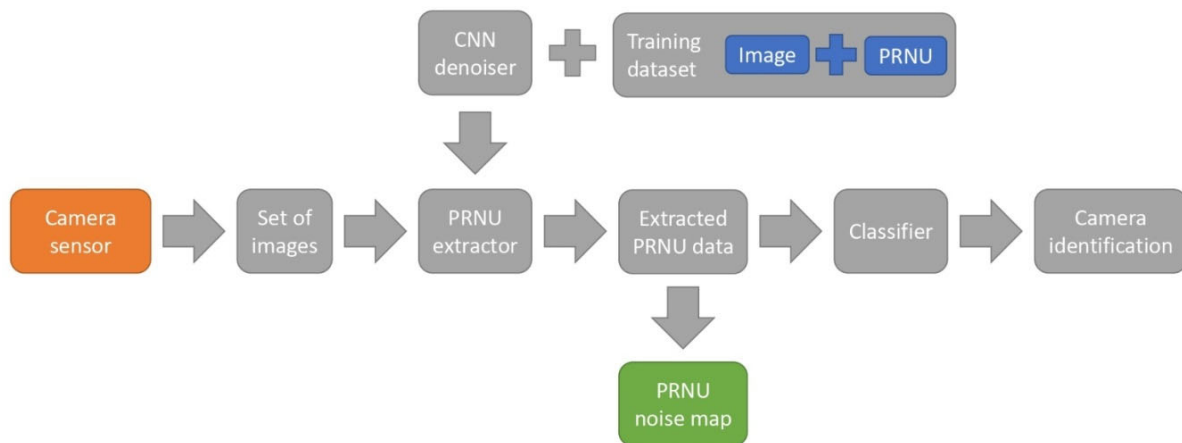
Table 4 presents the summary of methods discussed in this section.

**Table 4.** Neural network architectures for noise estimation and removal.

Architecture	Numerical Results	Training Conditions and Datasets	Notes
CBDNet [105]	PSNR: up to 41.31 dB, SSIM: 0.9421	DND [48], Nam [54] dataset, synthetic + real-world	Two-subnetwork: 5-layer noise estimation + 16-layer U-Net [70] denoising
NERNNet [108]	PSNR: up to 40.10 dB, SSIM: 0.942	SIDD [47], Nam [54], BSD68 [112]	Enhanced CBDNet [105] with pyramid feature fusion and attention mechanisms
FBI-Denoiser [109]	PSNR: up to 48.02 dB, SSIM: up to 0.9797 1560× speedup in estimation	BSD68 [112], FiveK [113], FMD [114], SIDD [47], DND [48]	GAT [110] preprocessing, 0.21 s inference time, 340 K parameters
PCANet + ResNet101 [117]	Mean estimation error: 0.22, patch selection accuracy: 92%	100 images of BSD300 [51], 1M+ training patches	Global statistical noise estimation with chi-square distribution
Metadata-enhanced model [125]	RMS errors: from 0.09 to 0.47 DN, PSNR: up to 43.74 dB	Sony ICX285 CCD, EV76C661 CMOS, synthetic data	DRNE-based [88] with EXIF metadata integration, 1.3 ms inference time
Condformer + LoNPE [126]	RMS error: up to 0.023, 300× speedup, PSNR improvement: 0.34 dB	Urban100 [127] with synthetic Poisson-Gaussian noise	Transformer-based with noise prior embedding, 27 M parameters, 565 GFLOPs

#### 4.4. Source Camera Identification

Digital forensics is an important application for NN-based image processing. In particular, models for determining whether an image was taken on a specific digital camera are being actively developed [131–147]. These methods are usually based on the approximated PRNU( $i, j$ ) values extracted from images with the subsequent classification by camera manufacturers, models or by specific devices [2] (Figure 12). The PRNU extraction approaches have demonstrated high reliability for source camera attribution under mild image processing operations such as moderate JPEG compression and Gaussian blurring, owing to its origin in intrinsic sensor nonuniformities [148,149]. However, PRNU extraction can fail if the high-frequency spatial spectrum is deliberately perturbed by strong denoising filters, geometric desynchronization (e.g., rotation, cropping, scaling), or adversarial noise injection that disrupts spatial correlation. To overcome these limitations, robust fingerprinting schemes can incorporate fusing of the PRNU with complementary low-frequency device-specific fingerprints via deep learning architectures [150]. Nonetheless, PRNU fusing can itself distort the camera noise estimations, deeming such approach unsuitable for noise extraction.



**Figure 12.** PRNU extraction pipeline for source camera identification.

NN methods originally developed for extraction of  $PRNU(i,j)$  or other types of camera fingerprints are of particular interest. In this case, noise suppression isolates the weak but stable high-frequency PRNU component from other low-frequency noise [151]. Therefore, the extracted noise fingerprint can then be used to identify the image source.

In [152], a review of successful applications of CNNs and other machine learning tools for source camera identification is presented.  $PRNU(i,j)$  extraction can be performed using CNNs as well as other methods [153]. One approach to  $PRNU(i,j)$  extraction is the segmentation and suppression of low-frequency noise corresponding to the scene content. Both pre-trained NNs such as the VGG [154] and the specially designed NNs such as the RemNet [155] and the MSFFN [156] can be used to segment and suppress the scene content before the PRNU extraction. Such strategy is used in the Constrained-Net model adapted for video [157], which employs an extended constrained convolutional layer to process color inputs and suppress scene content by learning pixel value prediction errors. The model was evaluated on the VISION dataset [158] with 1539 videos from 28 different camera devices, including 13 instances of the same brand and model to facilitate device-level identification. Videos represented three distinct scenarios (flat, indoor, outdoor) and three versions: native, WhatsApp-compressed, and YouTube-compressed. Under these conditions, the model achieved a peak overall video classification accuracy of 66.5% when aggregating frame classifications via majority vote. Performance varied significantly by scenario, with accuracy rates of 89.1% for the homogeneous ‘flat’ scenario (e.g., walls, blue skies,), 53.7% for ‘indoor’, and 55.2% for ‘outdoor’, indicating a strong dependence on content homogeneity. When evaluating the sensitivity to compression on flat-content videos, the method achieved accuracies of 89.7% on native videos, 93.1% on WhatsApp versions, and 84.5% on YouTube versions. A comparative study demonstrated the critical role of the constrained layer, as its absence significantly decreases the performance. However, this approach necessitates extensive computational resources and training data, requiring over 100,000 frames for training, and its performance degrades considerably with inhomogeneous, dynamic scene content where scene-dependent features dominate. Furthermore, fixed input size requirements and the lack of a patch-based strategy to focus on homogeneous regions within frames limit the flexibility of the approach.

Another group of  $PRNU(i,j)$  extraction methods relies on CNNs to extract features from images. The similarity of the tools suggests that NN denoising can be applied, to some extent, to  $PRNU(i,j)$  extraction. The comparative study [84] shows that common denoising CNNs such as DnCNN [75], ADNet [85], DANet [86], and FFDNet [89] can be adapted for  $PRNU(i,j)$  extraction. For this purpose, the NN was trained using pairs “image-PRNU” with correlation loss function. The study was conducted on the DID [46] dataset using

40 cameras of 11 different models, with images cropped into  $128 \times 128$  and  $64 \times 64$  patches to evaluate the performance under low-resolution conditions. Among the CNN-based methods, the FFDNet achieved the highest average peak-to-correlation energy (PCE) value of 16.5 on  $128 \times 128$  patches, followed by the ADNet (15.1), the DnCNN (13.6), and the DANet (9.6). These results were comparable or superior to non-data-driven methods such as the BM3D [95] (15.8), the DWT [159] (13.5), and the DTCWT [160] (15.3). However, training universal CNN models requires large datasets with at least several thousand images and significant computational resources. For instance, the DANet's adversarial training is prohibitively time-consuming, and GPU acceleration is often necessary for practical use—the FFDNet required 0.24 s per  $1024 \times 1024$  image on a GPU and 2.05 s on a CPU.

The inclusion of a separate noise-estimation stage in a convolutional neural network is not necessary to achieve effective PRNU extraction. The densely connected hierarchical denoising network DHDN provides an efficient alternative [100]. Its modified U-Net backbone [70], augmented by dense connectivity and residual learning, suppresses scene content and isolates sensor noise. The DHDN was evaluated on two datasets: the DID [46] (74 cameras, 100 images per device) and the DSD [161] (90 devices, 100 images per device) datasets. Cropped images with resolutions of  $128 \times 128$ ,  $256 \times 256$ , and  $512 \times 512$  pixels were used. Performance was measured using kappa statistics and compared with three other methods: the BM3D [95], wavelet denoising [162], and the xDnCNN [163] model. Across the used benchmarks, the DHDN shows improved kappa scores by at least 0.0473 on the DID dataset and 0.0073 on the DSD dataset for all image sizes. The DHDN also demonstrated resilience to JPEG compression at quality factors of 60, 70, 80, and 90. The DHDN, as well as other learning-based approaches, requires substantial computational and memory resources both for training and deployment. Additionally, successful training requires at least 40 images per device used, which rapidly outgrows the requirements for traditional approaches.

Another possible approach to improve the efficiency of camera identification and PRNU extraction is the addition of a residual noise extractor [143]. Specifically, the original image is first denoised using a U-Net [70], then denoised output is subtracted from the original to obtain the residual noise [164]. A ResNet-based convolutional network [67] is then trained on this residual noise to isolate the deterministic PRNU component from a residual noise. The model is able to extract PRNU fingerprints and classify the source camera with 92.41% accuracy on 2194 patches of size  $256 \times 256$  from ten devices in the VISION dataset [158].

Analysis of convolutional neural network training indicates that learning-based PRNU methods impose high memory and computational demands, often requiring tens of gigabytes of GPU memory and hours to days of training on high-end hardware to converge [165]. Likewise, extraction of reference PRNU patterns via maximum-likelihood estimation mandates acquisition and processing of at least 50 flat-field images per device—representing a substantial dataset collection and preprocessing burden that scales linearly with the number of cameras [166]. Furthermore, studies of PRNU robustness under aggressive encoding have shown that heavy image and video compression, such as that employed by social media platforms, reduces PRNU correlation values by over 0.3 relative to the uncompressed material, frequently dropping below reliable detection thresholds and thus degrading identification performance [150].

Table 5 presents the summary of methods discussed in this section.

**Table 5.** Source camera identification neural network method comparison.

Architecture	Numerical Results	Training Conditions and Datasets	Notes
Constrained-Net [157]	Video classification accuracy: 66.5% overall, 89.1% flat scenes	VISION [158], 1539 videos, 28 camera devices, >100 K training frames	Extended constrained convolutional layer for video PRNU extraction
CNN adaptations [84]	PCE: up to 16.5 (FFDNet [89])	DID [46], 40 cameras, 11 models, 128 × 128 and 64 × 64 patches	Adaptation of denoising CNNs for PRNU extraction with correlation loss
DHDN [100]	Kappa improvement: at least 0.0473	DID [46] (74 cameras), DSD [161] (90 devices), 100 images per device	Modified U-Net [70] with dense connectivity for sensor noise isolation
ResNet-based extractor [143]	Classification accuracy: 92.41%	VISION [158], 2194 patches 256 × 256, 10 devices	U-Net [70] denoising with ResNet [67] residual noise extraction for PRNU fingerprints

#### 4.5. Other Applications

Apart from noise suppression, identification, characterization, and camera selection, NNs have been used for other applications in noise analysis and image processing. One such application is the NN classification of noise type in images. For instance, a study utilizing a custom CNN architecture achieved a training accuracy of 99.87% and a validation accuracy of 99.92% in classifying facial images corrupted with Gaussian, Poisson, and salt and pepper noise at multiple noise factors (0.05, 0.07, and 0.1), using a dataset of 34,034 preprocessed and augmented images resized to 48×48 pixels [167]. The CNN classifier served as the first stage in an automated pipeline, determining which specialized U-Net denoiser (each trained on a specific noise type) to activate for the subsequent denoising step. This two-stage system demonstrates the utility of noise classification for enabling targeted, noise-specific restoration strategies. Another approach combined deep wavelet scattering transformations with a support vector machine to classify five noise types (gaussian, lognormal, Rayleigh, salt and pepper, speckle). This method, evaluated on a dataset of 1,100 images resized to 28 × 28, achieved a classification accuracy of 91.30% [168]. The noise was applied at a consistent intensity level across all types to ensure a fair comparison; specifically, Gaussian noise was added with a mean of 0 and a variance of 0.01, salt and pepper noise with a noise density of 0.05, speckle noise with a variance of 0.05, while lognormal and Rayleigh noise parameters were scaled to produce a similar visual degradation and signal-to-noise ratio. In [169], a combination of a traditional filtration algorithm and a NN was proposed for noise estimation and classification. This method applies discrete wavelet transformation for high frequency band extraction followed by the noise level estimation. However, this approach does not provide correspondence of digital estimates to physically reasonable noise parameters. In addition, reported precision and recall of 1.0 raises suspicions of overfitting the model; additional analysis of model performance was not reported.

Based on the PRNU extraction methods and JPEG error level analysis (ELA), methods for artificial image detection have been developed [170]. Testing was conducted using a balanced dataset of 918 images (459 AI-generated, 459 real), extracting PRNU and ELA features from central 512 × 512 crops for input to a custom CNN. The PRNU-based model achieved 0.95 accuracy and 0.95 F1-score, while the ELA-based model reached 0.98 accuracy and 0.98 F1-score on the validation set. Similar results were achieved with the Siamese NN for deepfake image detection [171]. Testing was conducted on five benchmark deepfake datasets (FF++, Celeb-DF, DFD, DeeperForensics, DFDC) using a Siamese two-stream

network with the Inception-v3 backbone; video-level predictions were aggregated over 25 frames. The proposed SiamNet achieved frame-level accuracies of 98.7% on FF++, 94.2% on Celeb-DF, 92.1% on DFD, 99.8% on DeeperForensics, and 86.9% on DFDC, and video-level accuracies of 99.9% on FF++, 98.3% on Celeb-DF, 96.08% on DFD, 100% on DeeperForensics, and 89.2% on DFDC.

## 5. Discussion

Current solutions such as the EMVA 1288 standard provide accurate estimation of noise parameters, but they are complex and computationally intensive. Qualitative estimation methods based on automatic scene segmentation [13,24–26] use only 2–4 images, but require specialized setups. Simplified statistical methods for noise estimation and suppression using Gaussian [14] or Poisson [17] models, or their combinations [15,16], work using only a single image, but are not as reliable and depend on specific shooting conditions and camera parameters.

Compared to conventional approaches NNs offer several advantages. For example, they are able to account for complex dependencies between noise and signal, thereby greatly increasing the flexibility and versatility of NN models. The main methods and their features are shown in Table 6.

**Table 6.** Features and description of main neural network-based methods.

Architecture	Features	Description
<b>NOISE SUPPRESSION</b>		
DnCNN [75]	Unsupervised noise suppression	Residual convolutional neural network that uses unsupervised training to suppress noise in images
DRNE [88]	Pixel-based noise mapping	Network for noise estimation capable of utilizing metadata and additional information
CBDNet [105], NERNet [108], FBI-Denoiser [109], DCANet [122], DCBDNet [123]	Two-network structure: noise estimator and noise suppressor	Separation of noise estimation and suppression allows them to be improved separately
Other neural networks for noise suppression [94,96–104]	Convolutional (U-Net based, etc.) or generative-adversarial networks	Can be used for approximated PRNU extraction
<b>SOURCE CAMERA IDENTIFICATION</b>		
U-Net modifications [70]	Feature extraction	U-Net’s high spatial resolution feature extraction capability is adapted for approximated PRNU extraction
DnCNN [75], FFDNet [89], ADNet [85], DANet [86]	Adapted noise suppression networks	Can be adapted for approximated PRNU extraction
Constrained-Net [157]	Works with video data	Requires large (over 100 thousand frames) amount of video data for training
<b>SYNTHETIC DATASET GENERATION</b>		
NoiseFlow [56], CANGAN [57], C2N [59]	Noise synthesizers	Models based on generative-adversarial networks or normalizing flows
Noise2NoiseFlow [60], DCD-Net [81]	Noise suppression with a noise synthesizer	Improving noise suppression by utilizing a noise synthesizer or by using only noisy data for training

In recent years, CNN-based autoencoders, transformers, and generative models [55–61,70,73–75,81,84–90,92–94,96–109,115–117,120–126,129–147,154–157,164,167–171] have demonstrated promising results for noise estimation and suppression. CNNs gen-

erally demonstrate high accuracy of noise feature extraction with unsupervised training. Trained NNs often require only a single image to accurately estimate individual noise components. In this aspect, NN-based approaches significantly outperform conventional methods, which often require hundreds of images [8]. NN models are also several times faster than conventional methods based on automatic scene segmentation [13,24–26] and provide more accurate estimations than single-image methods [14–17].

The flexibility of NN architectures and approaches facilitates the adaptation of NN-based methods for a wide variety of applications in camera and image noise processing [1]. For example, NN-based noise suppression methods proved useful in both direct estimation of several noise characteristics [70,75,84–86,88–90,94,96–105,108,109,117,122,126] and in mapping the approximated pixel spatial heterogeneity for source camera identification [70,75,84,86,88,90,125]. GAN models show promise for noise modeling and generation [56–61], suppression [81,94], and extraction [81]. Last, but not least, some transformer architectures show exceptional ability for noise parameter evaluation [90,126].

The ability to generalize and integrate additional context data is a significant advantage of NNs over traditional approaches. For example, integration of EXIF metadata and information about shooting conditions into the training process improved the quality of noise suppression and camera identification [125].

However, for all their advantages, NN-based methods also have their limitations that manifest across multiple dimensions of computational efficiency and practical deployment.

#### **Training Dataset and Development Constraints**

NN training fundamentally limits the development speed due to the requirements for high-quality, domain-specific datasets that are notoriously difficult to create. The acquisition of hundreds or thousands of real-world images for training and testing demands significant time and labor investment, exemplified by datasets like SIDD [47] requiring ~30,000 raw-RGB image pairs from multiple camera models under diverse conditions. The computational burden extends beyond data collection: training modern architectures like the Condformer [126] (27M parameters, 565 GFLOPs) or CANGAN [57] frameworks can require days to weeks on high-end GPU clusters, representing substantial infrastructure costs that traditional methods avoid entirely.

#### **Computational Efficiency Trade-offs**

The computational landscape reveals stark disparities between neural network-based and traditional approaches. While traditional methods like the BM3D [95] maintain consistent, moderate resource requirements (71 MB memory for  $512 \times 512$  images, ~0.2 GFLOPs), learning-based methods exhibit dramatic variability. Efficient architectures like the NoiseFlow [56] achieve competitive performance with fewer than 2500 parameters, while resource-intensive models like the FADNet [90] demand 22 M parameters and ~150 GFLOPs per forward pass. This represents a three-order-of-magnitude difference in parameter count within the neural network domain alone.

Dual-network architectures embody the performance–efficiency trade-off: methods like CBDNet [105] and NERNet [108] that separate noise estimation from suppression approximately double computational overhead compared to single-stage designs (DudeNet [120]: 1.87 GFLOPs vs. DnCNN [75]: 0.94 GFLOPs), while delivering only marginal PSNR improvements (typically 0.5–2 dB). However, specialized designs like the FBI-Denoiser [109] demonstrate that architectural innovation can achieve substantial time-efficiency gains.

#### **Memory and Deployment Scalability**

Memory requirements present another critical constraint. Contemporary neural denoisers typically consume 2–4 GB of GPU memory during inference, compared to traditional methods that require less than 100 MB. This disparity becomes prohibitive for

deployment on resource-constrained devices or real-time applications. The asymmetric resource utilization—computationally intensive training followed by relatively lightweight deployment—creates infrastructure challenges distinct from traditional methods' consistent moderate load throughout their pipeline.

#### **Noise Separation and Interpretability Challenges**

Qualitative separation of scene-dependent noise from photosensor noise remains a fundamental limitation. For valid comparison with traditional camera characterization methods, NN-based pipelines must distinguish between these types of noise within their processing workflows. Currently, this separation is typically achieved only partially for individual noise components (e.g., PRNU extraction), but comprehensive simultaneous separation of all four EMVA 1288 [8] noise types remain unresolved. The interpretability issue compounds this challenge: features extracted from architectures adapted for noise processing cannot be reliably classified as photosensor vs. image noise, undermining their utility for camera characterization applications.

#### **Benchmarking and Evaluation Gaps**

Evaluation and comparison of traditional and learning-based methods are complicated by fundamental differences in performance. This primarily applies to estimates of computational complexity, memory usage, and the efficiency of the methods used in relation to human labor when using different methods. Traditional methods offer deterministic, physics-based parameters estimation suitable for standardized camera characterization, but they tend to be inefficient in terms of human labor. Learning-based methods excel at statistical pattern recognition optimized for specific datasets, offering labor-efficient usage after deployment at the cost of computational complexity and strict requirement for training datasets. This methodological divergence necessitates development of hybrid evaluation frameworks that can assess both accuracy and computational efficiency across different paradigms.

#### **Research Priorities and Future Directions**

Considering these limitations, several critical research priorities emerge:

- **Physics-informed architecture design:** Developing neural networks that explicitly model the four-component noise structure defined by EMVA 1288 [8], enabling direct separation of photosensor noise from scene artifacts while maintaining computational efficiency.
- **Standardized synthetic dataset creation:** Establishing large-scale, validated synthetic datasets that accurately model complex noise pipelines of modern smartphones and cameras, reducing dependency on labor-intensive real-world data collection.
- **Efficiency-optimized architectures:** Investigating architectural innovations that achieve the accuracy of dual-network models while approaching single-network computational costs, particularly for real-time and mobile deployment scenarios.
- **Hybrid evaluation frameworks:** Creating benchmarking protocols that quantitatively compare learning-based and traditional methods across multiple dimensions—accuracy, computational cost, memory efficiency, and physical parameter interpretability.
- **Interpretable feature extraction:** Developing methods to ensure that neural network outputs correspond to physically meaningful noise parameters, enabling their use in camera characterization and forensic applications where interpretability is paramount.
- **Adaptive computational scaling:** Designing networks capable of trading accuracy for efficiency based on deployment constraints, allowing the same architecture to serve both high-accuracy offline applications and resource-constrained real-time scenarios.

## 6. Conclusions

Different approaches to neural network-based digital camera and image noise estimation, suppression, classification, extraction, etc., are being actively developed. These methods can be applied for image quality enhancement, digital forensics, device classification, comparison, identification, etc. Modern neural network architectures can account for both the physical properties of noise and the contextual data, such as shooting parameters. This additional information increases the accuracy and reliability of noise estimation and classification.

The convolutional neural networks discussed in this paper demonstrate high accuracy extraction of real-world noise features, while generative models demonstrate efficient generation of synthetic noise. Special attention needs to be paid to the possibility of using neural networks for extraction, interpretation, and separation of noise components arising from image or photosensor noise.

Neural network-based methods for image processing and camera noise estimation demonstrate great potential for further improvement and practical application in different areas, such as image restoration, signal-to-noise ratio enhancement, forensics, and image source identification.

**Author Contributions:** Conceptualization, P.A.C., V.G.R., R.S.S. and E.Y.Z.; methodology, P.A.C., A.V.S., A.V.K. and R.S.S.; software, A.A.V. and A.V.K.; validation, A.A.V. and A.V.K.; formal analysis, D.A.R., R.S.S., E.Y.Z. and V.A.N.; investigation, A.A.V., P.A.C. and A.V.K.; resources, V.G.R.; data curation, A.V.S., E.K.P. and V.A.N.; writing—original draft preparation, A.A.V., A.V.K., P.A.C. and D.A.R.; writing—review and editing, A.A.V., A.V.K. and P.A.C.; visualization, A.A.V., A.V.K. and D.A.R.; supervision, V.G.R.; project administration, R.S.S., A.V.S. and E.K.P.; funding acquisition, V.G.R. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by grant 24-19-00898 from the Russian Science Foundation (RSF).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Data sharing is not applicable.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

AIRM	Adaptive Instance Residual Module
AWGN	Additive White Gaussian Noise
BM3D	Block-Matching and 3D Filtering
CNN	Convolutional Neural Network
CPU	Central Processing Unit
DN	Digital Number
DS	Dilation Selective (block)
DSNU	Dark Signal Non-Uniformity
DWT	Discrete Wavelet Transformation
EMVA	European Machine Vision Association
GAN	Generative Adversarial Network
GAT	Generalized Anscombe Transformation
GPU	Graphics Processing Unit
KL	Kullback–Leibler (Divergence)
NLF	Noise Level Functions
NLL	Negative Log-Likelihood

NN	Neural Network
PCE	Peak-to-Correlation Energy
RDAM	Residual Dynamic Attention Modules
ReLU	Rectified Linear Unit
RMS	Root Mean Square (error)
PRNU	Photo Response Non-Uniformity
PSNR	Peak Signal-to-Noise Ratio
SSIM	Structural Similarity Index Measure
Std	Standard deviation

## References

- Xu, P.; Wang, J.; Jiang, Y.; Gong, X. Applications of Artificial Intelligence and Machine Learning in Image Processing. *Front. Mater.* **2024**, *11*, 1431179. [[CrossRef](#)]
- Bernacki, J.; Scherer, R. Algorithms and Methods for Individual Source Camera Identification: A Survey. *Sensors* **2025**, *25*, 3027. [[CrossRef](#)]
- Nematollahi, M.A.; Vorakulpipat, C.; Rosales, H.G. *Digital Watermarking*; Springer: Berlin/Heidelberg, Germany, 2016; ISBN 9789811020940.
- Sadia, R.T.; Chen, J.; Zhang, J. CT Image Denoising Methods for Image Quality Improvement and Radiation Dose Reduction. *J. Appl. Clin. Med. Phys.* **2024**, *25*, e14270. [[CrossRef](#)]
- Hussein, T.D.H.; Jihad, K.H.; Omar, H.K. A study on image noise and various image denoising techniques. *Res. J. Anal. Invent.* **2021**, *2*, 27–44. [[CrossRef](#)]
- Fan, L.; Zhang, F.; Fan, H.; Zhang, C. Brief Review of Image Denoising Techniques. *Vis. Comput. Ind. Biomed. Art* **2019**, *2*, 7. [[CrossRef](#)]
- Li, Y.; Liu, C.; You, X.; Liu, J. A Single-Image Noise Estimation Algorithm Based on Pixel-Level Low-Rank Low-Texture Patch and Principal Component Analysis. *Sensors* **2022**, *22*, 8899. [[CrossRef](#)]
- European Machine Vision Association EMVA Standard 1288, Standard for Characterization of Image Sensors and Cameras. Available online: <https://www.emva.org/standards-technology/emva-1288/> (accessed on 14 September 2025).
- Maitre, H. *From Photon to Pixel*; John Wiley & Sons: Hoboken, NJ, USA, 2017; ISBN 9781119402466.
- Kozlov, A.V.; Nikitin, N.V.; Rodin, V.G.; Cheremkhin, P.A. Improving the Reliability of Digital Camera Identification by Optimizing the Algorithm for Comparing Noise Signatures. *Meas. Tech.* **2024**, *66*, 923–934. [[CrossRef](#)]
- Nakamoto, K.; Hotaka, H. Efficient and Accurate Conversion-Gain Estimation of a Photon-Counting Image Sensor Based on the Maximum Likelihood Estimation. *Opt. Express* **2022**, *30*, 37493. [[CrossRef](#)] [[PubMed](#)]
- Widrow, B.; Kollár, I. *Quantization Noise*; Cambridge University Press: Cambridge, UK, 2008; ISBN 9781139472845.
- Kozlov, A.V.; Rodin, V.G.; Starikov, R.S.; Evtikhiev, N.N.; Cheremkhin, P.A. A Family of Methods Based on Automatic Segmentation for Estimating Digital Camera Noise: A Review. *IEEE Sens. J.* **2024**, *24*, 17353–17365. [[CrossRef](#)]
- Zheng, L.; Jin, G.; Xu, W.; Qu, H.; Wu, Y. Noise Model of a Multispectral TDI CCD Imaging System and Its Parameter Estimation of Piecewise Weighted Least Square Fitting. *IEEE Sens. J.* **2017**, *17*, 3656–3668. [[CrossRef](#)]
- Jeong, B.G.; Kim, B.C.; Moon, Y.H.; Eom, I.K. Simplified Noise Model Parameter Estimation for Signal-Dependent Noise. *Signal Process.* **2014**, *96*, 266–273. [[CrossRef](#)]
- Zhang, Y.; Wang, G.; Xu, J. Parameter Estimation of Signal-Dependent Random Noise in CMOS/CCD Image Sensor Based on Numerical Characteristic of Mixed Poisson Noise Samples. *Sensors* **2018**, *18*, 2276. [[CrossRef](#)]
- Ta, C.-K.; Aich, A.; Gupta, A.; Roy-Chowdhury, A.K. *Poisson2Sparse: Self-Supervised Poisson Denoising from a Single Image*; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2022; pp. 557–567. [[CrossRef](#)]
- Hai Thai, T.; Retraint, F.; Cogramme, R. Generalized Signal-Dependent Noise Model and Parameter Estimation for Natural Images. *Signal Process.* **2015**, *114*, 164–170. [[CrossRef](#)]
- Li, J.; Wu, Y.; Zhang, Y.; Zhao, J.; Si, Y. Parameter Estimation of Poisson–Gaussian Signal-Dependent Noise from Single Image of CMOS/CCD Image Sensor Using Local Binary Cyclic Jumping. *Sensors* **2021**, *21*, 8330. [[CrossRef](#)]
- Foi, A.; Trimeche, M.; Katkovnik, V.; Egiazarian, K. Practical Poissonian-Gaussian Noise Modeling and Fitting for Single-Image Raw-Data. *IEEE Trans. Image Process.* **2008**, *17*, 1737–1754. [[CrossRef](#)]
- Liu, C.; Szeliski, R.; Kang, S.B.; Zitnick, C.L.; Freeman, W.T. Automatic Estimation and Removal of Noise from a Single Image. *IEEE Trans. Pattern Anal. Mach. Intell.* **2008**, *30*, 299–314. [[CrossRef](#)] [[PubMed](#)]
- Dong, L.; Zhou, J.; Tang, Y.Y. Effective and Fast Estimation for Image Sensor Noise via Constrained Weighted Least Squares. *IEEE Trans. Image Process.* **2018**, *27*, 2715–2730. [[CrossRef](#)] [[PubMed](#)]

23. Gastasini, E.; Capecci, N.; Lupi, F.; Gagliardi, A.; Saponara, S.; Lanzetta, M. An Instrument for the Characterization and Calibration of Optical Sensors. *Sensors* **2021**, *21*, 5141. [[CrossRef](#)] [[PubMed](#)]
24. Evtikhiev, N.N.; Kozlov, A.V.; Krasnov, V.V.; Rodin, V.G.; Starikov, R.S.; Cheremkhin, P.A. Estimation of the Efficiency of Digital Camera Photosensor Noise Measurement through the Automatic Segmentation of Non-Uniform Target Methods and the Standard EMVA 1288. *Meas. Tech.* **2021**, *64*, 296–304. [[CrossRef](#)]
25. Kozlov, A.V.; Rodin, V.G.; Starikov, R.S.; Evtikhiev, N.N.; Cheremkhin, P.A. Estimation of Camera's Noise by Uniform Target Segmentation. *IEEE Sens. J.* **2023**, *23*, 4883–4891. [[CrossRef](#)]
26. Evtikhiev, N.N.; Kozlov, A.V.; Krasnov, V.V.; Rodin, V.G.; Starikov, R.S.; Cheremkhin, P.A. A Method for Measuring Digital Camera Noise by Automatic Segmentation of a Striped Target. *Comput. Opt.* **2021**, *45*, 267–276. [[CrossRef](#)]
27. Bilcu, R.C.; Vehvilainen, M. A New Method for Noise Estimation in Images. In Proceedings of the IEEE-Eurasip Nonlinear Signal and Image Processing, Sapporo, Japan, 18–20 May 2005; p. 25. [[CrossRef](#)]
28. Tai, S.-C.; Yang, S.-M. A Fast Method for Image Noise Estimation Using Laplacian Operator and Adaptive Edge Detection. In Proceedings of the 3rd International Symposium on Communications, Control and Signal Processing, St Julians, Malta, 12–14 March 2008; pp. 1077–1081. [[CrossRef](#)]
29. Rank, K.; Lendl, M.; Unbehauen, R. Estimation of Image Noise Variance. *IEE Proc.—Vis. Image Signal Process.* **1999**, *146*, 80. [[CrossRef](#)]
30. Yang, S.-M. Fast and Reliable Image-Noise Estimation Using a Hybrid Approach. *J. Electron. Imaging* **2010**, *19*, 033007. [[CrossRef](#)]
31. De Stefano, A.; White, P.R.; Collis, W.B. Training Methods for Image Noise Level Estimation on Wavelet Components. *EURASIP J. Adv. Signal Process.* **2004**, *2004*, 405209. [[CrossRef](#)]
32. Starck, J.; Murtagh, F. Automatic Noise Estimation from the Multiresolution Support. *Publ. Astron. Soc. Pac.* **1998**, *110*, 193–199. [[CrossRef](#)]
33. Pimpalkhute, V.A.; Page, R.; Kothari, A.; Bhurchandi, K.M.; Kamble, V.M. Digital Image Noise Estimation Using DWT Coefficients. *IEEE Trans. Image Process.* **2021**, *30*, 1962–1972. [[CrossRef](#)]
34. Donoho, D.L. De-Noising by Soft-Thresholding. *IEEE Trans. Inf. Theory* **1995**, *41*, 613–627. [[CrossRef](#)]
35. Hashemi, M.; Beheshti, S. Adaptive Noise Variance Estimation in BayesShrink. *IEEE Signal Process. Lett.* **2009**, *17*, 12–15. [[CrossRef](#)]
36. Liu, X.; Tanaka, M.; Okutomi, M. Single-Image Noise Level Estimation for Blind Denoising. *IEEE Trans. Image Process.* **2013**, *22*, 5226–5237. [[CrossRef](#)]
37. Pyatykh, S.; Hesser, J.; Zheng, L. Image Noise Level Estimation by Principal Component Analysis. *IEEE Trans. Image Process.* **2013**, *22*, 687–699. [[CrossRef](#)]
38. Amer, A.; Mitiche, A.; Dubois, E. Reliable and Fast Structure-Oriented Video Noise Estimation. In Proceedings of the International Conference on Image Processing, Rochester, NY, USA, 22–25 September 2003; p. I. [[CrossRef](#)]
39. Chen, G.; Zhu, F.; Heng, P.A. An Efficient Statistical Method for Image Noise Level Estimation. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 477–485. [[CrossRef](#)]
40. Ponomarenko, N.N.; Lukin, V.V.; Zriakhov, M.S.; Kaarna, A.; Astola, J. An Automatic Approach to Lossy Compression of AVIRIS Images. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, Barcelona, Spain, 23–28 July 2007; pp. 472–475. [[CrossRef](#)]
41. Shin, D.-H.; Park, R.-H.; Yang, S.; Jung, J.-H. Block-Based Noise Estimation Using Adaptive Gaussian Filtering. *IEEE Trans. Consum. Electron.* **2005**, *51*, 218–226. [[CrossRef](#)]
42. Danielyan, A.; Foi, A. Noise Variance Estimation in Nonlocal Transform Domain. In Proceedings of the International Workshop on Local and Non-Local Approximation in Image Processing, Tuusula, Finland, 19–21 August 2009; pp. 41–45. [[CrossRef](#)]
43. Li, F.; Fang, F.; Li, Z.; Zeng, T. Single Image Noise Level Estimation by Artificial Noise. *Signal Process.* **2023**, *213*, 109215. [[CrossRef](#)]
44. Luka, J.; Fridrich, J.; Goljan, M. Digital Camera Identification from Sensor Pattern Noise. *IEEE Trans. Inf. Forensics Secur.* **2006**, *1*, 205–214. [[CrossRef](#)]
45. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; The MIT Press: Cambridge, MA, USA, 2016; ISBN 9780262035613.
46. Gloe, T.; Böhme, R. The “Dresden Image Database” for Benchmarking Digital Image Forensics. In Proceedings of the 2010 ACM Symposium on Applied Computing—SAC '10, Sierre, Switzerland, 22–26 March 2010; pp. 1584–1590. [[CrossRef](#)]
47. Abdelhamed, A.; Lin, S.; Brown, M.E. A High-Quality Denoising Dataset for Smartphone Cameras. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 1692–1700. [[CrossRef](#)]
48. Plötz, T.; Roth, S. Benchmarking Denoising Algorithms with Real Photographs. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1586–1595. [[CrossRef](#)]
49. Chen, C.; Chen, Q.; Xu, J.; Koltun, V. Learning to See in the Dark. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 3291–3300. [[CrossRef](#)]

50. Wei, K.; Fu, Y.; Zheng, Y.; Yang, J. Physics-Based Noise Modeling for Extreme Low-Light Photography. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 8520–8537. [[CrossRef](#)] [[PubMed](#)]
51. Martin, D.; Fowlkes, C.; Tal, D.; Malik, J. A Database of Human Segmented Natural Images and Its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics. In Proceedings of the Eighth IEEE International Conference on Computer Vision ICCV 2001, Vancouver, BC, Canada, 7–14 July 2001. [[CrossRef](#)]
52. Wu, X. Color Demosaicking by Local Directional Interpolation and Nonlocal Adaptive Thresholding. *J. Electron. Imaging* **2011**, *20*, 023016. [[CrossRef](#)]
53. Arbeláez, P.; Maire, M.; Fowlkes, C.; Malik, J. Contour Detection and Hierarchical Image Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 898–916. [[CrossRef](#)]
54. Nam, S.; Hwang, Y.; Matsushita, Y.; Kim, S.J. A Holistic Approach to Cross-Channel Image Noise Modeling and Its Application to Image Denoising. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 1683–1691. [[CrossRef](#)]
55. Fu, B.; Zhang, X.; Wang, L.; Ren, Y.; Thanh, D.N.H. A Blind Medical Image Denoising Method with Noise Generation Network. *J. X-Ray Sci. Technol.* **2022**, *30*, 531–547. [[CrossRef](#)]
56. Abdelhamed, A.; Brubaker, M.; Brown, M. Noise Flow: Noise Modeling with Conditional Normalizing Flows. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 3165–3173. [[CrossRef](#)]
57. Chang, K.-C.; Wang, R.; Lin, H.-J.; Liu, Y.-L.; Chen, C.-P.; Chang, Y.-L.; Chen, H.-T. Learning Camera-Aware Noise Models. *Lect. Notes Comput. Sci.* **2020**, *12369*, 343–358. [[CrossRef](#)]
58. Zou, Y.; Fu, Y. Estimating Fine-Grained Noise Model via Contrastive Learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 12672–12681. [[CrossRef](#)]
59. Jang, G.; Lee, W.; Son, S.; Lee, K. C2N: Practical Generative Noise Modeling for Real-World Denoising. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 2330–2339. [[CrossRef](#)]
60. Maleky, A.; Kousha, S.; Brown, M.S.; Brubaker, M.A. Noise2NoiseFlow: Realistic Camera Noise Modeling without Clean Images. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 17611–17620. [[CrossRef](#)]
61. Lehtinen, J.; Munkberg, J.; Hasselgren, J.; Laine, S.; Karras, T.; Aittala, M.; Aila, T. Noise2Noise: Learning Image Restoration without Clean Data. *arXiv* **2018**, arXiv:1803.04189. [[CrossRef](#)]
62. Hubel, D.H.; Wiesel, T.N. Receptive Fields, Binocular Interaction and Functional Architecture in the Cat's Visual Cortex. *J. Physiol.* **1962**, *160*, 106–154. [[CrossRef](#)]
63. Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-Based Learning Applied to Document Recognition. *Proc. IEEE* **2018**, *86*, 2278–2324. [[CrossRef](#)]
64. Boureau, Y.-L.; Ponce, J.; LeCun, Y. A Theoretical Analysis of Feature Pooling in Visual Recognition. In Proceedings of the 27th International Conference on Machine Learning (ICML-10), Haifa, Israel, 21–24 June 2010; pp. 111–118.
65. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556. [[CrossRef](#)]
66. Nair, V.; Hinton, G. Rectified Linear Units Improve Restricted Boltzmann Machines. In Proceedings of the 27th International Conference on Machine Learning (ICML-10), Haifa, Israel, 21–24 June 2010; pp. 807–814.
67. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [[CrossRef](#)]
68. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V. Going Deeper with Convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1–9. [[CrossRef](#)]
69. Minaee, S.; Boykov, Y.Y.; Porikli, F.; Plaza, A.J.; Kehtarnavaz, N.; Terzopoulos, D. Image Segmentation Using Deep Learning: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 3523–3542. [[CrossRef](#)] [[PubMed](#)]
70. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. *Lect. Notes Comput. Sci.* **2015**, *9351*, 234–241. [[CrossRef](#)]
71. Schawinski, K.; Zhang, C.; Zhang, H.; Fowler, L.; Santhanam, G.K. Generative Adversarial Networks Recover Features in Astrophysical Images of Galaxies beyond the Deconvolution Limit. *Mon. Not. R. Astron. Soc. Lett.* **2017**, *467*, L110–L114. [[CrossRef](#)]
72. Krull, A.; Vičar, T.; Prakash, M.; Lalit, M.; Jug, F. Probabilistic Noise2Void: Unsupervised Content-Aware Denoising. *Front. Comput. Sci.* **2020**, *2*, 5. [[CrossRef](#)]
73. Chi, J.; Wu, C.; Yu, X.; Ji, P.; Chu, H. Single Low-Dose CT Image Denoising Using a Generative Adversarial Network with Modified U-Net Generator and Multi-Level Discriminator. *IEEE Access* **2020**, *8*, 133470–133487. [[CrossRef](#)]

74. Zou, Y.; Fu, Y.; Zhang, Y.; Zhang, T.; Yan, C.; Timofte, R. Calibration-Free Raw Image Denoising via Fine-Grained Noise Estimation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2025**, *47*, 5368–5384. [[CrossRef](#)]
75. Zhang, K.; Zuo, W.; Chen, Y.; Meng, D.; Zhang, L. Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising. *IEEE Trans. Image Process.* **2017**, *26*, 3142–3155. [[CrossRef](#)]
76. Li, L.; Song, S.; Lv, M.; Jia, Z.; Ma, H. Multi-Focus Image Fusion Based on Fractal Dimension and Parameter Adaptive Unit-Linking Dual-Channel PCNN in Curvelet Transform Domain. *Fractal Fract.* **2025**, *9*, 157. [[CrossRef](#)]
77. Cao, Z.-H.; Liang, Y.-J.; Deng, L.-J.; Vivone, G. An Efficient Image Fusion Network Exploiting Unifying Language and Mask Guidance. *IEEE Trans. Pattern Anal. Mach. Intell.* **2025**, 1–18. [[CrossRef](#)]
78. Sun, H.; Duan, R.; Sun, G.; Zhang, H.; Chen, F.; Yang, F.; Cao, J. SARFT-GAN: Semantic-Aware ARConv Fused Top-k Generative Adversarial Network for Remote Sensing Image Denoising. *Remote Sens.* **2025**, *17*, 3114. [[CrossRef](#)]
79. Yu, S.; Park, B.; Jeong, J. Deep Iterative Down-up CNN for Image Denoising. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Long Beach, CA, USA, 16–17 June 2019; pp. 2095–2103. [[CrossRef](#)]
80. Hong, Z.; Fan, X.; Jiang, T.; Feng, J. End-To-End Unpaired Image Denoising with Conditional Adversarial Networks. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 4140–4149. [[CrossRef](#)]
81. Zou, Y.; Yan, C.; Fu, Y. Iterative Denoiser and Noise Estimator for Self-Supervised Image Denoising. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 1–6 October 2023; pp. 13219–13228. [[CrossRef](#)]
82. Zeyde, R.; Elad, M.; Protter, M. On Single Image Scale-up Using Sparse-Representations. *Lect. Notes Comput. Sci.* **2010**, *6920*, 711–730. [[CrossRef](#)]
83. Wang, Z.; Liu, J.; Li, G.; Han, H.J. Blind2Unblind: Self-Supervised Image Denoising with Visible Blind Spots. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 2017–2026. [[CrossRef](#)]
84. Zeng, H.; Hosseini, M.D.M.; Deng, K.; Peng, A.; Goljan, M. A Comparison Study of CNN Denoisers on PRNU Extraction. Available online: <https://arxiv.org/abs/2112.02858> (accessed on 3 April 2024).
85. Tian, C.; Xu, Y.; Li, Z.; Zuo, W.; Fei, L.; Liu, H. Attention-Guided CNN for Image Denoising. *Neural Netw.* **2020**, *124*, 117–129. [[CrossRef](#)] [[PubMed](#)]
86. Yue, Z.; Zhao, Q.; Zhang, L.; Meng, D. Dual Adversarial Network: Toward Real-World Noise Removal and Noise Generation. *Lect. Notes Comput. Sci.* **2020**, *12355*, 41–58. [[CrossRef](#)]
87. Kousha, S.; Maleky, A.; Brown, M.S.; Brubaker, M.A. Modeling SRGB Camera Noise with Normalizing Flows. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 17442–17450. [[CrossRef](#)]
88. Tan, H.; Xiao, H.; Lai, S.; Liu, Y.; Zhang, M. Pixelwise Estimation of Signal-Dependent Image Noise Using Deep Residual Learning. *Comput. Intell. Neurosci.* **2019**, *2019*, 1–12. [[CrossRef](#)]
89. Zhang, K.; Zuo, W.; Zhang, L. FFDNet: Toward a Fast and Flexible Solution for CNN-Based Image Denoising. *IEEE Trans. Image Process.* **2018**, *27*, 4608–4622. [[CrossRef](#)]
90. Ma, R.; Zhang, Y.; Zhang, B.; Fang, L.; Huang, D.; Qi, L. Learning Attention in the Frequency Domain for Flexible Real Photograph Denoising. *IEEE Trans. Image Process.* **2024**, *33*, 3707–3721. [[CrossRef](#)]
91. Zamir, S.W.; Arora, A.; Khan, S.; Hayat, M.; Khan, F.S.; Yang, M.-H.; Shao, L. CycleISP: Real Image Restoration via Improved Data Synthesis. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 2693–2702. [[CrossRef](#)]
92. Zamir, S.W.; Arora, A.; Khan, S.; Hayat, M.; Khan, F.S.; Yang, M.-H.; Shao, L. Multi-Stage Progressive Image Restoration. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 14816–14826. [[CrossRef](#)]
93. Zamir, S.W.; Arora, A.; Khan, S.; Hayat, M.; Khan, F.S.; Yang, M.-H. Restormer: Efficient Transformer for High-Resolution Image Restoration. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 5718–5729. [[CrossRef](#)]
94. Yu, J.; Zhou, Y.; Sun, M.; Wang, D. Dual-Path Adversarial Denoising Network Based on UNet. *Sensors* **2025**, *25*, 4751. [[CrossRef](#)]
95. Dabov, K.; Foi, A.; Katkovnik, V.; Egiazarian, K. Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering. *IEEE Trans. Image Process.* **2007**, *16*, 2080–2095. [[CrossRef](#)]
96. Wang, L.; Li, J.; Zhang, R.; Guo, X. Multi-Stage Progressive Generative Adversarial Network for Low-Dose CT Denoising. In Proceedings of the 6th International Conference on Communications, Information System and Computer Engineering (CISCE), Guangzhou, China, 10–12 May 2024; pp. 750–753. [[CrossRef](#)]

97. Mao, X.-J.; Shen, C.; Yang, Y.-B. Image Restoration Using Very Deep Convolutional Encoder-Decoder Networks with Symmetric Skip Connections. In Proceedings of the NIPS'16: Proceedings of the 30th International Conference on Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; Volume 29, pp. 2810–2818.
98. Santhanam, V.; Morariu, V.I.; Davis, L.S. Generalized Deep Image to Image Regression. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5395–5405. [[CrossRef](#)]
99. Tai, Y.; Yang, J.; Liu, X.; Xu, C. MemNet: A Persistent Memory Network for Image Restoration. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; Volume 2017, pp. 4549–4557. [[CrossRef](#)]
100. Park, B.; Yu, S.; Jeong, J. Densely Connected Hierarchical Network for Image Denoising. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Long Beach, CA, USA, 16–17 June 2019; pp. 2103–2113. [[CrossRef](#)]
101. Wang, M.; Yuan, P.; Qiu, S.; Jin, W.; Li, L.; Wang, X. Dual-Encoder UNet-Based Narrowband Uncooled Infrared Imaging Denoising Network. *Sensors* **2025**, *25*, 1476. [[CrossRef](#)]
102. Zhang, K.; Zuo, W.; Gu, S.; Zhang, L. Learning Deep CNN Denoiser Prior for Image Restoration. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2808–2817. [[CrossRef](#)]
103. Ignatov, A.; Kobyshev, N.; Timofte, R.; Vanhoey, K.; Gool, L.V. DSLR-Quality Photos on Mobile Devices with Deep Convolutional Networks. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 3297–3305. [[CrossRef](#)]
104. Bartlett, O.J.; Benoit, D.M.; Pimblet, K.A.; Simmons, B.; Hunt, L. Noise Reduction in Single-Shot Images Using an Auto-Encoder. *Mon. Not. R. Astron. Soc.* **2023**, *521*, 6318–6329. [[CrossRef](#)]
105. Guo, S.; Yan, Z.; Zhang, K.; Zuo, W.; Zhang, L. Toward Convolutional Blind Denoising of Real Photographs. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–17 June 2019; pp. 1712–1722. [[CrossRef](#)]
106. Xu, J.; Zhang, L.; Feng, X.; Zhang, D. Multi-Channel Weighted Nuclear Norm Minimization for Real Color Image Denoising. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 1105–1113. [[CrossRef](#)]
107. Xu, J.; Zhang, L.; Zhang, D. A Trilateral Weighted Sparse Coding Scheme for Real-World Image Denoising. *Lect. Notes Comput. Sci.* **2018**, *11212*, 21–38. [[CrossRef](#)]
108. Guo, B.; Song, K.; Dong, H.; Yan, Y.; Tu, Z.; Zhu, L. NERNet: Noise Estimation and Removal Network for Image Denoising. *J. Vis. Commun. Image Represent.* **2020**, *71*, 102851. [[CrossRef](#)]
109. Byun, J.; Cha, S.; Moon, T. FBI-Denoiser: Fast Blind Image Denoiser for Poisson-Gaussian Noise. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 5764–5773. [[CrossRef](#)]
110. Makitalo, M.; Foi, A. Optimal Inversion of the Generalized Anscombe Transformation for Poisson-Gaussian Noise. *IEEE Trans. Image Process.* **2013**, *22*, 91–103. [[CrossRef](#)]
111. Liu, X.; Tanaka, M.; Okutomi, M. Practical Signal-Dependent Noise Parameter Estimation from a Single Noisy Image. *IEEE Trans. Image Process.* **2014**, *23*, 4361–4371. [[CrossRef](#)]
112. Roth, S.; Black, M.M. Fields of Experts: A Framework for Learning Image Priors. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–25 June 2005; pp. 860–867. [[CrossRef](#)]
113. Bychkovskiy, V.; Paris, S.; Chan, E.; Durand, F. Learning Photographic Global Tonal Adjustment with a Database of Input/Output Image Pairs. In Proceedings of the CVPR 2011, Colorado Springs, CO, USA, 20–25 June 2011; pp. 97–104. [[CrossRef](#)]
114. Zhang, Y.; Zhu, Y.; Nichols, E.; Wang, Q.; Zhang, S.; Smith, C.; Howard, S. A Poisson-Gaussian Denoising Dataset with Real Fluorescence Microscopy Images. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–17 June 2019; pp. 11702–11710. [[CrossRef](#)]
115. Wu, X.; Liu, M.; Cao, Y.; Ren, D.; Zuo, W. Unpaired Learning of Deep Image Denoising. *Lect. Notes Comput. Sci.* **2020**, *12349*, 352–368. [[CrossRef](#)]
116. Cha, S.; Moon, T. Fully Convolutional Pixel Adaptive Image Denoiser. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 4159–4168. [[CrossRef](#)]
117. Li, X.; Wang, Z.; Fang, H.; Fan, Z.; Li, S.; Huang, Z. Adaptive Image Noise Level Estimation with Chi-Square Distribution on the Flat Patches Selected by Improved PCANet and ResNet101. *Optik* **2023**, *287*, 171107. [[CrossRef](#)]
118. Chan, T.-H.; Jia, K.; Gao, S.; Lu, J.; Zeng, Z.; Ma, Y. PCANet: A Simple Deep Learning Baseline for Image Classification? *IEEE Trans. Image Process.* **2015**, *24*, 5017–5032. [[CrossRef](#)]
119. Rao, Y.; He, L.; Zhu, J. A Residual Convolutional Neural Network for Pan-Sharpening. In Proceedings of the International Workshop on Remote Sensing with Intelligent Processing (RSIP), Shanghai, China, 18–21 May 2017; pp. 1–4. [[CrossRef](#)]
120. Tian, C.; Xu, Y.; Zuo, W.; Du, B.; Lin, C.-W.; Zhang, D. Designing and Training of a Dual CNN for Image Denoising. *Knowl. Based Syst.* **2021**, *226*, 106949. [[CrossRef](#)]

121. Wu, W.; Ge, A.; Lv, G.; Xia, Y.; Zhang, Y.; Xiong, W. Two-Stage Progressive Residual Dense Attention Network for Image Denoising. *arXiv* **2024**, arXiv:2401.02831. [[CrossRef](#)]
122. Wu, W.; Lv, G.; Duan, Y.; Liang, P.; Zhang, Y.; Xia, Y. Dual Convolutional Neural Network with Attention for Image Blind Denoising. *Multimed. Syst.* **2024**, *30*, 263. [[CrossRef](#)]
123. Wu, W.; Liao, S.; Lv, G.; Liang, P.; Zhang, Y. Image Blind Denoising Using Dual Convolutional Neural Network with Skip Connection. *Signal Process. Image Commun.* **2025**, *138*, 117365. [[CrossRef](#)]
124. Zhang, K.; Li, Y.; Zuo, W.; Zhang, L.; Van Gool, L.; Timofte, R. Plug-And-Play Image Restoration with Deep Denoiser Prior. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 6360–6376. [[CrossRef](#)] [[PubMed](#)]
125. Wischow, M.; Irmisch, P.; Boerner, A.; Gallego, G. Real-Time Noise Source Estimation of a Camera System from an Image and Metadata. *Adv. Intell. Syst.* **2024**, *6*, 2300479. [[CrossRef](#)]
126. Huang, Y.; Huang, H. Beyond Image Prior: Embedding Noise Prior into Conditional Denoising Transformer. *arXiv* **2024**, arXiv:2407.09094. [[CrossRef](#)]
127. Huang, J.-B.; Singh, A.; Ahuja, N. Single Image Super-Resolution from Transformed Self-Exemplars. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 5197–5208. [[CrossRef](#)]
128. Makitalo, M.; Foi, A. Noise Parameter Mismatch in Variance Stabilization, with an Application to Poisson–Gaussian Noise Estimation. *IEEE Trans. Image Process.* **2014**, *23*, 5348–5359. [[CrossRef](#)]
129. Wang, Z.; Cun, X.; Bao, J.; Zhou, W.; Liu, J.; Li, H. Uformer: A General U-Shaped Transformer for Image Restoration. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 17662–17672. [[CrossRef](#)]
130. Guo, H.; Li, J.; Dai, T.; Ouyang, Z.; Ren, X.; Xia, S.-T. MambaIR: A Simple Baseline for Image Restoration with State-Space Model. *Lect. Notes Comput. Sci.* **2024**, *15076*, 222–241. [[CrossRef](#)]
131. Tuama, A.; Comby, F.; Chaumont, M. Camera Model Identification with the Use of Deep Convolutional Neural Networks. In Proceedings of the IEEE International Workshop on Information Forensics and Security (WIFS), Abu Dhabi, United Arab Emirates, 4–7 December 2016; pp. 1–6. [[CrossRef](#)]
132. Chen, Y.; Huang, Y.; Ding, X. Camera Model Identification with Residual Neural Network. In Proceedings of the IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; pp. 4337–4341. [[CrossRef](#)]
133. Bondi, L.; Baroffio, L.; Guera, D.; Bestagini, P.; Delp, E.J.; Tubaro, S. First Steps toward Camera Model Identification with Convolutional Neural Networks. *IEEE Signal Process. Lett.* **2017**, *24*, 259–263. [[CrossRef](#)]
134. Huang, N.; He, J.; Zhu, N.; Xuan, X.; Liu, G.; Chang, C. Identification of the Source Camera of Images Based on Convolutional Neural Network. *Digit. Investig.* **2018**, *26*, 72–80. [[CrossRef](#)]
135. Yao, H.; Qiao, T.; Xu, M.; Zheng, N. Robust Multi-Classifer for Camera Model Identification Based on Convolution Neural Network. *IEEE Access* **2018**, *6*, 24973–24982. [[CrossRef](#)]
136. Marra, F.; Gragnaniello, D.; Verdoliva, L. On the Vulnerability of Deep Learning to Adversarial Attacks for Camera Model Identification. *Signal Process. Image Commun.* **2018**, *65*, 240–248. [[CrossRef](#)]
137. Wang, B.; Yin, J.; Tan, S.; Li, Y.; Li, M. Source Camera Model Identification Based on Convolutional Neural Networks with Local Binary Patterns Coding. *Signal Process. Image Commun.* **2018**, *68*, 162–168. [[CrossRef](#)]
138. Ding, X.; Chen, Y.; Tang, Z.; Huang, Y. Camera Identification Based on Domain Knowledge-Driven Deep Multi-Task Learning. *IEEE Access* **2019**, *7*, 25878–25890. [[CrossRef](#)]
139. Sameer, V.U.; Naskar, R. Deep Siamese Network for Limited Labels Classification in Source Camera Identification. *Multimed. Tools Appl.* **2020**, *79*, 28079–28104. [[CrossRef](#)]
140. Freire-Obregón, D.; Narducci, F.; Barra, S.; Castrillón-Santana, M. Deep Learning for Source Camera Identification on Mobile Devices. *Pattern Recognit. Lett.* **2019**, *126*, 86–91. [[CrossRef](#)]
141. Bennabhaktula, G.S.; Alegre, E.; Karastoyanova, D.; Azzopardi, G. Camera Model Identification Based on Forensic Traces Extracted from Homogeneous Patches. *Expert Syst. Appl.* **2022**, *206*, 117769. [[CrossRef](#)]
142. Bennabhaktula, G.S.; Timmerman, D.; Alegre, E.; Azzopardi, G. Source Camera Device Identification from Videos. *SN Comput. Sci.* **2022**, *3*, 316. [[CrossRef](#)]
143. Bharathiraja, S.; Rajesh Kanna, B.; Hariharan, M. A Deep Learning Framework for Image Authentication: An Automatic Source Camera Identification Deep-Net. *Arab. J. Sci. Eng.* **2022**, *48*, 1207–1219. [[CrossRef](#)]
144. Huan, S.; Liu, Y.; Yang, Y.; Law, N.-F.B. Camera Model Identification Based on Dual-Path Enhanced ConvNeXt Network and Patches Selected by Uniform Local Binary Pattern. *Expert Syst. Appl.* **2023**, *241*, 122501. [[CrossRef](#)]
145. Sychandran, C.S.; Shreelekshmi, R. SCCRNNet: A Framework for Source Camera Identification on Digital Images. *Neural Comput. Appl.* **2023**, *36*, 1167–1179. [[CrossRef](#)]
146. Liu, Y.-y.; Chen, C.; Lin, H.; Li, Z. A New Camera Model Identification Method Based on Color Correction Features. *Multimed. Tools Appl.* **2023**, *83*, 29179–29195. [[CrossRef](#)]

147. Nayerifard, T.; Amintoosi, H.; Ghaemi Bafghi, A. A Robust PRNU-Based Source Camera Attribution with Convolutional Neural Networks. *J. Supercomput.* **2024**, *81*, 25. [[CrossRef](#)]
148. Martín-Rodríguez, F.; Isasi-de-Vicente, F.; Fernández-Barciela, M. A Stress Test for Robustness of Photo Response Nonuniformity (Camera Sensor Fingerprint) Identification on Smartphones. *Sensors* **2023**, *23*, 3462. [[CrossRef](#)] [[PubMed](#)]
149. Bernacki, J. Robustness of Digital Camera Identification with Convolutional Neural Networks. *Multimed. Tools Appl.* **2021**, *80*, 29657–29673. [[CrossRef](#)]
150. Manisha; Li, C.-T.; Lin, X.; Kotegar, K.A. Beyond PRNU: Learning Robust Device-Specific Fingerprint for Source Camera Identification. *Sensors* **2022**, *22*, 7871. [[CrossRef](#)]
151. Liu, Y.; Xiao, Y.; Tian, H. Plug-And-Play PRNU Enhancement Algorithm with Guided Filtering. *Sensors* **2024**, *24*, 7701. [[CrossRef](#)]
152. Yang, P.; Baracchi, D.; Ni, R.; Zhao, Y.; Argenti, F.; Piva, A. A Survey of Deep Learning-Based Source Image Forensics. *J. Imaging* **2020**, *6*, 9. [[CrossRef](#)]
153. Tian, N.; Qiu, X.; Pan, Q. An Improved PRNU Noise Extraction Model for Highly Compressed Image Blocks with Low Resolutions. *Multimed. Tools Appl.* **2024**, *83*, 66657–66690. [[CrossRef](#)]
154. Liu, Y.; Zou, Z.; Yang, Y.; Law, N.-F.; Bharath, A.A. Efficient Source Camera Identification with Diversity-Enhanced Patch Selection and Deep Residual Prediction. *Sensors* **2021**, *21*, 4701. [[CrossRef](#)]
155. Rafi, A.M.; Tonmoy, T.I.; Kamal, U.; Wu, Q.M.J.; Hasan, M.K. RemNet: Remnant Convolutional Neural Network for Camera Model Identification. *Neural Comput. Appl.* **2020**, *33*, 3655–3670. [[CrossRef](#)]
156. Hui, C.; Jiang, F.; Liu, S.; Zhao, D. Source Camera Identification with Multi-Scale Feature Fusion Network. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2022. [[CrossRef](#)]
157. Timmerman, D.; Bennabhaktula, S.; Alegre, E.; Azzopardi, G. Video Camera Identification from Sensor Pattern Noise with a Constrained ConvNet. *arXiv* **2020**, arXiv:2012.06277. [[CrossRef](#)]
158. Shullani, D.; Fontani, M.; Iuliani, M.; Shaya, O.A.; Piva, A. VISION: A Video and Image Dataset for Source Identification. *EURASIP J. Inf. Secur.* **2017**, *2017*, 15. [[CrossRef](#)]
159. Edwards, T. *Discrete Wavelet Transforms: Theory and Implementation*; Stanford University: Stanford, CA, USA, 1991.
160. Zeng, H.; Wan, Y.; Deng, K.; Peng, A. Source Camera Identification with Dual-Tree Complex Wavelet Transform. *IEEE Access* **2020**, *8*, 18874–18883. [[CrossRef](#)]
161. Tian, H.; Xiao, Y.; Cao, G.; Zhang, Y.; Xu, Z.; Zhao, Y. Daxing Smartphone Identification Dataset. *IEEE Access* **2019**, *7*, 101046–101053. [[CrossRef](#)]
162. Goljan, M.; Fridrich, J.; Filler, T. Large Scale Test of Sensor Fingerprint Camera Identification. In Proceedings of the Media Forensics Security, San Jose, CA, USA, 19 January 2009; Volume 7254. [[CrossRef](#)]
163. Kligvasser, I.; Shaham, T.R.; Michaeli, T. XUnit: Learning a Spatial Activation Function for Efficient Image Restoration. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 2433–2542. [[CrossRef](#)]
164. Magistris, G.; Grycuk, R.; Mandelli, L.; Scherer, R. New Approaches Based on PRNU-CNN for Image Camera Source Attribution in Forensic Investigations. In Proceedings of the SYSYEM 2024: 10th Scholar’s Yearly Symposium of Technology, Engineering and Mathematics, Rome, Italy, 2–5 December 2024; pp. 67–72.
165. Torres, L.; Barrios, C.; Denneulin, Y. Computational Resource Consumption in Convolutional Neural Network Training—A Focus on Memory. *Supercomput. Front. Innov.* **2021**, *8*, 45–61. [[CrossRef](#)]
166. de Roos, L.; Geradts, Z. Factors That Influence PRNU-Based Camera-Identification via Videos. *J. Imaging* **2021**, *7*, 8. [[CrossRef](#)]
167. Tripathi, M. Facial Image Noise Classification and Denoising Using Neural Network. *Sustain. Eng. Innov.* **2021**, *3*, 102–111. [[CrossRef](#)]
168. Mudhafar, R.; Abbadi, N. Image Noise Detection and Classification Based on Combination of Deep Wavelet and Machine Learning. *Al-Salam J. Eng. Technol.* **2023**, *3*, 23–36. [[CrossRef](#)]
169. Kavitha, G.; Prakash, C.; Alhomrani, M.; Pradeep, N.; Alamri, A.S.; Pareek, P.K.; Alhassan, M. Noise Estimation and Type Identification in Natural Scene and Medical Images Using Deep Learning Approaches. *Contrast Media Mol. Imaging* **2023**, *2023*, 1–15. [[CrossRef](#)]
170. Martín-Rodríguez, F.; Garcia-Mojon, R.; Fernández-Barciela, M. Detection of AI-Created Images Using Pixel-Wise Feature Extraction and Convolutional Neural Networks. *Sensors* **2023**, *23*, 9037. [[CrossRef](#)]
171. Kingra, S.; Aggarwal, N.; Kaur, N. SiamNet: Exploiting Source Camera Noise Discrepancies Using Siamese Network for Deepfake Detection. *Inf. Sci.* **2023**, *645*, 119341. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.