sensors

Article

Noise Immunity and Robustness Study of Image Recognition Using a Convolutional Neural Network

Vadim Ziyadinov * and Maxim Tereshonok

Science and Research Department, Moscow Technical University of Communications and Informatics, 111024 Moscow, Russia; m.v.tereshonok@mtuci.ru
* Correspondence: v.v.ziyadinov@mtuci.ru

Abstract: The problem surrounding convolutional neural network robustness and noise immunity is currently of great interest. In this paper, we propose a technique that involves robustness estimation and stability improvement. We also examined the noise immunity of convolutional neural networks and estimated the influence of uncertainty in the training and testing datasets on recognition probability. For this purpose, we estimated the recognition accuracies of multiple datasets with different uncertainties; we analyzed these data and provided the dependence of recognition accuracy on the training dataset uncertainty. We hypothesized and proved the existence of an optimal (in terms of recognition accuracy) amount of uncertainty in the training data for neural networks working with undefined uncertainty data. We have shown that the determination of this optimum can be performed using statistical modeling. Adding an optimal amount of uncertainty (noise of some kind) to the training dataset can be used to improve the overall recognition quality and noise immunity of convolutional neural networks.

Keywords: neural networks; pattern recognition; noise in imaging systems; robustness; training dataset; noise immunity

Citation: Ziyadinov, V.; Tereshonok, M. Noise Immunity and Robustness Study of Image Recognition Using a Convolutional Neural Network. *Sensors* **2022**, *22*, 1241. https://doi.org/10.3390/s22031241

Academic Editor: Christophoros Nikou

Received: 24 December 2021
Accepted: 5 February 2022
Published: 6 February 2022

1. Introduction

Deep learning and big data analytics are, nowadays, important fields in computational science. Various organizations face the necessity of bringing these areas into their work processes to keep up with current trends. Deep learning neural networks can identify the most complex patterns in the data quickly and efficiently at high levels of abstraction, while those patterns are not observed in the first approximation. Information from machine learning applications can deepen the understanding of many hidden processes, as well as solve problems of prediction and automation in many areas of life, such as speech recognition [1], computer vision [2,3], and data visualization [4].

Automatic pattern and image recognition technologies have the widest applications in image processing. The use of convolutional neural networks (CNNs) increases its success for image processing, character and handwritten text recognition [5], license plate recognition [6], human, plant, and animal pathology detection [7,8], face and emotion recognition [9,10], distinguishing objects of interest in a video stream [11], etc.

Most publications deal with new datasets from various problem domains [7,8,12]. Many publications are concerned with neural network topology and learning technique improvements [13]. However, there are many unsolved problems in image recognition tasks. First, *the recognition accuracy is sometimes poor or insufficient*. False diagnoses made by a neural network, while not being a big problem nowadays (since the data obtained from the network are verified by the operator), can be a barrier toward expanding the impact of automatic recognition algorithms in the future. The same might be said for automatic driving systems, such as automobile autopilots.

Second, the results of neural networks are affected by data distortions, such as adversarial attacks [14,15]. Figure 1 shows an example of this kind of attack: an image of a dog with added (invisible to the human eye) noise is recognized by the network as an image of a submarine with a large coefficient of confidence. Thus, *images with different amounts of distortions are recognized with different accuracies*.

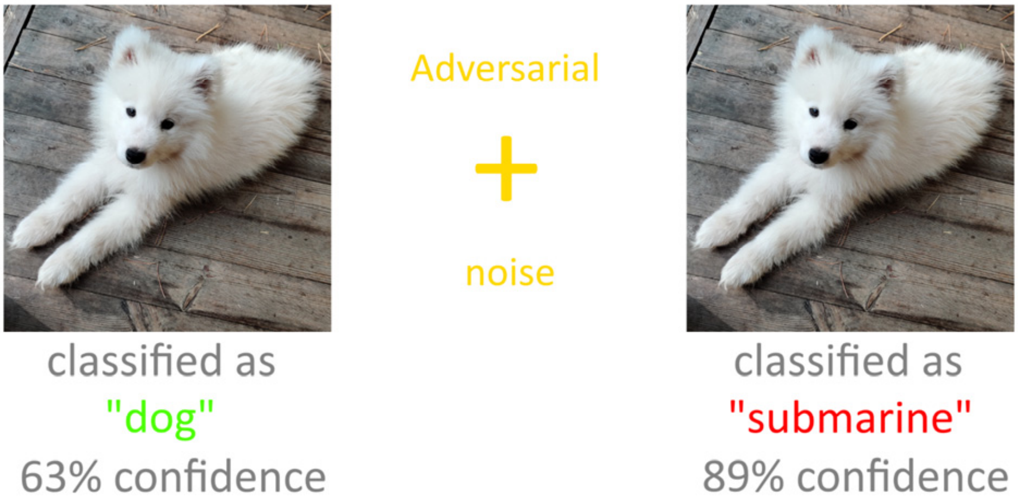

Figure 1. An example of an adversarial attack on a convolutional neural network.

Third, *there is no universal approach to estimate the optimality and robustness* of a trained neural network. It cannot be predicted in advance how the trained neural network will behave when new data are obtained, and we cannot be unequivocally sure that the network will correctly recognize new data, especially if statistical characteristics of new data differ from those of the data used for training.

Recent publication analyses showed that robustness studies are performed only in terms of precision–recall curve plotting [16]. Some publications deal with the estimation of adversarial attack success rates rather than with increasing noise immunity of the neural networks [17]. Recent works concerning uncertainty quantifications in neural networks have not provided solutions for increasing neural network noise immunity [18]. A noise immunity study of a neural network is still "out of consideration". However, such an investigation seems to be the key to solve adversarial attack problems and to improve the robustness and correctness of recognition of various data by neural networks.

The most important aspect in the neural network application is training, and its success mainly depends on the correct training data representation. Complex and large neural networks trained on poorly represented data, in addition to their high resource consumptions, turn out to be much poorer than simple networks that are trained on correct and well-organized data [19,20]. As a result, feature construction—the correct feature generation process (for the training data)—is the most important part of machine learning.

Correct and deep feature construction involves the potential of the CNN's generalization ability, which can (and should) be utilized to withstand noise added to the recognized data and even adversarial attacks. Noise-immune CNN should gather the information from its layers, where the noise influence on the detectable features is quite low. For example, to fight high-frequency noise, the CNN should utilize the features obtained in its deep convolutional layers, where input data are averaged throughout large regions of the original image. We suppose that training CNNs using noisy data can help to improve their generalization abilities and capabilities of withstanding adversarial attacks.

In our study, we strived to use an easy-to-understand example to investigate the neural network behavior without loss of generality. The first example was chosen for its clarity: the amount of noise (here represented as point location uncertainty) can be estimated visually. Images with low point density (Figure 2) are convenient to distinguish distortions from the object shape.

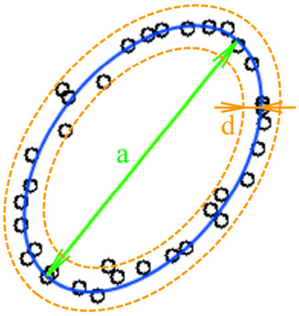

Figure 2. An example of the low-density image.

Figure 2 shows an example of the low-density image. The ideal figure is an ellipse; one can see the deviations of point placements from the perfect ellipse. These deviations are considered as noise or uncertainty. In this paper, we analyze the low-point density image recognition accuracy as a function of the amount of point location uncertainty (further: uncertainty) in the training and test datasets. We also define a way to determine the optimal training parameters.

The research method consists in generating a dataset with pseudo-random low-density point images with different uncertainties and then analyzing the recognition accuracy of these datasets by a trained convolutional neural network, as well as determining the optimal dataset parameters for convolutional neural network training to obtain the best recognition accuracy results.

Robustness estimation of low-density image recognition was investigated in [21]. The robustness study was performed in application to signature verification and resulted in obtaining the miss probability vs. false alarm probability charts with fixed uncertainty. The noise influence on low-density image recognition was also researched in [22]. In both papers, the influence of the training and testing dataset uncertainty on the recognition results was not investigated. The parameter defining the uncertainty measure can be described as $\boldsymbol{U} = \boldsymbol{d}/\boldsymbol{a}$, where $\boldsymbol{d}$—variance and $\boldsymbol{a}$—linear size of the figure (Figure 2). Further, we consider the training dataset uncertainty as $\boldsymbol{U_{TR}}$ and the testing dataset uncertainty as $\boldsymbol{U_{TS}}$. In other tasks, such as noisy image recognition shown in Figure 1, $\boldsymbol{U}$ can be described as

$$\boldsymbol{U} = \boldsymbol{I_{noise}}/\boldsymbol{I_{info}}, \tag{1}$$

where $\boldsymbol{I_{noise}}$ is the average intensity of noise and $\boldsymbol{I_{info}}$ is the average intensity (size) of the meaningful part of the recognized image [17]. In both cases, uncertainty $\boldsymbol{U}$ describes the ratio of the noisy component of the image to the informative component of the image. The addition of noise of any matter and level can drastically decrease the performance of recognition for a neural network trained on an ideal dataset, so the robustness study should be conducted to avoid this effect by altering the training dataset properties.

The low point density image recognition tasks have already been researched by the authors in a series of papers dedicated to the estimation and prediction of behavior of mobile communication network subscriber groups and complex clusters by analyzing teletraffic and geolocation data [23–25]. The locations of subscribers in groups can be considered as sparsely located points on images. Convolutional neural networks have shown their effectiveness in solving this task [26], but the analysis and justification of the applied method's stability on the initial data uncertainty have not been conducted so far. In [23], the mathematical model describing the subscribers' cluster shapes was implemented, and it was shown that typical cluster shapes, representing images with low point density, can also be automatically classified by convolutional neural networks.

The exhaustive characteristics of the proposed method stability have not been obtained so far, but there is a good basis to suppose that different uncertainties in datasets can significantly distort the forms, representing subscriber groups, and complicate the recognition of these forms. Thus, the evaluation and optimization of the convolutional neural network robustness in solving the recognition tasks require specific statistical modeling. This task presents a great theoretical interest, since the results of this paper can be applied to all fields of machine learning—the work describes a largely simplified model, thereby summarizing its conclusions for most cases solved by neural networks and other machine learning models.

2. Research Plan

In a convolutional neural network, the overall supervised learning and inference system structure is as shown in Figure 3.

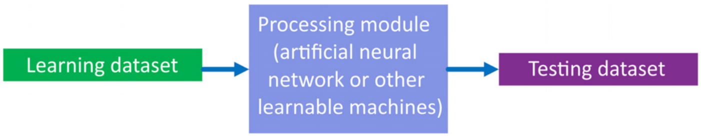
Learning dataset
Processing module (artificial neural network or other learnable machines)
Testing dataset

Figure 3. Common structure of supervised learning and inference system.

As stated before, the optimization of the neural network itself, without taking the training dataset influence into account, will generally not provide an exhaustive result and will only allow improving the behavior of the system in some cases. Although the deepening of a neural network (generally, but not applicable in this work because of the simplicity of images, which will be shown further) leads to a better ability of this network to detect and generalize the hidden features [26–28], but also creates many problems, such as network resource consumption increase, vanishing gradients problem, etc. Optimization must be performed, taking the properties of all "training dataset-processing module-testing dataset" system components into account.

A widespread traditional approach involves obtaining the fixed test dataset recognition accuracy by a network trained on a fixed training dataset with a given uncertainty $\boldsymbol{U_0}$. This accuracy can be described as a single number, a scalar $\boldsymbol{P_0}$. Accuracy $\boldsymbol{P_0}$ is described as follows:

$$P_0 = \frac{M_{correct}}{M_{total}}, \tag{2}$$

where $\boldsymbol{M_{correct}}$ is the number of correctly recognized items in the testing dataset and $\boldsymbol{M_{total}}$ is a total number of items in the testing dataset.

This scalar approach only allows estimating the local properties of the learning-recognition system, but it does not allow estimating the behavior of this system at different data uncertainties. We propose a deeper vector and matrix approach to evaluate the network stability and robustness, which includes the following two sequential steps:

(1) Obtaining an array of test dataset recognition accuracies $\boldsymbol{P}$ at various test dataset uncertainties $\boldsymbol{U_{TS}}$ with a fixed training dataset uncertainty $\boldsymbol{U_{TR}}$—vector $\boldsymbol{P(U_{TS})}$.

(2) Obtaining a two-dimensional array of testing dataset recognition accuracies $\boldsymbol{P}$, depending on their uncertainties and on training dataset uncertainties $\boldsymbol{U_{TR}}$ at the same time—matrix $\boldsymbol{P}(\boldsymbol{U_{TR}}; \boldsymbol{U_{TS}})$.

Thus, there is an increase in the informativity concerning learning-recognition system robustness and optimality estimation at each following step. With known $\boldsymbol{P}(\boldsymbol{U_{TR}}; \boldsymbol{U_{TS}})$, we can obtain $\boldsymbol{P}(\boldsymbol{U_{TS}})$ and $\boldsymbol{P_0}$:

$$P(U_{TS}) = \frac{1}{N_{TR}} \cdot \sum_{U_{TR}} P(U_{TR}; U_{TS}); \tag{3}$$

$$P_0 = \frac{1}{N_{TS}} \cdot \sum_{U_{TS}} P(U_{TS}), \tag{4}$$

where $\boldsymbol{N_{TR}}$ is the amount of various learning dataset uncertainties $\boldsymbol{U_{TR}}$, $\boldsymbol{N_{TS}}$ is the amount of various testing dataset uncertainties $\boldsymbol{U_{TS}}$.

A convenient image experiment model was chosen to evaluate the external specifications of the learning and inference system. The mathematical model described in paper [23] allows us to automatically generate the datasets used for convolutional neural network training and testing, and it also allows us to set various uncertainty parameters (for example, the point position offset relative to the shape vector model (Figure 4). This fact allows evaluating the convolutional neural network stability to the changing dataset uncertainty parameters and to evaluate the neural network characteristics in the conditions of the factors, increasing the input data distortions. This fact allows evaluating the convolutional neural network stability to the changing dataset uncertainty parameters and evaluating the neural network characteristics in the conditions of the presence of the factors, increasing the input data distortions.

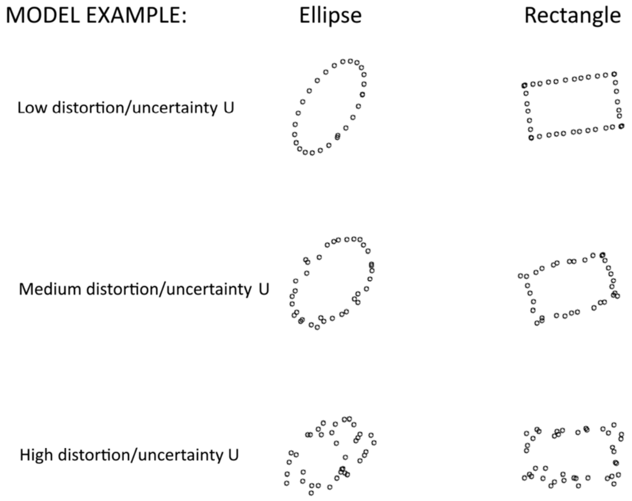
MODEL EXAMPLE:
Ellipse
Rectangle
Low distortion/uncertainty U
Medium distortion/uncertainty U
High distortion/uncertainty U

Figure 4. Low-density image model with different point location uncertainties.

To evaluate the trained neural network noise immunity characteristics in paper [23], we generated 200 datasets with different uncertainties (Figure 5).

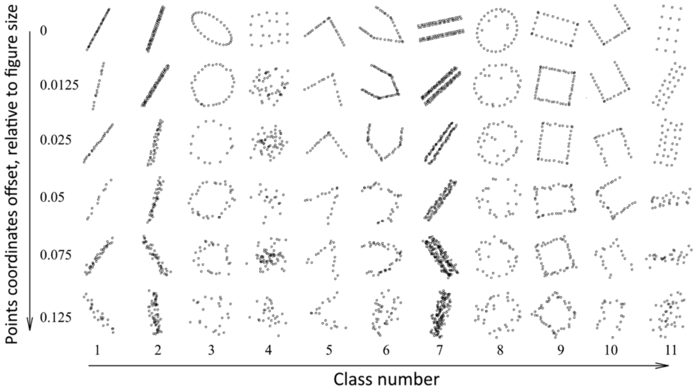

Figure 5. Examples of generated images with various distortions (uncertainties).

As one can see in Figure 5, the uncertainty of the coordinates of individual points distorts the image, but the common shapes retain their characteristic features.

3. Image Generation and Distortion Model

Figure 6 shows our way of offset implementation—as the uncertainty increases, the images are distorted more strongly. Figure 4 shows the result of image generation. The result of the modeling is a set of images with a resolution of 256 × 256 pixels. This image generation model is based on its interpretation ease; research results in the future can be generalized toward a wider class of tasks. The chosen resolution is sufficient to provide necessary accuracy in the representation of distorted images without affecting the convolutional neural network speed and complexity and the array size. The resulting image generation and distortion model can be described as follows (Figure 6):

1. Creating a shape vector model with uniformly distributed points (the number of points is random and distributed uniformly in some range).
2. Adding (to every point) the individual position offset described with a Gaussian random distribution (the variance of the distribution sets the amount of uncertainty). The Gaussian random distribution is well-suited to describe the uncertainty that occurs due to a set of different reasons. The uncertainty is measured relative to the shape vector model size. For example, an uncertainty value of 0.1 relative units says that the position of points may vary within 0.1 of the maximum linear size of the figure.
3. The resulting figure is rotated at a random angle.

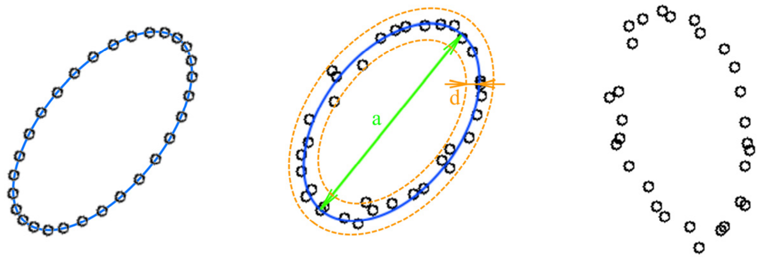

Figure 6. Image generation.

4. Structure of Neural Network

This architecture is one of the simplest and popular [29]. The convolutional network consists of alternating convolutional and subsampling layers (Figure 7), and the training process consists of repeatedly presenting a training dataset to the network (each iteration is called an epoch) and correcting the synaptic weights of the network at each iteration. When the synaptic weights stabilize, the mean error on the whole training set is minimized, the network can be considered trained. Images obtained from the image generation module merged to create the training dataset.

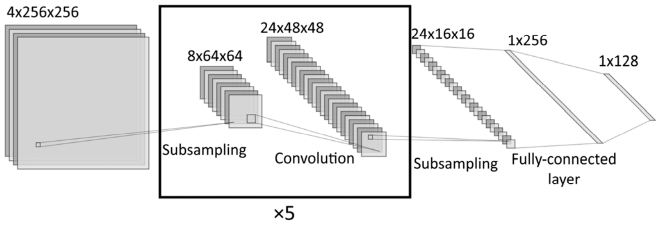

Figure 7. Structure of CNN.

5. Estimation of the Recognition Quality Dependence on the Amount of Uncertainty

First, we used a neural network to recognize each testing dataset with its amount of uncertainty $\boldsymbol{U_{TS}}$, which allowed us to determine the dependence of the recognition accuracy on uncertainty in the testing dataset $\boldsymbol{P}(\boldsymbol{U_{TS}})$.

To obtain more information, we trained two independent CNNs with identical structures on two training datasets, with two different amounts of uncertainty: $\boldsymbol{U_{TR}} = 0$ and $\boldsymbol{U_{TR}} = [0 \ldots 0.025]$. The hyperparameters remained unchanged. Separate datasets with image sequence randomizations were created for each experiment. The rule for dataset generation is described in [23]. Initial weights of CNNs were randomized as run-to-run. Trained CNNs were used for the recognition of separately generated datasets containing images with various uncertainties $\boldsymbol{U_{TS}}$. All recognition probabilities $\boldsymbol{P}$ obtained in these series of simulations were averaged over all series of experiments with fixed $\boldsymbol{U_{TS}}$.

As a result, two arrays of recognition accuracies, as functions of testing dataset uncertainty $\boldsymbol{P}(\boldsymbol{U_{TS}})$, were obtained. The results of this experiment are summarized in the graph shown in Figure 8.

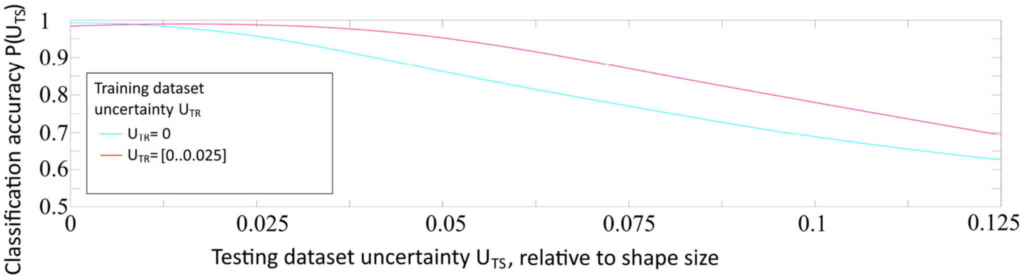

Figure 8. Dependence of the recognition accuracy on the amount of testing dataset uncertainty $\boldsymbol{U_{TS}}$, obtained by a network trained with uncertainty $\boldsymbol{U_{TR}}$ = 0 and $\boldsymbol{U_{TR}}$ = (0..0.025).

The comparison of two graphs allowed us to draw three main conclusions:

1. At $\boldsymbol{U_{TR}}$ = 0 the accuracy curve is monotonous. It confirms the consistency and robustness of the chosen model.
2. At $\boldsymbol{U_{TR}}$ = (0 ... 0.025) the accuracy curve is "no more" monotonous; it shows a small drop of accuracy below $\boldsymbol{U_{TS}}$ = 0.01. This phenomenon shows that changing the uncertainty proportions in the training dataset may affect the recognition of ideal images.
3. The maximum accuracy is achieved at $\boldsymbol{U_{TR}}$ = 0 and $\boldsymbol{U_{TS}}$ = 0, but the integral (overall) accuracy at all considered values of $\boldsymbol{U_{TS}}$ is achieved at $\boldsymbol{U_{TR}}$ = (0 ... 0.025). This phenomenon can be explained by the limited ability of the neural network, trained only on perfect images without uncertainty, to generalize the features presented in the distorted examples with uncertainty.

It is well known that the characteristics of the training dataset strongly affect the neural network training quality and the accuracies of their future new dataset recognition tasks [30]. To optimize the convolutional neural network training (to discover the optimal training dataset parameters to improve the recognition quality of images with different uncertainties) in this paper, we conducted experiments by training the convolutional neural network using datasets with various amounts of uncertainty. From the graphs shown in Figures 7 and 8, we cannot make unequivocal conclusions about training optimality. In our research, we conducted more training experiments, thereby "unfolding" the results in a new dimension (amount of uncertainty in the training dataset). Figure 9 shows the dependencies of recognition accuracy on the amount of uncertainty $\boldsymbol{U_{TS}}$, obtained by the networks trained on datasets with $\boldsymbol{U_{TR}}$, varying from 0 to 0.125 in increments of 0.025, shown in one graph for clarity.

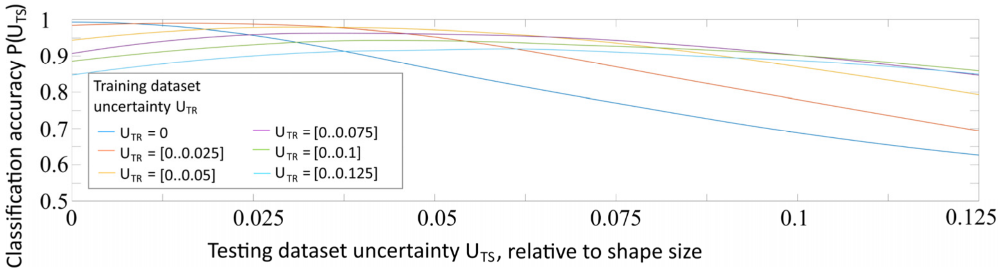

Figure 9. Dependency of recognition accuracy on the amount of uncertainty, obtained by the networks trained on datasets with $\boldsymbol{U_{TR}}$ varying from 0 to 0.125 stepping 0.025.

Figure 9 shows that the maximum recognition accuracy is achieved at $\boldsymbol{U_{TS}} \leq \boldsymbol{U_{TR}}$. Moreover, an analysis of the graphs in Figure 9 shows that for $\boldsymbol{U_{TR}} \geq 0.025$, the accuracy graphs change their shapes from monotonic to non-monotonic, indicating that the training is not optimal and is non-robust. The dependence of $P(U_{TS})$ should be monotonous, $dP/dU_{TS} \leq 0$ due to the fall of the meaningful information quantity with rising uncertainty. This rule may be used as the criterion for training correctness and robustness.

6. Evaluating the Dependence of Recognition Quality on Uncertainties in Testing (U_{TS}) and Training (U_{TR}) Datasets

The results obtained at this stage (presented in the graphs in Figure 9) prompted us to conduct comprehensive research on the network behaviors under changing training dataset uncertainty U_{TR}. For a more detailed analysis of the image recognition accuracy, we generated a manifold of training datasets with different uncertainties $U_{TR\ i} = d_i/a$. Individual copies of the convolutional neural network shown in Figure 7 were trained, and their weights were obtained for each training dataset. Then, each trained network was made to recognize each test dataset with different uncertainties $U_{TS\ j} = d_j/a$. It allowed us to obtain a two-dimensional array (matrix) of recognition accuracy, depending on the training and testing dataset uncertainties $P = P(U_{TR}; U_{TS})$. The resulting recognition accuracy matrix contains complete information about the external training-recognition system specifications and can be used to assess the system consistency and robustness (Figure 10).

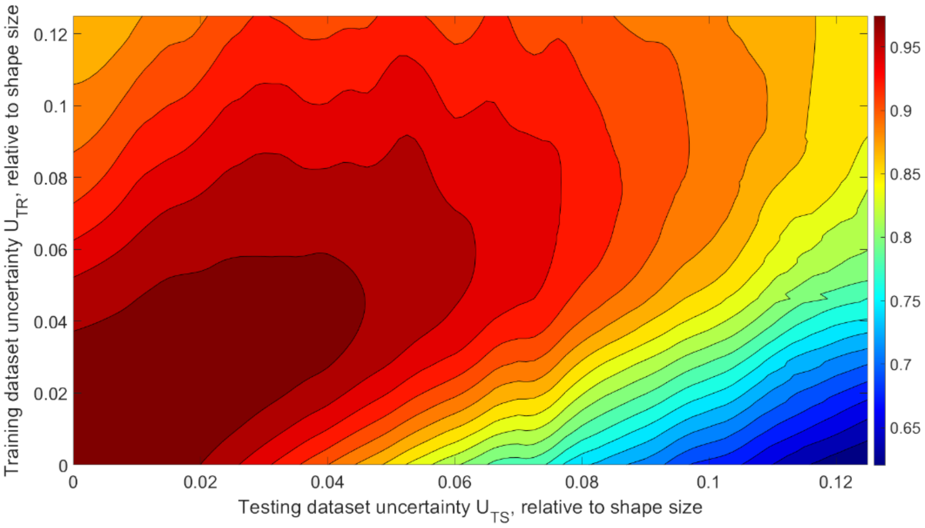

Figure 10. Dependency of recognition accuracy on the amount of training and testing dataset uncertainties U_{TR} and U_{TS}.

As one can see in Figure 10, the network better recognizes data with the same or a lower uncertainty, which was used in the training dataset, which confirms the correctness of the training because the network better recognizes data with similar statistical characteristics than those used during training. The recognition accuracy achieved by the network trained by the dataset with high U_{TR} drops on data with low U_{TS}; it is caused by changing proportions of meaningful and noisy data proportions during training. The "ripple" that can be seen in the graph is an effect of the limited dataset and should be considered a statistical inaccuracy. Since each recognition system in real life has its minimum accuracy requirements [31], it is necessary to analyze the recognition accuracy for different values of the minimum recognition accuracy thresholds.

7. Cumulative Recognition Accuracy at Different Thresholds

In practical tasks solved by neural networks, it is often not necessary to recognize data with extremely high distortions. Moreover, there are often minimal classification/recognition accuracy requirements to the developed solutions that use neural networks. Often, in practice, there is a necessity to gain a high enough "certainty" of the system. To obtain more valuable results, we selected areas that included the values of the testing dataset uncertainty with a recognition accuracy higher than $\boldsymbol{P_{thr}}$, in which the recognition accuracy was higher than the selected thresholds, which allowed estimating the acceptable coordinate uncertainties to provide the necessary recognition accuracy. Figure 11 shows the area that included the values of the testing dataset uncertainty $\boldsymbol{U_{TS}}$ that provided recognition accuracy $\boldsymbol{P_{thr}}$ higher than 90%.

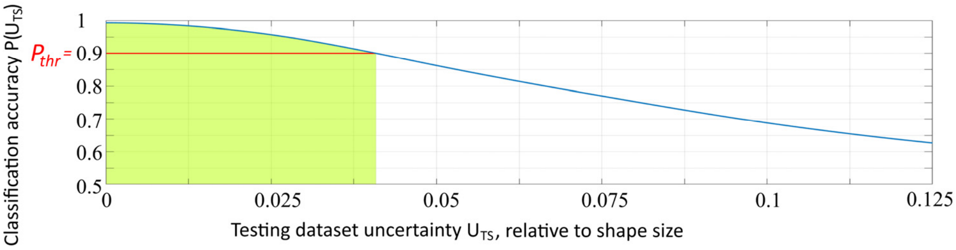

Figure 11. The area that includes the values of testing dataset uncertainty with recognition accuracy higher than 90%.

The highlighted area in Figure 11 is calculated as

$$Q(P_{thr}) = \sum_{U_{TS}=U_{TS}^{\min}, P \geq P_{thr}}^{U_{TS}=U_{TS}^{\max}, P \geq P_{thr}} P(U_{TS}). \tag{5}$$

The task of determining the optimal training dataset parameters to obtain the required recognition accuracy above threshold $\boldsymbol{P_{thr}}$ arises. For each network trained on datasets with different uncertainties, we obtained the integral recognition accuracy values $\boldsymbol{Q}$ at different thresholds:

$$Q(U_{TR}; P_{thr}) = \sum_{U_{TS}=U_{TS}^{\min}, P \geq P_{thr}}^{U_{TS}=U_{TS}^{\max}, P \geq P_{thr}} P(U_{TR}; U_{TS}). \tag{6}$$

In this case, $\boldsymbol{Q}$ is an integral value of the classification accuracy for all test datasets (with all uncertainties), for which the recognition accuracy exceeded threshold $\boldsymbol{P \geq P_{thr}}$. The obtained data are summarized in the graph shown in Figure 12 below.

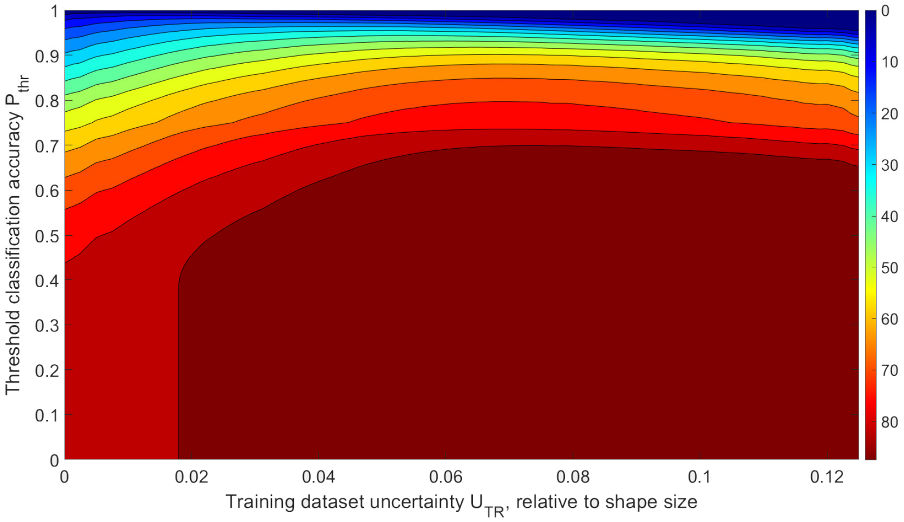

Figure 12. Dependence of the integral value of the recognition accuracy $\boldsymbol{Q}$ for all $\boldsymbol{P > P_{thr}}$ on $\boldsymbol{P_{thr}}$ and training dataset uncertainty $\boldsymbol{U_{TR}}$.

This graph represents the integral (overall) average rate of correct recognition for data giving the recognition probability greater than $\boldsymbol{P_{thr}}$, depending on $\boldsymbol{P_{thr}}$ and training dataset uncertainty $\boldsymbol{U_{TR}}$. New data recognized with accuracies lower than the required $\boldsymbol{P_{thr}}$ were not counted. The colors in the graph show areas with identical mean recognition accuracies for all $\boldsymbol{P} > \boldsymbol{P_{thr}}$. There is always an optimal training dataset uncertainty $\boldsymbol{U_{TR}}$, depending on the lower threshold of the required recognition accuracy.

Using Figure 12, one can determine the optimal training dataset uncertainty $\boldsymbol{U_{TR}}$ needed to achieve maximal integral recognition accuracy $\boldsymbol{Q}$ for all data with local recognition probability exceeding the threshold $\boldsymbol{P} \geq \boldsymbol{P_{thr}}$. It can be illustrated in Figure 13.

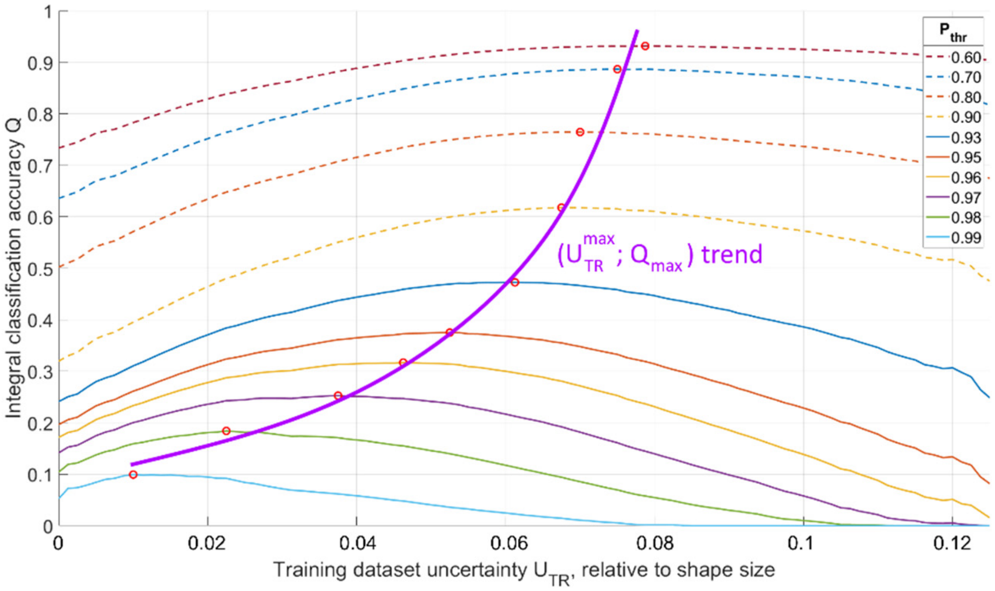

Figure 13. Dependence of the integral recognition accuracy $\boldsymbol{Q}$ on training dataset uncertainty $\boldsymbol{U_{TR}}$ for various $\boldsymbol{P_{thr}}$ and optimal values of training dataset uncertainty $\boldsymbol{U_{TR}}$ for different recognition accuracy thresholds $\boldsymbol{P_{thr}}$.

Assuming that the optimal training dataset at the required threshold of the minimum classification accuracy is the dataset that gives the highest value of integral classification accuracy $\boldsymbol{Q}$, the graph in Figure 13 is convenient for determining the optimal training dataset uncertainty $\boldsymbol{U_{TR}}$. If we analyze the dependence of the integral recognition accuracy $\boldsymbol{Q}$ on the training dataset uncertainty $\boldsymbol{U_{TR}}$ at a fixed threshold $\boldsymbol{P_{thr}}$, we obtain a graph with a clear maximum, the position of which will indicate the optimal value of training dataset uncertainty $\boldsymbol{U_{TR}}$ (Figure 13, $\boldsymbol{Q_{max}}$ for various $\boldsymbol{P_{thr}}$ are shown with red dots). An analysis of Figure 13 allows us to conclude that training the network with optimal $\boldsymbol{U_{TR}}$ for fixed $\boldsymbol{P_{thr}}$ significantly increases the integral recognition accuracy compared to training the network with an ideal dataset with $\boldsymbol{U_{TR}}$ = 0. For example, for $\boldsymbol{P_{thr}}$ = 0.9, $\boldsymbol{Q_{max}}$ exceeds $\boldsymbol{Q_0}$ by 94% ($\boldsymbol{Q_{max}}$ = 0.62 is obtained at $\boldsymbol{U_{TR}}$ = 0.068 and $\boldsymbol{Q_0}$ = 0.32 is obtained at $\boldsymbol{U_{TR}}$ = 0).

8. Noisy Images Recognition

To generalize the results of this study, we conducted a simulation with different types of images and noise using the same CNN structure and the same approach involving an analysis of $\boldsymbol{P}(\boldsymbol{U_{TR}}; \boldsymbol{U_{TS}})$ dependence. The examples of images are shown in Figure 14. We added white Gaussian noise with mean $\boldsymbol{\mu}$ = 0 and various standard deviations $\boldsymbol{\sigma}$ to the images to generate separate datasets for training and recognition. The uncertainty $\boldsymbol{U}$ is therefore defined via (1), where $\boldsymbol{I_{noise}}$ = $\boldsymbol{\sigma}$.

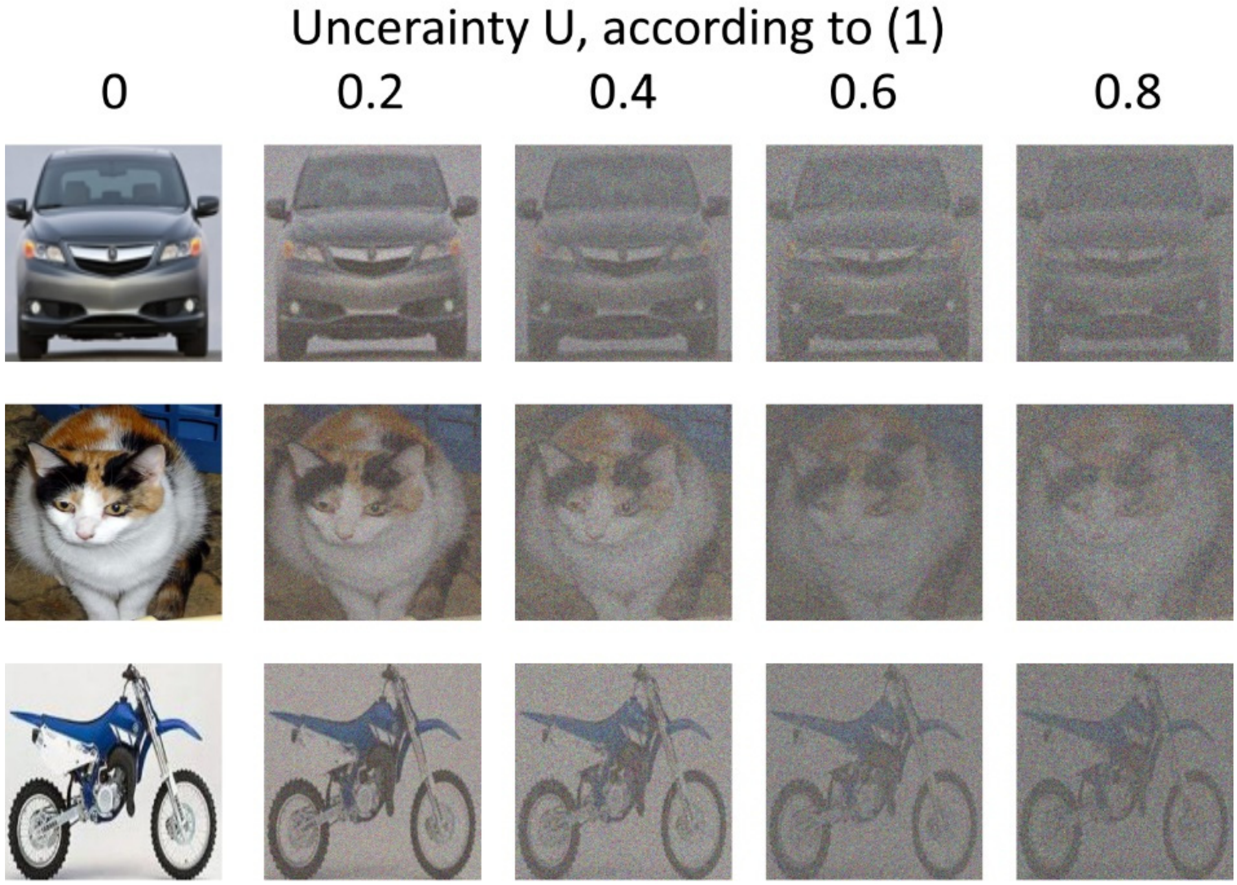

Figure 14. Examples of images with various amounts of noise added.

We used a convolutional neural network identical to the one described before to solve the noisy image classification task. Five datasets for training were generated. The training dataset parameters are described below:

(1) The first dataset had $\boldsymbol{U_{TR}}$ = 0 for all images (no noise was added).
(2) The second dataset was divided into three parts containing equal numbers of images; the first part had $\boldsymbol{U_{TR}}$ = 0, the second part had $\boldsymbol{U_{TR}}$ = 0.04, the third part had $\boldsymbol{U_{TR}}$ = 0.08.
(3) The third dataset was divided into three parts containing equal numbers of images; the first part had $\boldsymbol{U_{TR}}$ = 0, the second part had $\boldsymbol{U_{TR}}$ = 0.12, the third part had $\boldsymbol{U_{TR}}$ = 0.16.
(4) The fourth dataset was divided into three parts containing equal numbers of images; the first part had $\boldsymbol{U_{TR}}$ = 0, the second part had $\boldsymbol{U_{TR}}$ = 0.2, the third part had $\boldsymbol{U_{TR}}$ = 0.4.
(5) The fifth dataset was divided into three parts containing equal numbers of images; the first part had $\boldsymbol{U_{TR}}$ = 0, the second part had $\boldsymbol{U_{TR}}$ = 0.4, the third part had $\boldsymbol{U_{TR}}$ = 0.8.

Thus, four of five datasets had various amounts of noise added to the original images.

Five independent CNNs with identical structures were trained using five datasets described above. The hyperparameters remained unchanged. Separate datasets with image sequence randomizations were created for each experiment. Initial weights of CNNs were randomized run-to-run. Trained CNNs were used for recognition of separately generated testing datasets containing images with various uncertainty values $\boldsymbol{U_{TS}}$. Testing datasets had homogenous structures: all images in each dataset had the same amount of additional noise, giving us fixed $\boldsymbol{U_{TS}}$ for the whole dataset. All recognition probabilities $\boldsymbol{P}$ obtained in these series of simulations were averaged over all series of experiments with fixed $\boldsymbol{U_{TS}}$.

The simulation results are shown in Figure 15.

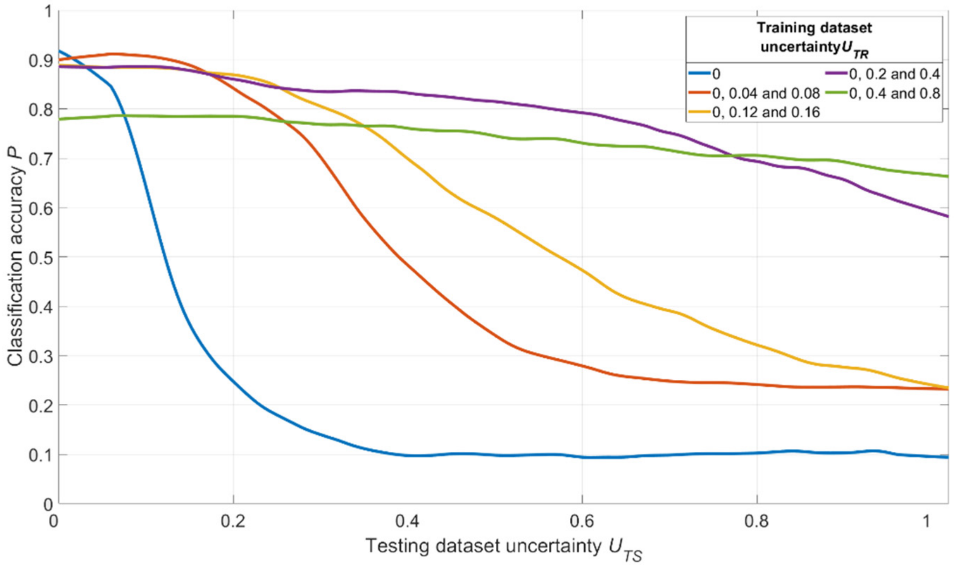

Figure 15. Dependence of the recognition accuracy on the amount of testing dataset uncertainty U_{TS}, obtained by five CNNs trained with various uncertainties U_{TR} shown in the image.

The results shown in Figures 15 and 16 allow us to state that moderate training dataset uncertainty U_{TR} should be optimal for recognition of noisy images with high threshold recognition probability P_{thr}. For the current example, the optimal training dataset is the second one, having U_{TR} = {0; 0.04; 0.08} for P_{thr} > 0.87. A further increase of U_{TR} leads to the fall of integral classification accuracy for high P_{thr}. This result allows us to generalize the confirmation of existence of optimal training dataset uncertainty for these types of data.

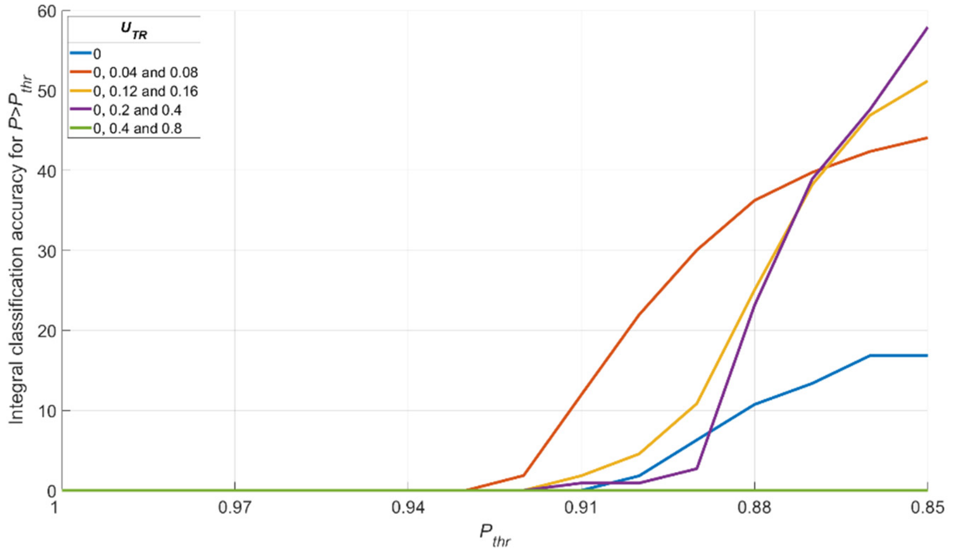

Figure 16. Dependence of the integral value of the recognition accuracy Q for all $P > P_{thr}$ on P_{thr} for various training dataset uncertainties U_{TR}.

9. Additional Types of Distortion

To check the correctness of our findings, we conducted a series of experiments with various types of image distortions. We conducted these series of simulations with different types of images and noise/distortions using the same CNN structure and the same approach, involving analysis of $P(U_{TR}; U_{TS})$ dependence. The examples of images are shown in Figure 17.

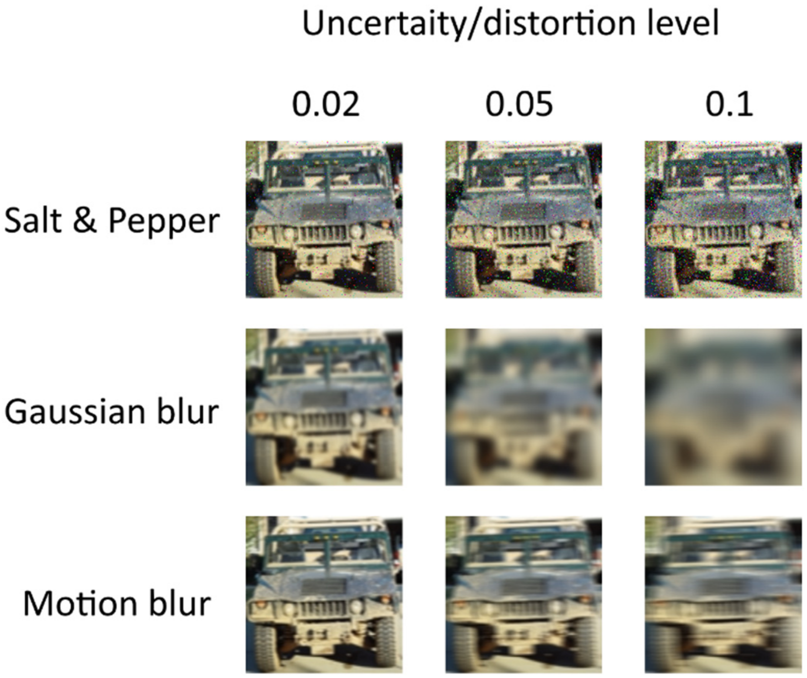

Figure 17. Examples of images with various amounts of salt and pepper noise, Gaussian blur and motion blur added.

The simulation results are shown in Figures 18–20. For salt and pepper noise, the uncertainty was calculated using Formula (1), where $\boldsymbol{I_{noise}}$ is a number of "noise" pixels and $\boldsymbol{I_{info}}$ is the total number of pixels in the image. For Gaussian and motion blur, uncertainty was calculated as follows:

$$U = S_{kernel} / S_{image}, \quad (7)$$

where $\boldsymbol{S_{kernel}}$ is the size of Gaussian and motion filter kernels and $\boldsymbol{S_{image}}$ is the size of the image.

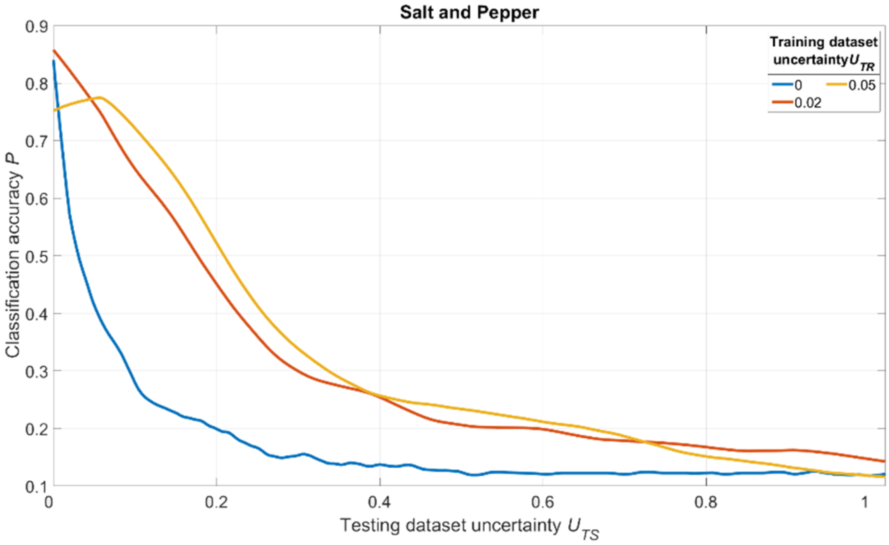

Figure 18. Dependence of the recognition accuracy on the amount of testing dataset uncertainty $\boldsymbol{U_{TS}}$, obtained by three CNNs trained with various uncertainties $\boldsymbol{U_{TR}}$ shown in the image. The uncertainty is produced by adding salt and pepper noise and calculated using (1).

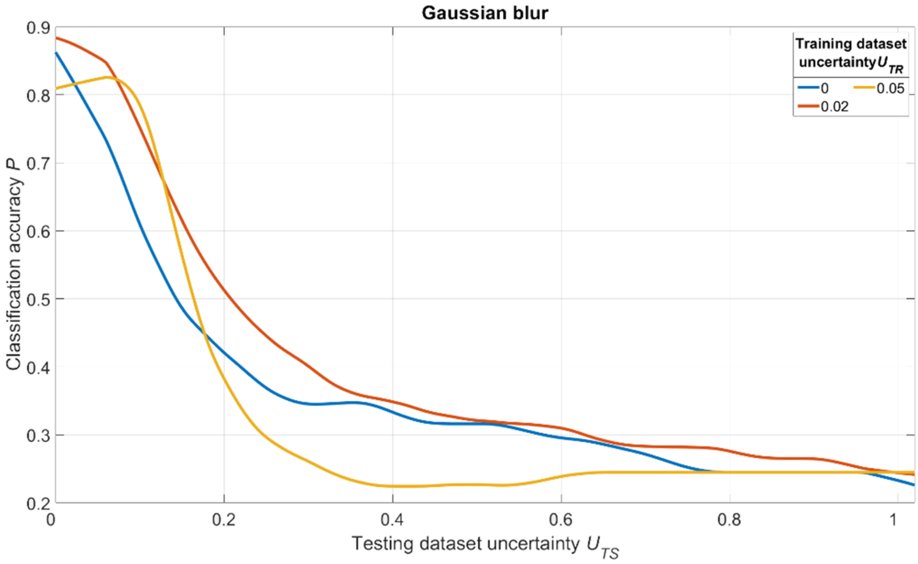

Figure 19. Dependence of the recognition accuracy on the amount of testing dataset uncertainty U_{TS}, obtained by three CNNs trained with various uncertainties U_{TR} shown in the image. The uncertainty is produced by adding Gaussian blur and calculated using (7).

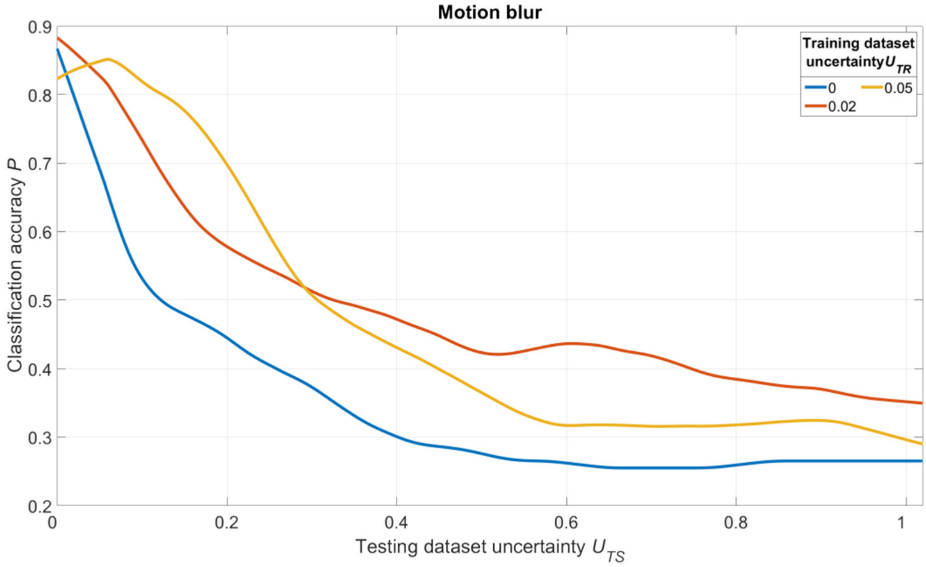

Figure 20. Dependence of the recognition accuracy on the amount of testing dataset uncertainty U_{TS}, obtained by three CNNs trained with various uncertainties U_{TR} shown in the image. The uncertainty is produced by adding motion blur and calculated using (7).

The analysis of Figures 18–20 shows one tendency: independently on noise/distortion type during training; its amount influences the recognition accuracy in the same way. Using images without additional uncertainty for training (U_{TR} = 0, blue curves) leads to a fast decrease of recognition accuracy with the rise of U_{TS}. This fact tells us that a CNN trained in such a manner would be vulnerable to adversarial attacks. The results also show that a moderate training dataset uncertainty U_{TR} (red curves) in case of salt and pepper noise, as well as of Gaussian and motion blur, should be optimal for recognition of noisy/distorted images without loss of accuracy for original image recognition. This result allows us to generalize the confirmation of the existence of optimal training dataset uncertainty for these types of data, noise, and distortion.

Thus, we conducted simulations under a variety of conditions, and all of them show the same regularity: there exists an optimal amount of uncertainty of various physical natures that, when applied to the training dataset, lead to significant improvements of trained CNN noise immunity and overall recognition quality. The excessive amount of training dataset uncertainty (yellow curves in Figures 18–20) makes the training non-robust. It can be seen as a non-monotonous character of the corresponding (yellow) noise immunity curves (growth of recognition accuracy with increasing $\boldsymbol{U_{TS}}$).

10. Conclusions

This study shows that the amount of uncertainty in the training dataset $\boldsymbol{U_{TR}}$ significantly affects the recognition accuracy itself and the dependence of the recognition accuracy on the uncertainty in the testing dataset $\boldsymbol{U_{TS}}$. We analyzed the recognition accuracies of multiple datasets with different uncertainties and obtained the dependence of recognition accuracy on the training dataset uncertainty. The existence of an optimal (in terms of recognition accuracy) amount of uncertainty in the training dataset (for the neural networks working with undefined uncertainty data) was hypothesized and proven for various types of images and noise. We have shown that the determination of this optimum can be performed using statistical modeling. Training the network using the dataset with optimal uncertainty $\boldsymbol{U_{TR}}$ provides a significant increase of recognition accuracy compared to training on the ideal dataset. The CNN learns not to use the noise/distortion as features during training because noise/distortion does not help the CNN distinguish different images. Therefore, it enhances the generalization ability of the CNN and its immunity to adversarial attacks. At the same time, excessive noise/distortion ruins the training, leading to recognition accuracy decrease. This finding can be used to improve recognition quality just by adding some (optimal) amounts of noise to the training dataset.

The obtained results are applicable to convolutional neural networks with common structures and different types of data uncertainty (Gaussian noise, distortion of point locations, salt and pepper noise, Gaussian and motion blur, etc.). Future work will be devoted toward expanding the results on different neural network structures and different tasks (for example, object detection). We hope to find an analytical solution for optimal training dataset uncertainty determination without massive statistical simulations.

Author Contributions: Conceptualization, V.Z. and M.T.; methodology, V.Z. and M.T.; software, V.Z. and M.T.; validation, V.Z. and M.T.; formal analysis, V.Z. and M.T.; investigation, V.Z. and M.T.; resources, V.Z. and M.T.; data curation, V.Z. and M.T.; writing—original draft preparation, V.Z. and M.T.; writing—review and editing, V.Z. and M.T.; visualization, V.Z. and M.T.; supervision, V.Z. and M.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hinton, G.; Deng, L.; Yu, D.; Dahl, G.E.; Mohamed, A.-R.; Jaitly, N.; Senior, A.; Vanhoucke, V.; Nguyen, P.; Sainath, T.N.; et al. Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups. *IEEE Signal Process. Mag.* **2012**, *29*, 82–97. [CrossRef]
2. Alex, K.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
3. Bishop, C.M. *Neural Networks for Pattern Recognition*; Oxford University Press: Oxford, UK, 1995.
4. Hong, P.; Wen, Z.; Huang, T.S. Real-time speech-driven face animation with expressions using neural networks. *IEEE Trans. Neural Netw.* **2002**, *13*, 916–927. [CrossRef] [PubMed]
5. Ptucha, R.; Such, F.P.; Pillai, S.; Brockler, F.; Singh, V.; Hutkowski, P. Intelligent character recognition using fully convolutional neural networks. *Pattern Recognit.* **2019**, *88*, 604–613. [CrossRef]
6. Masood, S.Z.; Shu, G.; Dehghan, A.; Ortiz, E.G. License plate detection and recognition using deeply learned convolutional neural networks. *arXiv* **2017**, arXiv:1703.07330.

7. Lu, Y.; Yi, S.; Zeng, N.; Liu, Y.; Zhang, Y. Identification of rice diseases using deep convolutional neural networks. *Neurocomputing* **2017**, *267*, 378–384. [CrossRef]
8. Beritelli, F.; Capizzi, G.; Sciuto, G.L.; Napoli, C.; Scaglione, F. Automatic heart activity diagnosis based on Gram polynomials and probabilistic neural networks. *Biomed. Eng. Lett.* **2018**, *8*, 77–85. [CrossRef]
9. Jain, N.; Kumar, S.; Kumar, A.; Shamsolmoali, P.; Zareapoor, M. Hybrid deep neural networks for face emotion recognition. *Pattern Recognit. Lett.* **2018**, *115*, 101–106. [CrossRef]
10. Li, K.; Jin, Y.; Akram, M.W.; Han, R.; Chen, J. Facial expression recognition with convolutional neural networks via a new face cropping and rotation strategy. *Vis. Comput.* **2020**, *36*, 391–404. [CrossRef]
11. Egmont-Petersen, M.; de Ridder, D.; Handels, H. Image processing with neural networks—A review. *Pattern Recognit.* **2002**, *35*, 2279–2301. [CrossRef]
12. Singh, D.; Kumar, V.; Vaishali; Kaur, M. Classification of COVID-19 patients from chest CT images using multi-objective differential evolution–based convolutional neural networks. *Eur. J. Clin. Microbiol. Infect. Dis.* **2020**, *39*, 1379–1389. [CrossRef] [PubMed]
13. Khan, A.; Sohail, A.; Zahoora, U.; Qureshi, A.S. A survey of the recent architectures of deep convolutional neural networks. *Artif. Intell. Rev.* **2020**, *53*, 5455–5516. [CrossRef]
14. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and harnessing adversarial examples. *arXiv* **2014**, arXiv:1412.6572.
15. Chaturvedi, A.; Garain, U. Mimic and Fool: A Task-Agnostic Adversarial Attack. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 1801–1808. [CrossRef] [PubMed]
16. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Region-Based Convolutional Networks for Accurate Object Detection and Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *38*, 142–158. [CrossRef]
17. Xiao, Y.; Pun, C.-M.; Liu, B. Fooling Deep Neural Detection Networks with Adaptive Object-oriented Adversarial Perturbation. *Pattern Recognit.* **2021**, *115*, 107903. [CrossRef]
18. Abdar, M.; Pourpanah, F.; Hussain, S.; Rezazadegan, D.; Liu, L.; Ghavamzadeh, M.; Fieguth, P.; Cao, X.; Khosravi, A.; Acharya, U.R.; et al. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Inf. Fusion* **2021**, *76*, 243–297. [CrossRef]
19. Domingos, P. A few useful things to know about machine learning. *Commun. ACM* **2012**, *55*, 78–87. [CrossRef]
20. Woods, W.A. Important issues in knowledge representation. *Proc. IEEE* **1986**, *74*, 1322–1334. [CrossRef]
21. Ferrer, M.A.; Vargas, J.F.; Morales, A.; Ordonez, A. Robustness of Offline Signature Verification Based on Gray Level Features. *IEEE Trans. Inf. Forensics Secur.* **2012**, *7*, 966–977. [CrossRef]
22. Jalalvand, A.; Demuynck, K.; De Neve, W.; Martens, J.-P. On the application of reservoir computing networks for noisy image recognition. *Neurocomputing* **2018**, *277*, 237–248. [CrossRef]
23. Ziyadinov, V.V.; Tereshonok, M.V. Mathematical models and recognitions methods for mobile subscribers mutual placement. *T-Comm* **2021**, *15*, 49–56. [CrossRef]
24. Studer, S.; Bui, T.B.; Drescher, C.; Hanuschkin, A.; Winkler, L.; Peters, S.; Müller, K.-R. Towards CRISP-ML(Q): A Machine Learning Process Model with Quality Assurance Methodology. *Mach. Learn. Knowl. Extr.* **2021**, *3*, 20. [CrossRef]
25. Limberg, C.; Wersing, H.; Ritter, H. Beyond Cross-Validation—Accuracy Estimation for Incremental and Active Learning Models. *Mach. Learn. Knowl. Extr.* **2020**, *2*, 18. [CrossRef]
26. Bengio, Y. *Learning Deep Architectures for AI*; Now Publishers Inc.: Boston, MA, USA, 2009. [CrossRef]
27. Szegedy, C.; Toshev, A.; Erhan, D. Deep neural networks for object detection. In *Advances in Neural Information Processing Systems 26 (NIPS 2013)*; Curran Associates Inc.: Red Hook, NY, USA, 2013.
28. Goodfellow, I.J.; Bulatov, Y.; Ibarz, J.; Arnoud, S.; Shet, V. Multi-digit number recognition from Street View imagery using deep convolutional neural networks. In Proceedings of the International Conference on Learning Representations, Banff, AB, Canada, 14–16 April 2014.
29. Haykin, S.S. *Neural Networks and Learning Machines*; Prentice Hall: Hoboken, NJ, USA, 2009.
30. Shin, H.C.; Roth, H.R.; Gao, M.; Lu, L.; Xu, Z.; Nogues, I.; Yao, J.; Mollura, D.; Summers, R.M. Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Charac-teristics and Transfer Learning. *IEEE Trans. Med. Imaging* **2016**, *35*, 1285–1298. [CrossRef]
31. Ziyadinov, V.V.; Kurochkin, P.S.; Tereshonok, M.V. Convolutional Neural Network Training Optimization for Low Point Density Image Recognition. *J. Commun. Technol. Electron.* **2021**, *66*, 1363–1369. [CrossRef]