The Robustness of the Edge Detection on Noisy Natural Images with HED and PiDiNet Networks

Marina Polyakova[1], Serhii Khyrenko[1]

[1] *Odesa Polytechnic National University, 1, Shevchenko Ave., Odesa, 65044, Ukraine*

Abstract

The edge detection performance of the HED and PiDiNet networks is compared on noised natural images. To research the robustness of edge detection with HED and PiDiNet networks, the technique of detecting the influence of the noise level on the CNN edge detector efficiency is proposed. At first, as a result of literature review, HED and PiDiNet are selected to detect edges on noisy images. Then types of image noise of interest to the researcher and the noise parameters which will be controlled by the researcher are determined. We considered white Gaussian noise, impulse noise, and speckle noise. Mathematical modeling of noisy images is performed. Next, the training and test sets from image datasets for edge detection is selected. After that, the researched networks are learned to detect edges on training set of images or weights of pre-trained networks are loaded. Further the measures of edge detection performance are selected, specifically, Precision, Recall, F1-score, and Pratt's Figure of Merit. Then images of test set are corrupted with controlled level of noise. The trained networks are applied to noisy images and the edge detection performance is evaluated depending on the noise level in the images. The obtained results are analyzed to generate the recommendations allowing to determine which CNN is better for edge detection when natural images affected by different level of white Gaussian noise, or impulse noise, or speckle noise. In particular, the HED network is generally preferred at high noise levels. The PiDiNet network is best used when the noise level is low.

Keywords

Edge detection, noisy image, HED network, PiDiNet network

1. Introduction

The natural image edge detection is implemented in intelligent systems and computer vision to search images in databases and the Internet, to monitor the transport, infrastructure, and environment. The continuous growth of the computing power allows implementing the image edge detection even in mobile devices. Natural images are distinguished by smooth color change areas, a low level of noise, and texture regions. Then to detect object edges on natural images it is necessary to ignore the noisy pixels and background texture to establish a correspondence between boundaries of objects and the color differences [1, 2].

The object of research is the natural image edge detection in computer graphics and intelligent systems.

Much better results compared to conventional edge detection methods can be obtained by using convolutional neural networks (CNNs). CNNs automatically learn to identify features of images at different levels of abstraction. For example, the initial layers of the network can detect an edges and boundaries of texture regions, while the deeper layers concentrate on combining the obtained edges into the boundaries of more complex structures, such as objects. An important aspect of the use of deep learning artificial intelligence is their ability to fine-tune for solving similar tasks after preliminary training. This makes it possible to process images of different complexity and content by the same CNN [3, 4].

The key factors determining the edge detection performance are the level of detailing, that is, how many different objects are visible in the image, and the level of image noise. The first factor is regulated by CNN architecture and computing power of research devices. The second factor, the level of noise in the image, can still be a challenge and affect the result. Different CNNs have different edge detection performance on the same image, but with different noise levels.

CMIS-2024: Seventh International Workshop on Computer Modeling and Intelligent Systems, May 3, 2024, Zaporizhzhia, Ukraine

✉ marinapolyakova943@gmail.com (M. Polyakova); sergeyhyrenko@gmail.com (S. Khyrenko)
iD 0000-0001-7229-7657 (M. Polyakova); 0009-0003-2353-6617 (S. Khyrenko)

It should be noted that image noise can significantly complicate the edge detection and affect the quality of the obtained results. Usually, noise can occur due to various factors, such as electromagnetic interference, equipment defects, accidental impacts during photography or filming. Noise appears in random fluctuations of the intensity of pixels, which can lead to a loss of edge quality and the appearance of false details. Images affected by Gaussian white noise, impulse or speckle noise are most often considered.

The subject of the research is the robustness of edge detection with the CNNs, namely, the dependence of edge detection performance on the level of image noise.

The Holistically-nested edge detection (HED) network was chosen for the robustness research because of its multi-layered architecture with side replies to process images with varying levels of details. The next advantage of this network is the comparison of interpolated side replies with ground truth image to understand the overall context of the original noised image. Then HED architecture is used in autopilot systems of vehicles. The robustness of edge detection by Pixel difference network (PiDiNet), which also uses context analysis around the processed pixel, is also researched. But unlike the HED network, it applies a gradient estimation operator compatible with conventional convolution.

The aim of research is to compare the robustness of HED and PiDiNet networks to detect edges on noised natural images. The comparison will help determine which network is better for edge detection when natural images affected by white Gaussian noise, impulse noise, or speckle noise.

2. Problem statement

In this paper, the natural images are selected for research. The color natural image is represented as $I(x,y)=(I_R(x,y),\ I_G(x,y),\ I_B(x,y)\}$, where $x=1, \ldots, n$; $y=1, \ldots, l$. Here n is the number of rows of the image; l is the number of columns of the image; (x,y) are coordinates of the image pixel; $I(x,y)$ is a vector function representing an image by color channels; $I_R(x,y)$, $I_G(x,y)$, $I_B(x,y)$ are the functions of intensity of the red, green, or blue color channels respectively. To detect edges, each pixel of the original image must be associated with the value of the target feature. There is a label of one of two classes, specifically, 1 for boundary pixels, 0 for pixels inside homogeneous areas. The values of the target feature for the natural image should be represented as a binary image which is the result of edge detection [2].

There are determined the types of noise that are of interest to the researcher and inherent in natural images. The models of noise of these types are selected as $M = \{M_1, M_2, \ldots , M_L\}$, where L is a number of researched noise models. One or more parameters determining the noise level in the image are selected for each model.

Let an NN is a set of CNNs of deep learning $NN = \{NN_1, NN_2, \ldots , NN_K\}$, where K is a number of researched CNNs. These CNNs are designed to detect edges on natural images and trained on test database.

The problem of researching of the robustness of the edge detection with the CNNs is as follows. It is necessary to determine the dependences of the edge detection performance on the value of the noise parameter for each researched network and each considered type of noise on a selected class of images. Using the obtained dependencies, it is necessary to elaborate recommendations for the use of a specific neural network to detect edges on an image with a certain type and level of noise.

3. Review of the literature

Recent surveys of CNNs designed to detect edges considered their architectures and the performance of edge detection with these networks [1, 5]. In this paper the robustness of edge detection on noised images using CNNs is researched. In the literature, in view of the robustness the conventional edge detectors not using CNNs are divided into two groups. These are differential and correlation for prototype matching methods [2].

The differential methods apply gradient operator to each color channel of the image. Then the gradient magnitude is calculated and thresholded for each color channel. Next, the results of edge detection of the color channels are combined into edge map of original image. Otherwise, the gradient vectors of color channels are transformed using the norm operator into gray-scale image. Thresholding is performed only for intensities of this image and thus the color edges are obtained [6].

The advantages of differential methods are the high-speed performance and the low error of edge positioning. However, the robustness to noise is low because of the application of the gradient operators. Contour breaks and false edges occur. A wide variety of differential edge detection methods is due to attempts to apply gradient operators with the greatest possible robustness to noise using linear or nonlinear processing [2, 7].

To detect edges with correlation for prototype matching methods, W prototypes of edge are compiled. Based on these prototypes, the coefficients of the W filters are selected. The result of image processing by each of these filters is determined. Let the filtering result reach a maximum for some image area. This means that the edge intensity prototype matches the image area. The matching is defined by comparison with a threshold based on statistical decision theory. If edges are detected with correlation by prototype matching then several edge models are used, since the real edge can be ramped. These methods are robust to noise, but the positioning accuracy is low. The latter is due to poor discrimination of prototypes of different shape with the same energy. In addition, the use of W filters reduces high-speed performance of correlation by prototype matching [2, 7].

The authors propose to classify edge detectors with CNN into differential and correlation by prototype matching. We classify CNN as differential method, if some gradient operators are embedded into the convolution layers. If the architecture of the CNN includes conventional convolution layers, then we define edge detector with such CNN as correlation by prototype matching method. This is because the more similar the image area is to the convolution kernel, the larger the convolution result value.

CNNs that implement correlation by prototype matching methods for edge detection appeared earlier, since the convolutional layer is based on template matching. Examples of such networks are HED [8, 9], Richer convolutional features (RCF) network [10], Learning to predict crisp boundaries (LPCB) network [11] et al.

To resolve ambiguity in edge and object boundary detection, the HED is proposed in [8]. This fully convolutional network differs from other CNNs by side replies. When training the HED network, side replies are deeply-supervised and interpolated to initial image size. The results are fused to obtain multi-scale nested features. Thus rich hierarchical representation is automatically directed by deep supervision on HED side replies [12].

To solve the thick contour problem in edge detection the HED network is enhanced in the paper [11]. The obtained fully convolutional network of bottom-up/top-down architecture is named LPCB network. It is based on VGG16 network [13] and uses a new loss function, which evaluates image similarity and very effective for classifying unbalanced data. LPCB uses fewer parameters compared to the HED network. But the LPCB network shows better edge detection performance, and produces accurate results without post-processing.

Also, in [10] the RCF network is designed for accurate edge detection. It designed by removing the fully connected layer and the fifth pooling layer from the VGG16 network and estimates multi-scale features of the image. The RCF network uses convolutional layers with different perceptual fields following by pooling layers. Then the layer level features are fused, the automatic learning of all weight parameters is done. Thus, RCF combines the underlying feature maps to detect edges based on the pyramid architecture [3].

CNNs that implement differential edge detectors were designed to solve the problems of thick image contour and inaccurate positioning. These are PiDiNet [14], PiDiNeXt [15], Xception based network with fusion difference convolutions [16] et al.

A novel pixel difference convolution is integrated in [14] into network convolutional layer. The resulting PiDiNet network can easily retain the powerful learning ability of deep CNN to extract image features with semantic significance while capturing gradient information conducive to edge detection. Then the better edge detection accuracy with fewer parameters is achieved through the integration of the gradient estimation into the convolution operation.

Although PiDiNet achieves competitive results combining traditional difference operators with deep learning, the problem of the rationale behind the selection of gradient estimation operators remained unresolved. Therefore, in [15] the PiDiNeXt was proposed, which combines gradient estimation operators with deep learning-based model in parallel to solve the problem of operator selection and further feature improving. As a result, the experiments on test datasets demonstrate that PiDiNeXt outperforms PiDiNet in terms of edge detection accuracy.

In [16] the edge detection via fusion difference convolution is proposed. This new structure integrates vanilla convolutions with pixel difference convolutions to extract more semantic information from the image, thereby reducing noise and texture impact on the resulting edges.

Besides, fusion difference convolution strengthens the diagonal edges in image features. The last results in more accurate diagonal edges compared to the RCF and PiDiNet.

As a result of the literature review, the HED network implementing the correlation by prototype matching to detect edges and the PiDiNet network implementing the differential method were selected for robustness research. The combination of multi-scale convolutional kernels in the HED network architecture improves the robustness of edge detection by processing image details of different scales. The PiDiNet network architecture uses a combination of convolutional and separable convolutional layers with image gradient estimation to reduce edge positioning error. This research compares the edge detection performance of these two CNNs on noised natural images.

4. Materials and methods

To research the robustness of edge detection with HED and PiDiNet network, the technique of determining the influence of the noise level on the CNN edge detector efficiency is proposed. The stages of this technique are as follows.

1. As a result of literature review, CNNs are selected to detect edges on noisy images.
2. The types of image noise of interest to the researcher and the noise parameters whose values will be tuned by the researcher are determined. Mathematical modeling of noise images is performed.
3. The training and test sets of images from datasets for edge detection are selected.
4. The intervals of noise parameters for the research are determined. The images of test set are corrupted with controlled level noise.
5. The selected networks are learned to detect edges on training set of images or parameters of pre-trained networks are loaded.
6. The measures of edge detection performance are selected.
7. The trained networks are applied to noisy images and the edge detection performance is evaluated depending on the noise level in the images.
8. The obtained results are analyzed to generate the recommendations allowing to determine which CNN is better for edge detection when natural images affected by different level of white Gaussian noise, impulse noise, or multiplicative noise.

Then the literature is reviewed the proposed technique was not found by authors. If necessary, the elaborated technique of detecting the influence of the amount of noise on the CNN edge detector efficiency can be configured to research other CNNs with another known noise models. In the reviewed articles the processing of the noised images, ways to remove noise and improve image quality are considered, for example, using Gaussian filter, low-pass filters or wavelets [2]. The technique presented in this paper allows scientists to reasonably select the CNN architecture for edge detection, depending on the level of image noise, as well as to study the CNN in order to identify disadvantages and validate the obtained results.

Further the implementation of the technique of detecting the influence of the amount of noise on the CNN edge detector efficiency is considered. The first and second stages are discussed in this section. The following sections examine the remaining stages.

At the first stage the HED and PiDiNet networks are selected to detect edges on noisy images. The HED neural network is designed from several blocks of layers with side replies (Figure 1). Each layer block includes a convolutional layer, followed by a ReLU activation function, and then a pooling layer. As a result, image features at different scales are obtained with the side replies. After interpolation and concatenation, the side replies are combined into a single edge map [12].

The PiDiNet separable depth-wise convolutional structure has 4 stages with a shortcut for easy training. The max pooling layers are used among stages for down sampling. Each stage has 4 residual blocks including a depth-wise convolutional layer, a ReLU activation function, and a point-wise convolutional layer sequentially [14]. The exception is the first stage that has an initial convolutional layer and 3 residual blocks. The vanilla convolution in the depth-wise convolutional layers is replaced with pixel difference convolution in residual blocks (Figure 2). Any normalization layers are not included since the resolutions of the training images are not uniform.

At the second stage of determining the influence of the noise level on the CNN edge detector efficiency mathematical modeling of noisy images is performed. There are many types of noise affected on image processing [2, 17]. Gaussian noise, impulse noise, and speckle noise models are most often used when the image edge detection robustness is researched. These models allow reproducing the scenarios of real image corruption by controlled variation of the noise parameter. The noise values are considered as random variables determined by the probability density function. Then noise is a mask of pixels of random color and brightness superimposed on the image. Considering a certain class of noise as a function allows focusing on determining its parameters and the effect of these parameters on the image quality. The result of noise generation is a pixel mask with specified values of parameters, specifically, noise intensity and distribution. The most common are additive and multiplicative noise [18].

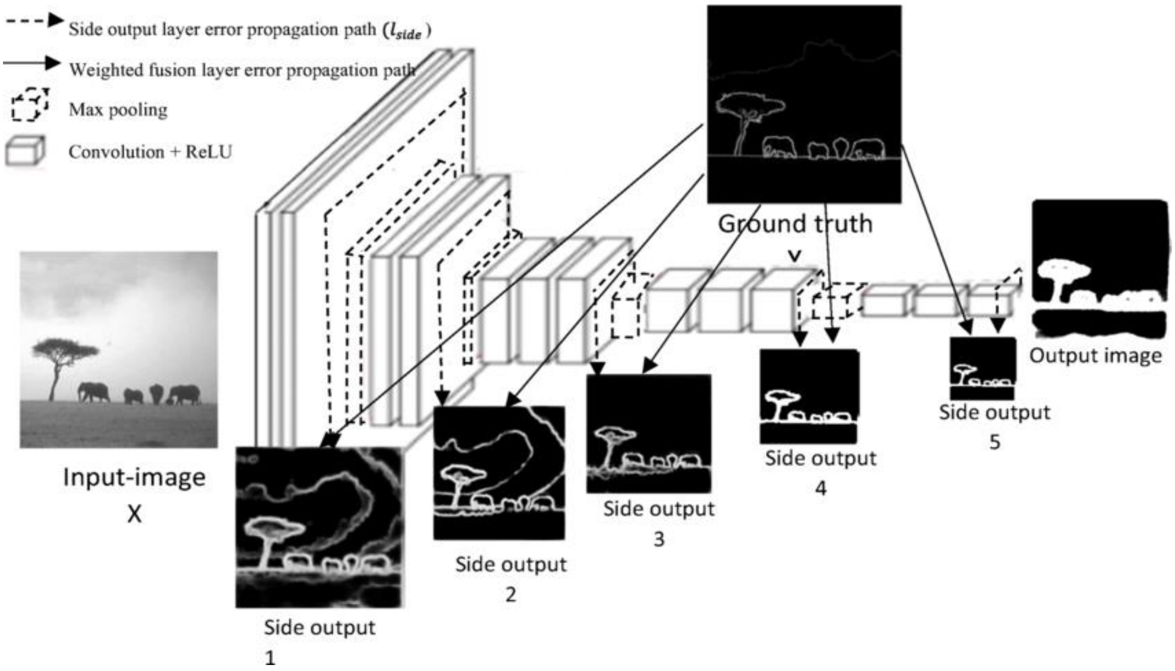

Figure 1: HED network architecture [9]

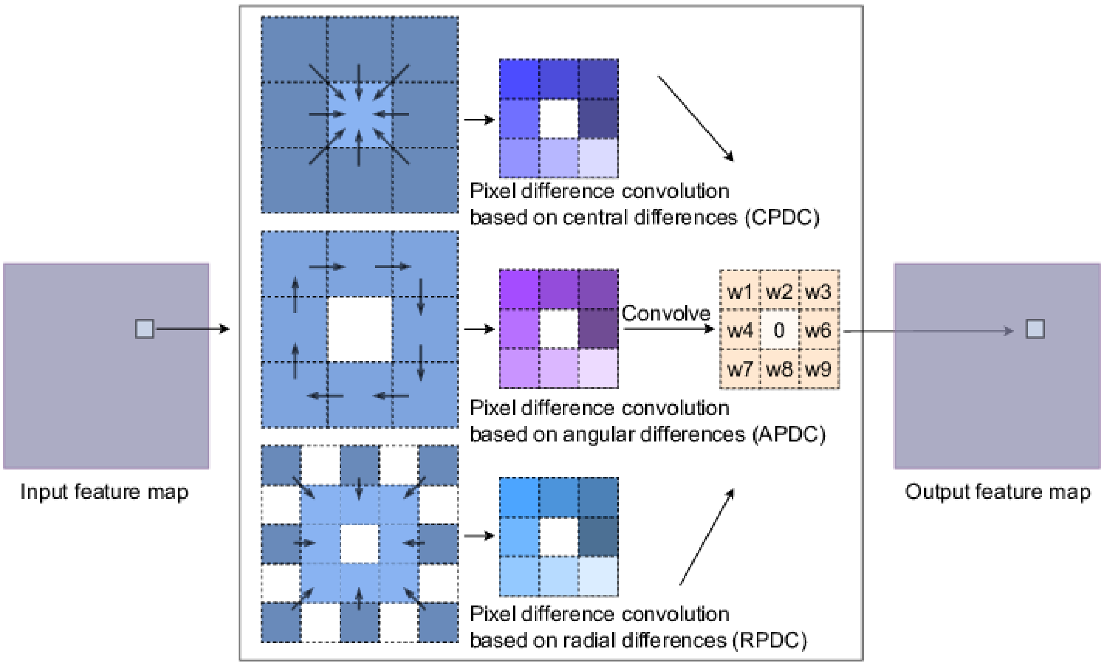
Pixel difference convolution
based on central differences (CPDC)
Convolve
w1
w2
w3
w4
0
w6
w7
w8
w9
Pixel difference convolution
based on angular differences (APDC)
Input feature map
Output feature map
Pixel difference convolution
based on radial differences (RPDC)

Figure 2: Pixel difference convolutions applied by PiDiNet network [14]

Additive noise predominantly determines the quality of natural images. It is assumed independent of the state of the system generating images. In most practical situations it is assumed to be Gaussian and have zero mean and standard deviation that is approximately constant over different color channels of the image. In addition, it is usually supposed that the values of additive noise are spatially uncorrelated, that is independent for neighboring pixels or pixels distant from each other [18].

Multiplicative noise model assumes dependence the noise on the state of the system generating images. Statistical characteristics of such noise depend on the content of the image. Often, the multiplicative noise variation is constant for image channels, since it is defined as the ratio of the local image variation to the square of the local mean [18].

At the same time, there are noises for which neither additive nor multiplicative models can be applied, for example, impulse noise. An image corrupted by impulse or salt and pepper noise will have dark pixels in light regions and light pixels in dark regions. This noise is typically caused by analog-to-digital conversion errors and bit errors while transmission [18].

The main reason for the appearance of Gaussian noise in the image is a lack of lighting while photographing. Another reasons for the Gaussian noise appearance are rather high temperature of the environment or the matrix of the device when long-term shooting. Impulse noise occurs because of image corruption while data transmission, or due to slight defects on the device's matrix. The cause of speckle noise can be the humidity of the environment, for example fog, as well as the use of wave lighting, which can be observed while laser scanning [2].

In statistical modeling of an image corrupted by white Gaussian noise, the latter is presented as (1) [17, 19]:

$$I_{K,gauss}(x,y) = I_K(x,y) + v_K(x,y), \tag{1}$$

where K means image channel which is red, green or blue; $I_{K,\ gauss}(x,\ y)$ is a noisy image channel; $I_K(x,\ y)$ is an original image channel; $v_K(x,\ y)$ is random additive noise in channel K. Here $v_K(x,\ y)$ is Gaussian distributed with m, s parameters. Here m is the mean; s is standard deviation of the noise.

When modeling impulse noise, the intensity of each point of the image channel is replaced with the probability Pa by the value a, and with the probability Pb is replaced by the value b. Additionally, the intensity of each point of the image channel is remained unchanged with the probability 1 – (Pa + Pb), where $Pa + Pb \leq 1$ [2]. Thus, the image corrupted with impulse noise is represented by the formulas (2) – (4):

$$I_{K,imp}(x,y) = a \text{ with probability } Pa, \tag{2}$$

$$I_{K,imp}(x,y) = b \text{ with probability } Pb, \tag{3}$$

$$I_{K,imp}(x,y) = I_K(x,y) \text{ with probability } 1-(Pa+Pb), \tag{4}$$

where $I_{K,\ imp}(x,\ y)$ is an image channel corrupted with impulse noise [2].

Noise impulses can be positive or negative. Image digitization usually involves scaling and intensity limiting. After digitization, noise impulse becomes extreme. This corresponds to the appearance of completely black and white points in the image, since the noise impulse intensity is usually greater than the value of the useful signal. Thus, the values of a and b are usually considered as "intense" in the sense that they are equal to the minimum and maximum values that can be present in a digitized image. Therefore, negative impulses appear as black dots on the image, and positive impulses appear as white dots on the image [2]. If one of the probability values is zero, then impulse noise is called unipolar.

Speckle noise is a multiplicative noise occurring much less frequently in natural images than additive white Gaussian noise. Speckle noise mostly affects ultrasound and laser imaging, and is also common in coherent imaging systems such as synthetic aperture radars and sonar. On natural images it appears as a result of unfavorable shooting conditions (humidity, poor lighting) [17, 18]. Speckle noise is non-uniform noise that makes images grainier. Its values often have Rayleigh distribution or gamma distribution. In the paper Gaussian distribution is used, and each pixel of the original image is multiplied by the noise value by the formula (5) [17, 19]:

$$I_{K,sp}(x,y) = I_K(x,y)(1 + hS_K(x,y)), \tag{5}$$

where $I_{K,\ sp}(x,\ y)$ is an image corrupted with speckle noise; $S_K(x,\ y)$ is a random noise with Gaussian distribution, $m = 0$, $s = 1$; h is a multiplicative noise intensity.

The spatial characteristics of noise, as well as the question of whether there is a relationship between noise and image, are crucial for further research. We will assume that the noise distribution is not related to the image and does not depend on the spatial coordinates. But such assumptions may be at least unfounded in some cases. Examples are images obtained in situations with few quanta, such as X-ray and nuclear imaging, which are not considered in this paper.

5. Experiments

In this section the implementation of the proposed technique from the third to the sixth stage of detecting the impact of noise on the CNN edge detector efficiency is considered.

At the third stage of research of the robustness of the edge detection with the CNNs the natural images from the BSDS300 dataset are considered [20]. This dataset contains a total of 300 images with different complexity of scenes and textures, including 200 training images and 100 test images. The ground-truth images are also presented which are binary images with edges selected by 5-8 experts (edge maps) [20]. The performance of edge detection was evaluated by comparing edge maps obtained using the researched CNNs, with ground-truth images.

The fourth stage of the proposed technique is implemented as follows. To provide real-world cases of the noised images, the original images of the researched dataset are corrupted by one of various types of noise, such as Gaussian, impulse and speckle. The level of noise is controlled to obtain a set of images with noise parameters from minimum to maximum. This allows for a more detailed analysis of the noise effect on edge detection.

At the fifth stage of the proposed technique parameters of the pre-trained HED and PiDiNet networks are loaded to detect edges on the controlled noised images [21, 22]. The HED network was implemented using the publicly available Caffe Library and pre-trained on BSDS500 dataset [23] with the following hyper-parameters. These are mini-batch size (10), learning rate (1e-6), loss-weight for each side-output layer (1), momentum (0.9), initialization of the fusion layer weights (0.2), number of training iterations (10,000; divide learning rate by 10 after 5,000) et al. [21].

Implementation of pre-trained PiDiNet is based on the PyTorch library [22]. In detail, this network is randomly initialized and trained for 14 epochs with Adam optimizer. The initial learning rate was 0.005, which was decayed in a multi-step way (at epoch 8 and 12 with decaying rate 0.1). The values of other hyper-parameters are indicated in [14].

At the sixth stage of the proposed technique to evaluate the edge detection results the Precision Pr, Recall Rc, F1-score F_1 were used [1, 3]. They are represented by the formulas (6), (7):

$$F_1 = 2\,Pr\,Rc\,/\,(Pr + Rc), \tag{6}$$

$$Pr = TP/(TP + FP), Rc = TP/(TP + FN), \tag{7}$$

where TP is a number of correctly detected edge pixels; FP is a number of background pixels that are incorrectly detected as edge pixels; FN is a number of edge pixels that are incorrectly detected as background pixels.

Precision is the fraction of relevant edge pixels detected on original image among the edge pixels on related ground-truth image. Recall is the fraction of relevant edge pixels among the all edge pixels detected on original image. F1-score is calculated as the harmonic mean of precision and recall. This is particularly useful in situations where false negatives and false positives are equally significant. In addition, the FOM value was used to evaluate the edge detection results [6]. There are three common errors associated with edge detectors: (1) missing valid edges, (2) inaccurate edge positioning, (3) classification of noise fluctuations as edges. Pratt has introduced a FOM that balances these three types of error, but mainly characterizes the error in the edge positioning [6]. The Pr, Rc, F_1, FOM are varied from 0 to 1 and equal one for a well detected edge.

6. Results

In this section the implementation of the proposed technique from the seventh to the eighth stage of detecting the impact of noise on the CNN edge detector efficiency is considered. For images corrupted by white Gaussian noise, impulse noise, and speckle noise graphs of the edge detection performance measures on the noise level are shown in Figures 3, 4, 5 respectively. Orange line corresponds to HED network, gray line corresponds to PiDiNet network.

The results of edge detection by the HED and PiDiNet networks were researched for values of the Gaussian noise parameter σ from 0 to 45 in increments of 5. The probability of pixels corrupted by the impulse noise changed from 0.01 to 0.18 in increments of 0.02. The intensity of the speckle noise changed from 0.01 to 0.27 in increments of 0.03. The parameter ranges were obtained experimentally. It should be noted that images with a higher noise parameter are also allowed, but in the realities of natural images, such images are considered too corrupted and are not subject to further research. Figure 6 shows the noised BSDS300 image with low and high levels of noise. The image corrupted by Gaussian noise with standard deviation 20 and 40 is presented on Figure 6, a, d. The image corrupted by impulse noise with probability of corrupted pixels 0.08 and 0.016 is shown on Figure 6, b, e. The image corrupted by speckle noise with intensity 0.25 and 0.5 is presented on Figure 6, c, f.

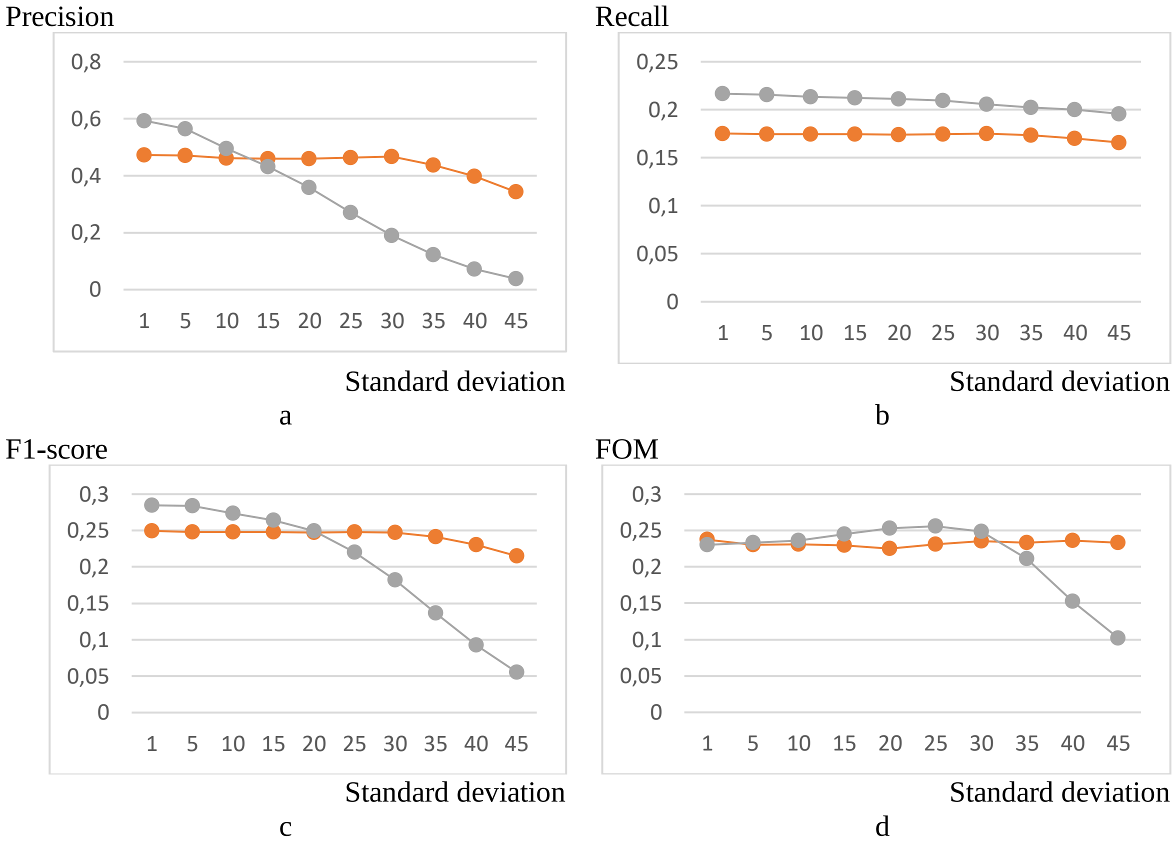

Figure 3: Graphs of dependence from the Gaussian noise parameter: a – Precision; b – Recall; c – F1; d – FOM; orange line corresponds to HED, gray line corresponds to PiDiNet network

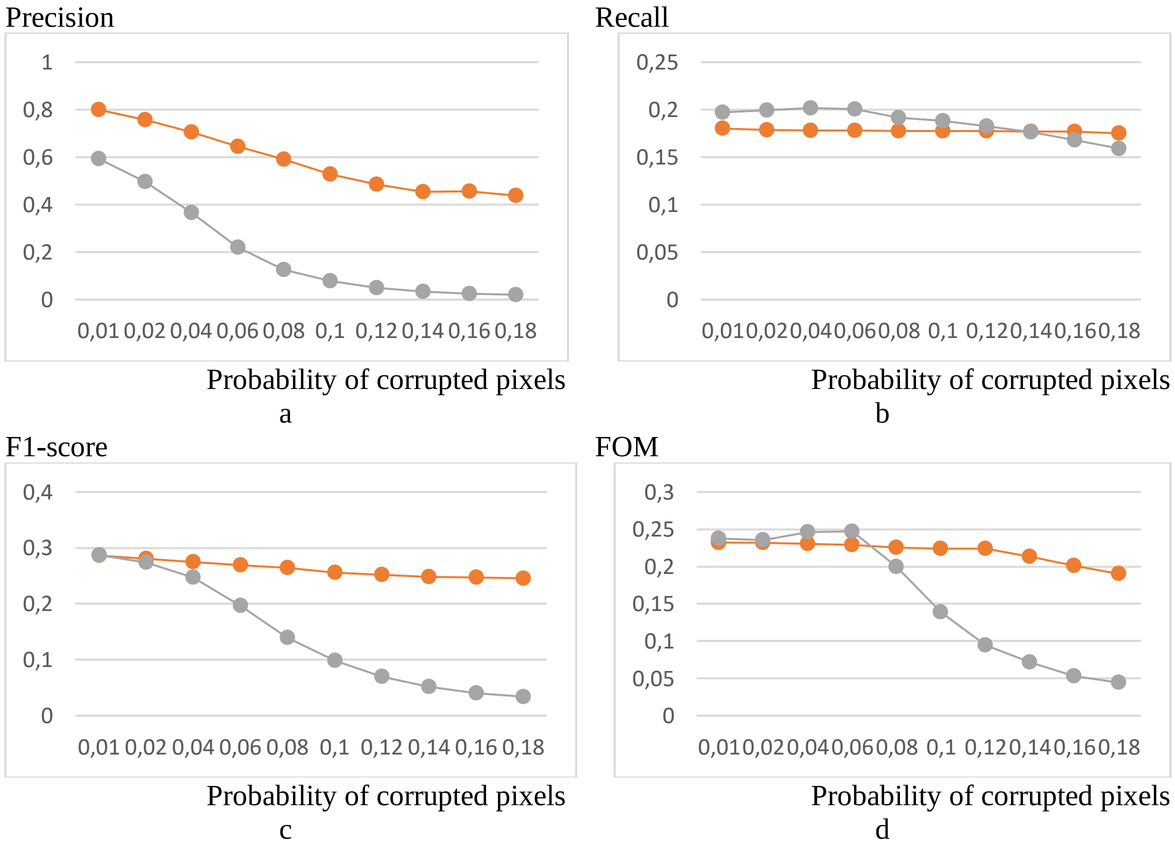

Figure 4: Graphs of dependence from the impulse noise parameter: a – Precision; b – Recall; c – F1; d – FOM; orange line corresponds to HED, gray line corresponds to PiDiNet network

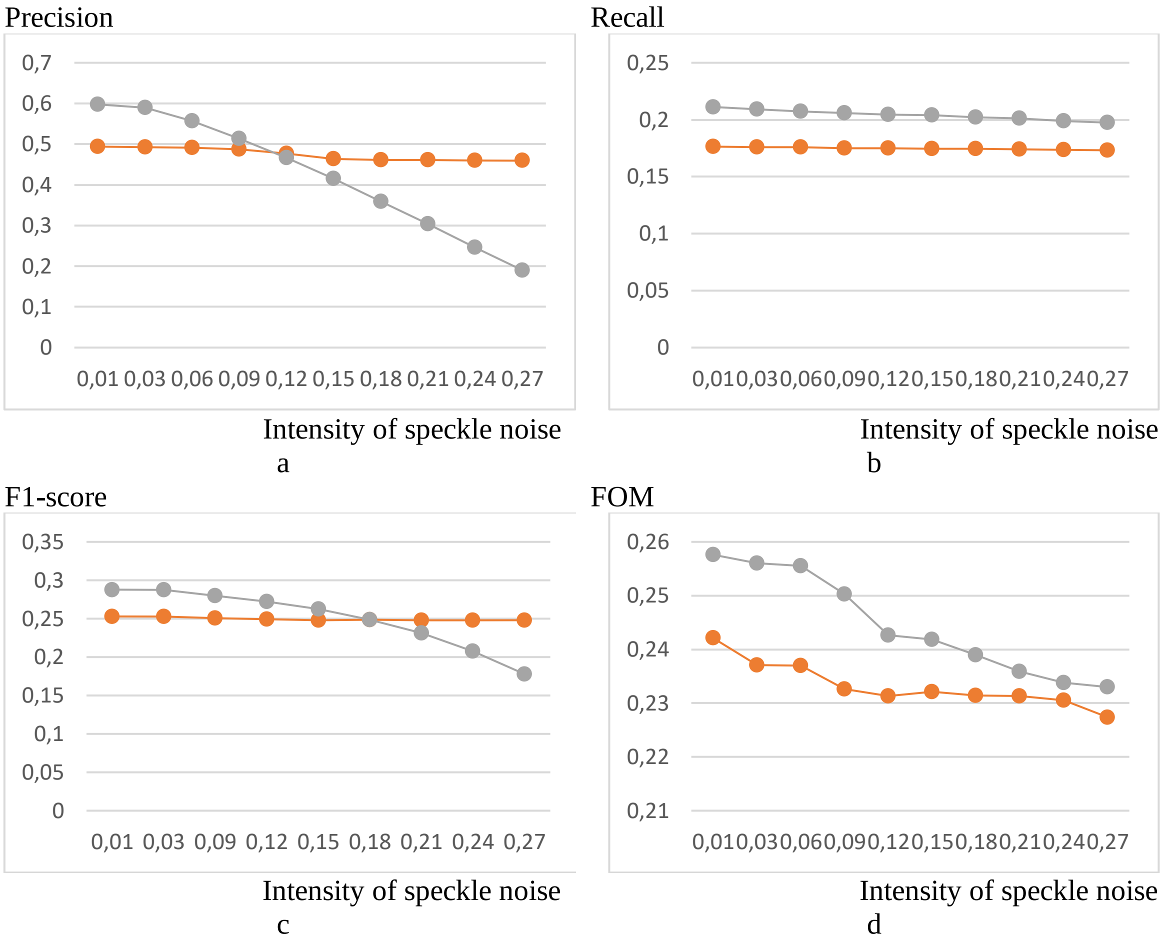

Figure 5: Graphs of dependence from the speckle noise parameter: a – Precision; b – Recall; c – F1-score; d – FOM; orange line corresponds to HED, gray line corresponds to PiDiNet

Also, the edge maps obtained by HED and PiDiNet networks are shown. The key feature of CNNs is altering the scale of abstraction while image processing. This can have a negative effect on edge detection, because the use of abstraction levels always leads to the loss of fine details and the accumulation of the strong edges. This is typical for edge detection by HED network (Figure 6, g–l).

The edge detection results obtained with PiDiNet network, on the contrary, includes weak and false edges because image gradient estimation applied with pixel difference convolution (Figure 6, m–r). Then the image noise level is increased the edge detection quality is reduced. The high level of impulse or speckle noise specially impacts on edge detection results (Figure 6, k, l, q, r), for which missing valid edges is observed. The detected edges are blurred.

7. Discussions

Analyzing the obtained results, we notice that the Precision and F1-score decreases almost proportionally when the Gaussian noise standard deviation increases (Figure 3, a, c). At first Precision and F1-score were taken into account. Then the edge detection performance of the PiDiNet network is up to 1.3 times higher than of the HED network if the standard deviation of the noise does not exceed 15. Otherwise, on the contrary, the edge detection performance of the HED network is up to 7 times higher than of the PiDiNet network.

Analyzing the obtained values of the Recall, notice that the PiDiNet network is up to 1.3 times more robust to Gaussian noise in images than the HED (Figure 3, b). This is due to the fact that ground-truth edges are quite thin, while the CNNs process images at several levels of abstraction. At high levels of abstraction there is always a loss of fine details, which generalizes and thickens the edges. The Pratt's FOM estimates the distance between ground truth edge and detected edge. Then for images corrupted by white Gaussian noise, the results of edge detection by the HED and PiDiNet networks showed almost similar FOM values for *s* less than 35 (Figure 3, d). For *s* greater than 35, the FOM of the HED network is up to 2 times higher than the FOM of the PiDiNet.

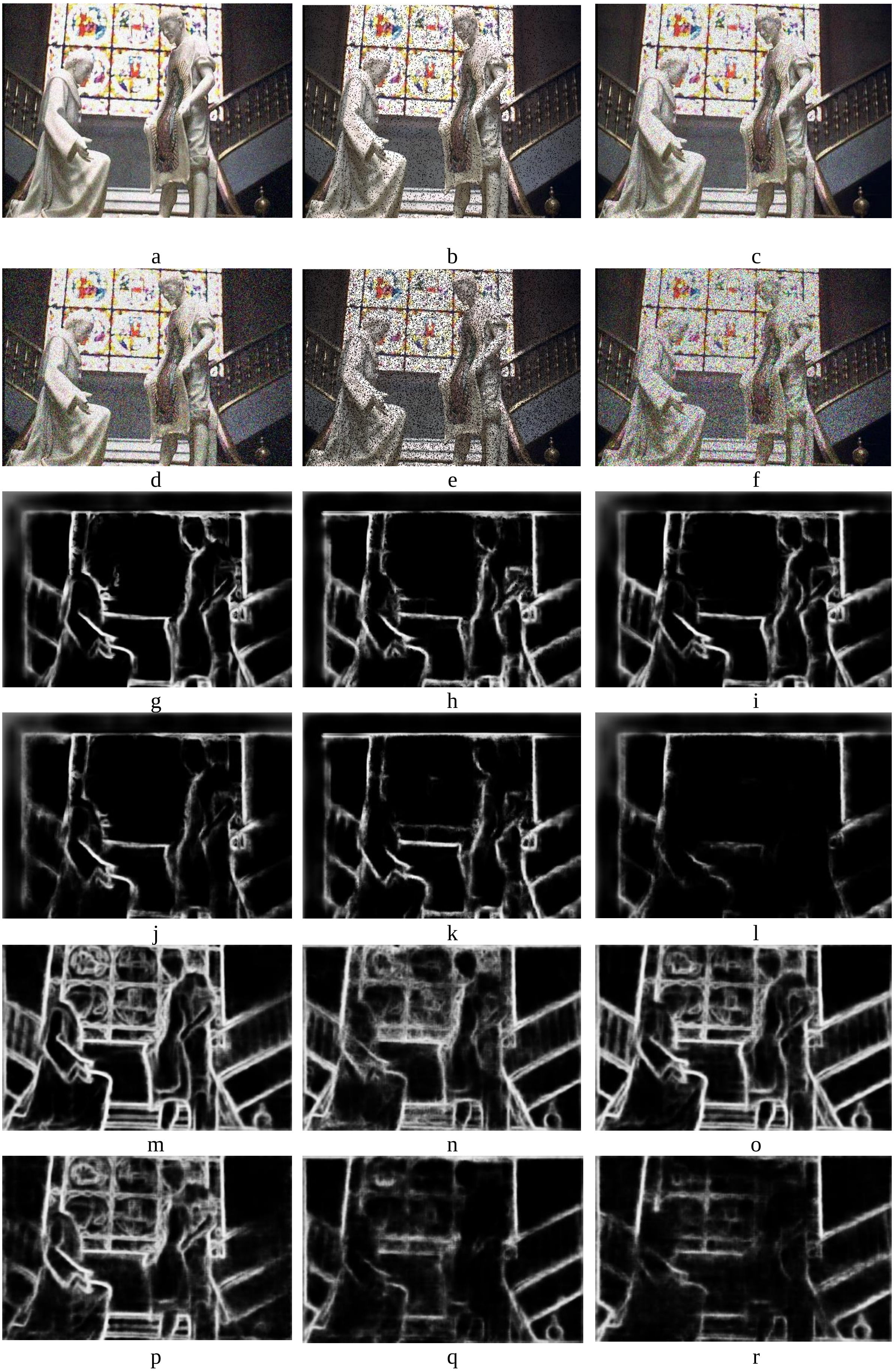

Figure 6: The edge detection results of noised images: a, d – the image corrupted by Gaussian noise with standard deviation 20 and 40; b, e – the image with probability of corrupted pixels 0.08 and 0.016; c, f – the image corrupted by speckle noise with intensity 0.25 and 0.5; g–l – edge maps

obtained by HED; m–r – edge maps obtained by PiDiNet

For images corrupted by impulse noise, the results of edge detection by HED and PiDiNet networks showed almost similar FOM values for probability of corrupted pixels less than 0.08 (Figure 4, d). For probability of corrupted pixels of 0.08 or more, the HED outperforms the PiDiNet in terms of FOM by up to a factor 4. For images corrupted by speckle noise, the PiDiNet network outperforms the HED in terms of FOM up to 1.1 times over the entire considered range of speckle noise intensity (Figure 5, d). This may be because the speckle noise can reduce the image contrast. Since the HED uses a multi-scale approach when detecting edges, an altering of pixel contrast negatively affects the network when edges are detected.

For images corrupted by speckle noise the edge detection performance on the noise level (Figure 5) are similar to the corresponding dependences for Gaussian noise (Figure 3). The Precision and F1-score decreases almost proportionally with increasing speckle noise intensity. In view of the Precision and F1-score, the quality of edge detection using the PiDiNet is up to 1.2 times higher than that of the HED if the speckle noise intensity does not exceed 0.15. Otherwise, the quality of edge detection using the HED is up to 2.3 times higher than that of the PiDiNet network. With the Recall, the PiDiNet is more robust to speckle noise than HED. Speckle noise can affect the contrast of the image, as well as create low-contrast edges, which complicates their detection. In the case the PiDiNet processed edges better than the HED. If for images corrupted by impulse noise the Precision and F1-score is estimated then the quality of edge detection by the HED is up to 5 times higher than that by the PiDiNet over almost the entire considered range of probability of corrupted pixels.

The Recall evaluates the ratio of correctly detected edges relative to all edges detected by the network. Low similar recall on the considered range of impulse noise parameter values is due to the architectures of the HED and PiDiNet. These networks can determine false edges when the level of abstraction is altered.

Since the F1-score evaluates the overall efficiency of the edge detector, its significance increases in the case of ambiguity of the Recall and Precision. In our case the F1-score confirms the advantage of HED over PiDiNet in edge detection on impulse noised images.

As a result of the analysis of the obtained results, recommendations were formulated to select the CNN for edge detection at a given or estimated noise level (Table 1). In particular, the HED is generally preferred at high noise levels. The PiDiNet is best used when the noise level is low. This indicates the weak robustness of the PiDiNet and the need for pre-processing of images when edges are detected by this network. The HED network has better values of Precision, F1 and FOM at significant noise levels.

Table 1
The recommendation on selection the CNN for edge detection at a given or estimated noise level

Type of noise	Parameter range	Recommended CNN
White Gaussian noise	[1, 20]	PiDiNet
	(20, 45]	HED
Impulse noise	[0.01, 0.06]	PiDiNet
	(0.06, 0.18]	HED
Speckle noise	[0.01, 0.15]	PiDiNet
	(0.15, 0.30]	HED

8. Conclusions

The actual scientific and applied problem of the robustness of the edge detection on noisy images by CNNs has been researched. The scientific novelty is the technique of determining the effect of the image noise level on the CNN edge detector efficiency which is proposed for the first time. When reviewing the literature devoted to the edge detection using neural networks, no similar techniques with the controlled noising of the image and subsequent evaluation of the edge detection performance were found.

The practical significance of obtained results is that to research the robustness of the edge detection by HED and PiDiNet networks the experiments have been conducted. The recommendations on robust CNN selection are formulated. The experimental results allow scientists to reasonably select the CNN architecture for edge detection depending on the level of image noise.

Prospects for further research are application of the elaborated technique to study the impact of mixed noise on the edge detection performance using CNNs. It is possible to complicate the representation of multiplicative noise by assuming gamma or Rayleigh distribution [24]. In a similar way, one can research the influence of the noise level on the effectiveness of solving other problems using CNNs, specifically, image segmentation, and identification or classification objects on images.

References

[1] R. Sun, T. Lei, Q. Chen et. al., Survey of image edge detection, Frontiers of Signal Processing 3 (2022). doi:10.3389/frsip.2022.826967.

[2] R. C. Gonzalez, R. E. Woods, Digital Image Processing, 4th ed., Pearson, New York, NY, 2017.

[3] M. Polyakova, RCF-ST: Richer Convolutional Features network with structural tuning for the edge detection on natural images, Radio electronics, computer science, management 4 (2023) 122–134. doi:10.15588/1607-3274-2023-4-12.

[4] M. S. J. Rogers, M. Bithell, S. Brooks, T. Spencer, VEdge_Detector: automated coastal vegetation edge detection using a convolutional neural network, International Journal of Remote Sensing 42.13 (2021) 4809–4839. doi:10.1080/01431161.2021.1897185.

[5] J. Jing, S. Liu, G. Wang, W. Zhang, C. Sun, Recent advances on image edge detection: A comprehensive review, Neurocomputing 503 (2022) 259–271. doi:10.1016/j.neucom.2022.06.083.

[6] C. Akinlar, C. Topal, ColorED: color edge and segment detection by edge drawing, Journal of Visual Communication and Image Representation 44 (2017) 82–94. doi:10.1016/j.jvcir.2017.01.024.

[7] M. Polyakova, V. Krylov, N. Volkova, The methods of image segmentation based on distributions and wavelet transform, in: Proceedings of Data Stream Mining & Processing: IEEE First International Conference, DSMP, IEEE, Lviv, Ukraine, 2016, pp. 243–247. doi:10.1109/DSMP.2016.7583550.

[8] S. Xie, Z. Tu, Holistically-nested edge detection, International Journal of Computer Vision 125.5 (2017) 1–16. doi:10.1007/s11263-017-1004-z.

[9] B. Ray, S. Mukhopadhyay, S. Hossain, S. Ghosal, R. Sarkar, Image steganography using deep learning based edge detection, Multimedia Tools and Applications 80.24 (2021) 33475–33503. doi:10.1007/s11042-021-11177-4.

[10] Y. Liu, M. M. Cheng, X. Hu, K. Wang, X. Bai, Richer convolutional features for edge detection, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, CVPR, IEEE, Honolulu, HI, USA, 2017, pp. 5872–5881. doi:10.1109/CVPR.2017.622.

[11] Deng, C. Shen, S. Liu, H. Wang, X. Liu, Learning to predict crisp boundaries, in: Proceedings of 15th European Conference on Computer Vision, ECCV, IEEE, Munich, 2018, VI, pp. 562–578. doi:10.1007/978-3-030-01231-1_35.

[12] R. Grompone von Gioi, G. Randall, A brief analysis of the holistically-nested edge detector, Image Processing On Line 12 (2022) 369–377. DOI:10.5201/ipol.2022.422.

[13] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: Proceedings of 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 2015. URL: https://arxiv.org/pdf/1409.1556. doi:10.48550/arXiv.1409.1556.

[14] Z. Su, W. Liu, Z. Yu et al., Pixel difference networks for efficient edge detection, in: Proceedings of IEEE/CVF International Conference on Computer Vision, ICCV, IEEE, online, 2021, pp. 5117–5127. doi:10.1109/ICCV48922.2021.00507.

[15] Y. Li, X. S. Poma, G. Li et al., PiDiNeXt: an efficient edge detector based on parallel pixel difference networks, in: Proceedings of 6th Chinese Conference on Pattern Recognition and Computer Vision, PRCV, Springer-Verlag, Xiamen, China, 2023, pp. 261–272. doi:10.1007/978-981-99-8549-4_22.

[16] Z. Yin, Z. Wang, C. Fan, X. Wang, T. Qiu, Edge detection via fusion difference convolution, Sensors, 23.15 (2023). doi:10.3390/s23156883.

[17] P. Arulpandy, M. Trinita Pricilla, Speckle noise reduction and image segmentation based on a modified mean filter, CAMES 27.4 (2020) 221–239. doi: 10.24423/cames.290.

[18] N. B. Shakhovska, O. I. Kosar, Analysis of common methods of noise overlaying on images, Scientific Bulletin of UNFU 28.1 (2018) 145–149. doi: 10.15421/40280129

[19] M. Uss, B. Vozel, V. Lukin, K. Chehdi, Comparison of learning-based and maximum-likelihood estimators of image noise variance for real-life and synthetic anisotropic textures, in: Proceedings of Image and Signal Processing for Remote Sensing XXVI, SPIE, Edinburgh, United Kingdom, 2020, pp. 153303. doi: 10.1117/12.2573934.

[20] Berkeley Segmentation Dataset 68. Kaggle. URL: https://www.kaggle.com/code/mpwolke/berkeley-segmentation-data-set-68.

[21] Code for Holistically-Nested Edge Detection. URL: https://github.com/s9xie/hed.

[22] Code for the ICCV 2021 paper "Pixel Difference Networks for Efficient Edge Detection" (Oral). URL: https://github.com/hellozhuo/pidinet.

[23] The Berkeley Segmentation Dataset and Benchmark Web site, 2019. URL: https://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds.

[24] M. Uss, V. Lukin, B. Vozel, K. Chehdi, Accuracy of model-based and learning-based approaches for image noise variance estimation, in: Proceedings of Ukrainian Microwave Week, UkrMW, IEEE, Kharkiv, Ukraine, 2020, pp. 438–442. doi: 10.1109/UkrMW49653.2020.9252696.