The Role of Noisy Data in Improving CNN Robustness for Image Classification

Oscar H. Ramírez-Agudelo[a], Nicoleta Gorea[a], Aliza Reif[a], Lorenzo Bonasera[a], and Michael Karl[a]

[a]German Aerospace Center (DLR), Institute for AI Safety and Security, Rathausallee 12, 53757 Sankt Augustin, Germany

ABSTRACT

Data quality plays a central role in the performance and robustness of convolutional neural networks (CNNs) for image classification. While high-quality data is often preferred for training, real-world inputs are frequently affected by noise and other distortions. This paper investigates the effect of deliberately introducing controlled noise into the training data to improve model robustness. Using the CIFAR-10 dataset, we evaluate the impact of three common corruptions, namely Gaussian noise, Salt-and-Pepper noise, and Gaussian blur at varying intensities and training set pollution levels. Experiments using a Resnet-18 model reveal that incorporating just 10% noisy data during training is sufficient to significantly reduce test loss and enhance accuracy under fully corrupted test conditions, with minimal impact on clean-data performance. These findings suggest that strategic exposure to noise can act as a simple yet effective regularizer, offering a practical trade-off between traditional data cleanliness and real-world resilience.

Keywords: deep learning, CNNs, data quality, CIFAR-10, noise injection, image classification, model robustness

1. INTRODUCTION

In recent years, convolutional neural networks (CNNs) have formed the backbone of numerous computer vision applications, achieving state-of-the-art results in image classification, object detection, and segmentation tasks.[1–4] These advancements have been largely driven by increased computational power, larger annotated datasets, and improved network architectures. However, such success is critically dependent on the quality of the data used during training and evaluation. Data quality, described by attributes such as accuracy, consistency, completeness, and representativeness, strongly influences model performance, generalizability, and robustness in practical deployment scenarios.[5–7]

Despite the availability of high-quality benchmark datasets like ImageNet[8] and medium-scale benchmarks such as CIFAR,[9] real-world image data often contains imperfections. It is susceptible to various degradations, including sensor noise, poor lighting, motion blur, occlusion, and compression artifacts.[10–12] These distortions can substantially impair model performance, particularly when the training data does not reflect such variability. The resulting domain shift between clean training data and noisy real-world inputs poses a significant challenge to robustness.[13–15]

To address this challenge, researchers should focus on both data quality management and model robustness. While conventional best practices emphasize curating clean and balanced datasets, recent work has shown that strategically incorporating degraded or noisy data during training can improve robustness to real-world noise.[16–19] Instead of removing all noise entirely, exposing models to controlled levels of degradation can serve as a form of regularization, enhancing their ability to generalize to unseen conditions.[20,21]

In this study, we investigate how different types of image degradation, specifically *Gaussian noise*, *Salt-and-Pepper noise*, and *Gaussian blur* affect the robustness of CNNs in image classification tasks. Building on prior research into noise robustness and data augmentation, we conduct a series of experiments using the CIFAR-10

Further author information: (Send correspondence to O.H.R-A.)
O.H.R-A.: E-mail: Oscar.RamirezAgudelo@dlr.de

Applications of Machine Learning 2025, edited by Michael E. Zelinski, Tarek M. Taha, Barath Narayanan Abdul A. S. Awwal, Khan M. Iftekharuddin, Proc. of SPIE Vol. 13606, 136060P
2025 Published by SPIE · 0277-786X · doi: 10.1117/12.3063563

dataset, adding different intensities of Gaussian noise, Salt-and-Pepper noise, and Gaussian blur to evaluate their impact on model performance.[9,22] We train the resulting models on both clean and noisy images to evaluate how noise exposure during training influences classification performance on both clean and degraded test sets. Performance is assessed using top-1 classification accuracy, with 10% of the training set held out for validation (see more details in Sect. 3). The contributions of this work are threefold:

- We conduct an empirical analysis to evaluate the effect of noisy data inclusion on CNN robustness, adding to recent efforts in data-centric deep learning.[23]
- We propose a simple yet effective noise-augmented training strategy that improves generalization to noisy test conditions while maintaining strong performance on clean data.[24]
- We discuss trade-offs between classical definitions of high data quality and the benefits of incorporating noise to promote real-world adaptability.[7,25]

The findings underscore the need to reconsider rigid definitions of data quality. We advocate for a broader perspective that values both cleanliness and representativeness of operational environments. This shift carries important implications for building machine learning systems that operate reliably under unpredictable and adverse real-world conditions.

Finally, we structure this paper as follows. Section 2 reviews related literature. Section 3 outlines the methodology used in this study. Section 4 presents the results, and Section 5 discusses them in detail. Section 6 concludes the paper. Appendix A reports the baseline performance of the adopted model prior to the experiments described in this work.

2. RELATED WORKS

The importance of *data quality* in training robust machine learning models has been extensively studied. Datasets such as ImageNet[8] and CIFAR-10[9] have been foundational for advancing deep learning in image classification. These datasets are carefully curated to ensure consistency and representativeness, which are critical for reliable model performance.[26–28]

Despite these benchmarks, real-world images often suffer from degradations including noise, blur, and compression artifacts, which can negatively impact model accuracy when models are trained solely on clean data.[29,30] Addressing this, Hendrycks and Dietterich[13] benchmarked the robustness of neural networks against a wide range of common corruptions, demonstrating the vulnerability of standard CNNs to noise and other perturbations.

The work by De Vries et al.[31] introduces Cutout, a regularization technique that randomly masks out square regions of input images during training. While Cutout is not explicitly designed for robustness to noise or corruptions, it improves generalization and has been shown to enhance accuracy on clean test data. Additionally, data augmentation has emerged as a practical method to improve robustness by introducing variability during training.[16,17,32] For example, Lopes et al.[16] propose Patch Gaussian, which injects Gaussian noise into random localized regions of input images. This method improves robustness to common corruptions without sacrificing and sometimes even improving accuracy on clean data.

Beyond generic augmentation, adversarial training, as introduced by Goodfellow et al.,[33] involves training models on deliberately perturbed examples to increase robustness against worst-case input alterations. Although adversarial robustness focuses on targeted perturbations, it shares the goal of improving performance under degraded input conditions.[34]

Techniques such as random cropping, flipping, and synthetic noise injection are widely used to enhance generalization.[35] In particular, noise injection during training has been shown to encourage models to learn more invariant features, improving resilience to noisy inputs.[35] Techniques such as mixup[36] blend clean and noisy samples to achieve better generalization and robustness.

Additional strategies to balance noise include curriculum learning,[37] noise-aware loss functions,[38] and adaptive noise filtering during training.[39] Domain adaptation techniques have also been applied to address shifts caused by image quality variations.[40]

This work aims to complement these efforts by empirically analysing the effect of incorporating different type of noises such as Gaussian noise, Salt-and-Pepper noise, and Gaussian blur into CIFAR-10 benchmark dataset as a test bed. By evaluating classification accuracy on both clean and noisy test sets, we aim to clarify how *data quality*, viewed as a spectrum from clean to noisy, influences model robustness in practical settings.

3. METHODOLOGY

This section describes the methodology adopted to investigate the impact of training data quality degradation on the robustness of convolutional neural networks (CNNs) for image classification by using the CIFAR-10 dataset.[9] In Section 3.1 details about the data, architecture, and training are provided. Subsequently, in Section 3.2 information about the type of noise and values selected are presented. Finally, the metrics used to evaluate the models are introduced in Section 3.3.

3.1 Model design and training overview

This section provides a comprehensive overview of the model design and training methodology employed in this work.

3.1.1 Dataset

The CIFAR-10 dataset contains 60,000 color images, each with a resolution of 32×32 pixels and three RGB channels. These images are evenly distributed across 10 classes: plane, car, bird, cat, deer, dog, frog, horse, ship, truck. This widely studied dataset offers excellent opportunities for comprehensive investigation and benchmarking.[1,2,9,16] Due to its manageable size and complexity, CIFAR-10 is frequently used to evaluate novel machine learning architectures, optimization techniques, and regularization strategies. Despite its simplicity compared to large-scale datasets like ImageNet, it remains a valuable testbed for rapid prototyping and ablation studies, particularly in convolutional neural networks (CNNs) and more recent deep learning models. Furthermore, CIFAR-10 continues to serve as a baseline dataset in exploring model robustness, transfer learning, and self-supervised learning paradigms.

3.1.2 Model Architecture

For this work, the Resnet-18 architecture implemented in the Github repository `https://github.com/kuangliu/pytorch-cifar` by *kuangliu* is adopted. Appendix A shows the training performance achieved by adopting this repository. An accuracy of approximately 93% is being retrieved, which is also reported in the open-source repository. This architecture serves as the backbone for our evaluations. To ensure fair comparisons, consistent training hyperparameters across all conditions are maintained. The network is trained using stochastic gradient descent (SGD) with a fixed learning rate $\alpha = 0.01$ and batch size of 128 and 100 for the training and testing set, respectively and total of 100 epochs. We select the best-performing model based on validation loss.

3.1.3 Training

For the experimental setup, we split 45000 images for training (90% of the original training set), 5000 images for validation (10% of the training set), and kept the test set unchanged at 10,000 images. We apply standard preprocessing, including normalization and random cropping, consistently across all experiments, following the implementation provided by *kuangliu*. To ensure robust results, we run each experiment 10 times per noise level and calculate the mean and standard deviation. This repetition mitigates the effects of randomness inherent in training, such as weight initialization, data shuffling, and stochastic optimization. By reducing variance and increasing consistency, this approach yields more reliable and statistically sound performance estimates, while also improving reproducibility by ensuring that results reflect genuine model behavior rather than random fluctuations.

3.2 Noise Injection

We introduce three different types of noise: Gaussian, Salt-and-Pepper, and Gaussian blur. To analyze the impact of noise severity on training, we create multiple training subsets with varying percentages of noisy images, corresponding to the following pollution levels: 0% (clean), 5%, 10%, 15%, 20%, 25%, 50%, 75%, and 100% (fully noisy). For each pollution level (9 levels), the corresponding fraction of training samples is randomly selected and corrupted by adding noise, while the respective rest remains clean. In the following, we describe the characteristics of each noise type in detail. This approach models realistic scenarios where data quality varies within the training set, allowing us to systematically evaluate the effect of mixed-quality data on model robustness.

3.2.1 Gaussian Noise

Gaussian noises with zero mean and several levels of standard deviation (σ) are added to the training images to simulate quality degradation*. Specifically, we focus on the effect of Gaussian noise introduced at varying levels to the training data, examining how different proportions of noisy samples influence model performance.

To assess the robustness of the proposed approach under varying levels of input perturbations, we inject additive Gaussian noise into the input data. The noise is sampled from a normal distribution $\mathcal{N}(0, \sigma^2)$, where σ denotes the standard deviation controlling the noise intensity. This simulates naturally occurring sensor noise or environmental variability in real-world acquisition settings. The noisy input I_{noisy} is generated as:

$$I_{\text{noisy}} = I_{\text{clean}} + N_{\text{gaussian}} \tag{1}$$

where I_{clean} is the original input and $N_{\text{gaussian}} \sim \mathcal{N}(0, \sigma^2)$ is the zero-mean Gaussian noise.

The probability density function of the Gaussian distribution is given by:

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{2}$$

with $\mu = 0$ and varying values of σ to control the noise strength.

3.2.2 Salt-and-Pepper Noise

To evaluate the robustness of the proposed approach against impulsive noise, we study the introduction of Salt-and-Pepper noise to the input data. Salt-and-Pepper is a form of impulse noise where certain pixel values are randomly replaced with either the minimum ("pepper") or maximum ("salt") intensity values in the input range.[41] This simulates sensor malfunctions or bit errors in transmission. We generate the noisy input I_{noisy} as follows:

$$I_{\text{noisy}}(i,j) = \begin{cases} I_{\min}, & \text{with probability } p_{\text{pepper}} \\ I_{\max}, & \text{with probability } p_{\text{salt}} \\ I_{\text{clean}}(i,j), & \text{with probability } 1 - p_{\text{salt\_and\_pepper}} \end{cases} \tag{3}$$

where $I_{\text{clean}}(i,j)$ denotes the original clean input at pixel location (i,j), and $p_{\text{salt\_and\_pepper}} = p_{\text{salt}} + p_{\text{pepper}}$ is the total noise density. $I_{\min}$ and $I_{\max}$ represent the minimum and maximum allowable intensities, respectively. In this work, p_{salt} and p_{pepper} contribute equally.

*The values of the levels are introduced in Section 3.2.4.

3.2.3 Gaussian Blur

In addition to noise injection, we apply Gaussian blur to evaluate the model's robustness against loss of fine-grained detail. Gaussian blur is a common image transformation that simulates sensor defocus or motion-induced blur by convoluting the image with a Gaussian kernel. This operation smooths the image, attenuating high-frequency components such as edges and textures. Formally, the blurred image I_{blurred} is obtained by the convolution:

$$I_{\text{blurred}}(i,j) = (I_{\text{clean}} * G_\sigma)(i,j), \tag{4}$$

where G_σ denotes a 2D Gaussian kernel with standard deviation σ, defined as:

$$G_\sigma(x,y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \tag{5}$$

The kernel is centered around $(x,y) = (0,0)$, and the convolution is applied across all pixels in the input.

3.2.4 Values

To systematically evaluate the robustness of our method under varying data corruptions, we consider three severity levels (mild, moderate, and strong) based on the corresponding parameter values for each type of noise. Intensity values for the Salt-and-Pepper noise are derived by the work of Wan et al.[42] Instead, Gaussian blur the values are adapted from the work Yoshihara et al.[43] Here, the moderate and strong levels are set to half of those used by the author[†]. In the case of Gaussian noise, values are decided through preliminary experiments, based on comparable perceived degradation in relation with the other pollution types. An overview of the parameter settings for each level is provided in Table 1. Figure 1 shows as an example the impact of the respective parameter settings for the class plane.

Table 1: Noise and level of pollution categorized as mild, moderate, and strong, with associated parameters. For the Salt-and-Pepper noise, p_{salt} and p_{pepper} contribute equally. For example, when $p_{\text{salt\_and\_pepper}} = 0.1$ (mild), corresponds to $p_{\text{salt}} = p_{\text{pepper}} = 0.05$.

Type	**Mild**	**Moderate**	**Strong**
Gaussian noise (σ)	0.1	0.3	0.5
Salt-and-Pepper noise ($p_{\text{salt\_and\_pepper}}$)	0.05	0.1	0.2
Gaussian blur (σ_{blur})	0.5	1.0	2.0

3.3 Evaluation Metrics and Analysis

Performance is primarily evaluated using classification accuracy on the clean test set and on test sets with varying the noise levels, to assess model robustness. Additionally, training and validation loss curves are analyzed to understand convergence behavior under different noise pollution levels. Results are aggregated over the 10 runs per pollution level to calculate mean and standard deviation for all metrics (see also Sect. 3.1.3). The main focus is to plot, compare, and assess the accuracy and loss as functions of the training data pollution percentage, revealing how incorporating noisy data during training influences robustness against noisy test inputs.

4. RESULTS

4.1 Loss

This section describes the results for the three different noise types in terms of test loss.

[†]By doing so, we can sufficiently demonstrate our method.

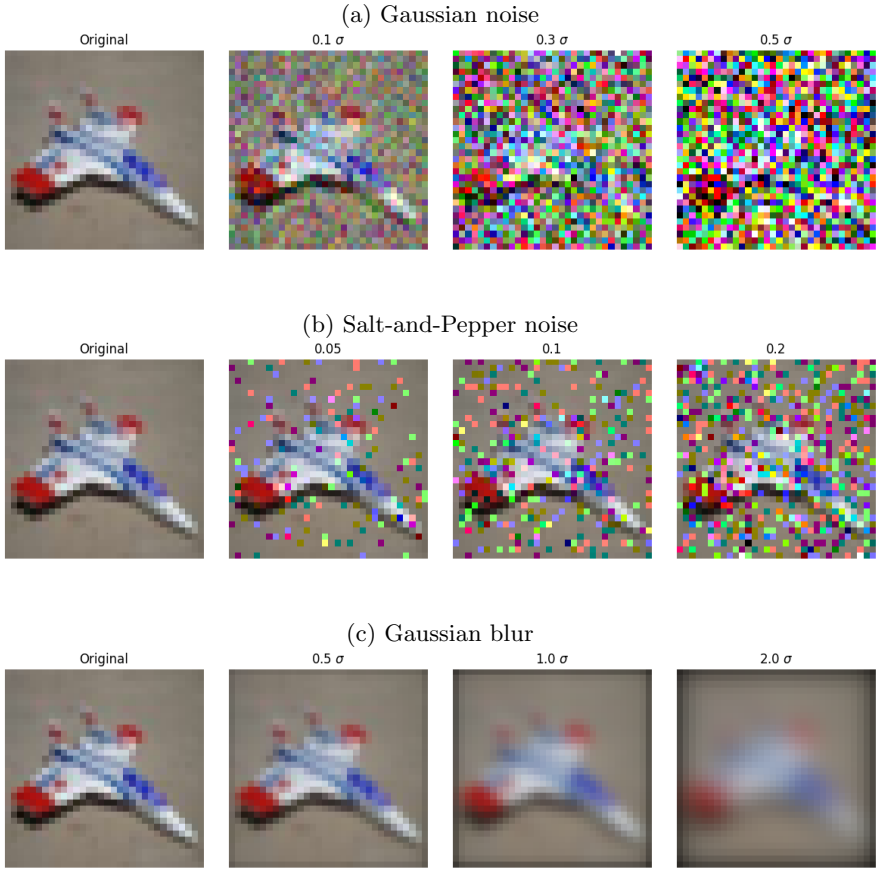


Figure 1: Impact of different types and intensities of image degradation on CIFAR-10.

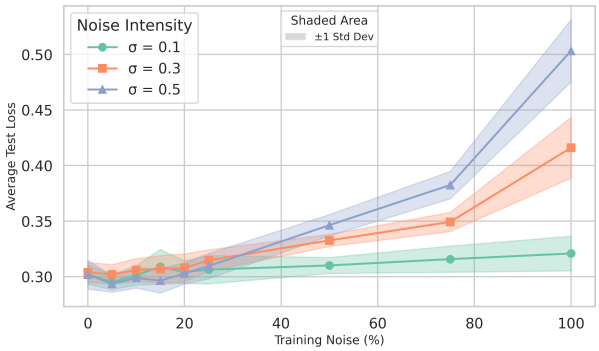
4.1.1 Gaussian Noise

Figure 2 shows the average test loss of the model training on CIFAR-10 data with varying intensities of Gaussian noise $\sigma \in \{0.1, 0.3, 0.5\}$. Figure 2a shows the test performance for the three levels of Gaussian noise on the clean test set, revealing that models trained with small amounts of Gaussian noise ($\sigma = 0.1$) maintain the lowest average test loss even as training noise levels increase. In contrast, higher-intensity noise ($\sigma = 0.5$) begins to degrade performance, especially at higher training noise percentages.

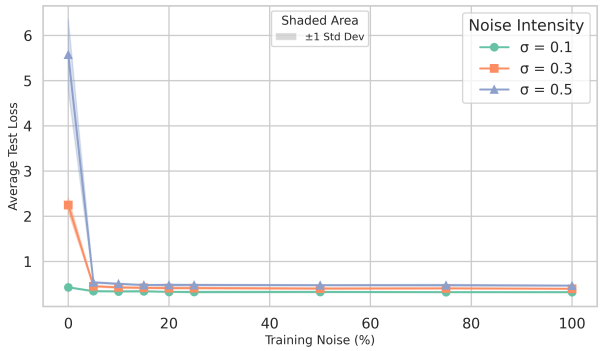
Figure 2b demonstrates the model's performance on the corresponding fully corrupted test set. A small amount of noise augmentation during training (as low as 5%) leads to a substantial reduction in test loss, indicating that robustness improves with moderate exposure. However, training with excessively strong noise ($\sigma = 0.5$) results in higher test loss, suggesting diminishing returns or over-regularization. Overall, these plots highlight the trade-off between robustness and clean-data performance as a function of both noise intensity and training exposure. Moderate augmentation with low-intensity noise ($\sigma = 0.1$ or 0.3) provides the best balance.

4.1.2 Salt-and-Pepper Noise

Figure 3 presents the average test loss of the model trained on CIFAR-10 data with varying levels of Salt-and-Pepper noise applied to the training set. Results are shown for three intensities $p_{\text{salt\_and\_pepper}} \in \{0.05, 0.1, 0.2\}$ and are evaluated in the same manner as in Section 4.1.1: on clean test data (left) and fully corrupted test data (right). On clean test data, as shown in Figure 3a, increasing the proportion of noisy images in the training set generally results in higher test loss, with a more pronounced effect at higher noise levels. The model trained with $p_{\text{salt\_and\_pepper}} = 0.2$ consistently exhibits the highest loss, indicating a greater susceptibility to overfitting on noisy features.

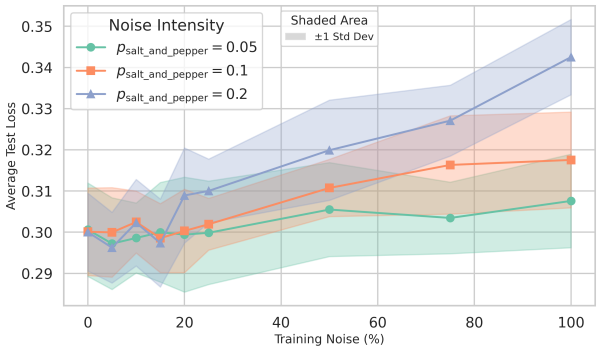


(a) Loss of the three different intensities of Gaussian noise tested on the clean test set.

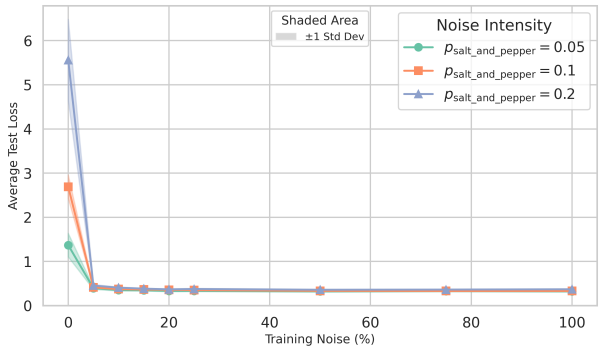


(b) Loss of the three different intensities of Gaussian noise tested on the maximum noisy test set.

Figure 2: Average test loss across different Gaussian noise intensities and varying training noise percentages. Shaded areas indicate ±1 standard deviation over 10 runs.



(a) Loss of the three different intensities of Salt-and-Pepper noise tested on the clean test set.



(b) Loss of the three different intensities of Salt-and-Pepper noise tested on the maximum noisy test set.

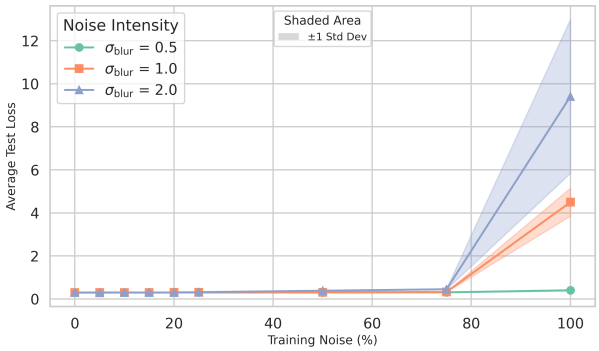
Figure 3: Average test loss across different Salt-and-Pepper noise intensities and varying training noise percentages. Shaded areas indicate ±1 standard deviation over 10 runs.

Conversely, when evaluated on fully corrupted test data, as shown in Figure 3b, the trend reverses: models trained without noise perform poorly, while those trained with any of the tested noise intensities achieve significantly lower test loss. The most notable improvement occurs when increasing training noise from 0% to 10%, after which performance levels off across all intensities. These results suggest that adding a small amount of noise to the training data acts as an effective regularizer, enhancing the model's robustness to real-world corruptions. Overall, the findings underscore the importance of aligning training conditions with deployment scenarios when robustness to noise is required.

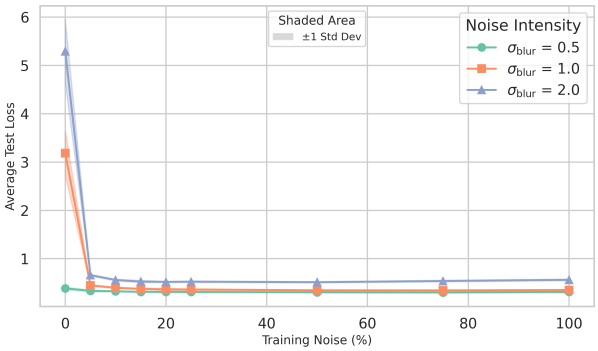
4.1.3 Gaussian Blur

Figure 4 presents the average test loss of the model trained on CIFAR-10 data with varying levels of Gaussian blur applied to the training set. The results are shown for three different noise intensities, $\sigma_{\text{blur}} \in \{0.5, 1.0, 2.0\}$, and are evaluated as described in the previous sections. On clean test data, as shown in Figure 4a, increasing the proportion of noisy images in the training set generally leads to higher test loss, with the model trained with $\sigma_{\text{blur}} = 2.0$ consistently exhibiting the highest loss. This indicates its vulnerability to overfitting on blurred features.

On the contrary, when evaluated on fully corrupted test set shown in Figure 4b, the trend is reversed: models trained with low or no noise perform poorly, while those trained with moderate noise intensities ($\sigma_{\text{blur}} = 1.0$ and 2.0) achieve significantly lower test loss. The most substantial improvement occurs when the noise is of the level



(a) Loss of the three different intensities of Gaussian blur tested on the clean test set.



(b) Loss of the three different intensities of Gaussian blur tested on the maximum noisy test set.

Figure 4: Average test loss across different Gaussian blur intensities and varying training noise percentages. Shaded areas indicate ±1 standard deviation over 10 runs.

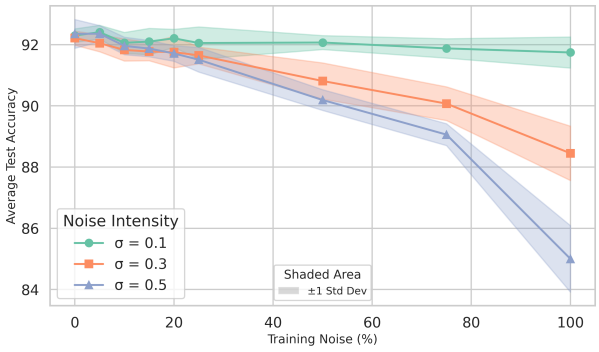
of 5% in the training noise, after which the performance plateaus across all intensities. Similarly to the results of the previous sections, these results also indicate that adding a small degree of noise to the training data can act as a beneficial form of regularization, helping the model better withstand input degradation. This underscores the value of tailoring training conditions to reflect the types of noise and variability expected during deployment, especially when robustness is critical.

4.2 Accuracy

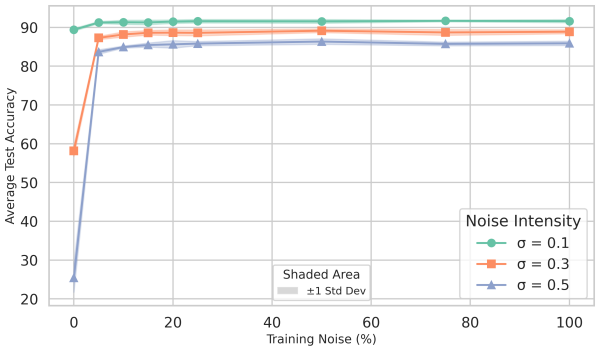
This section describes the results for the three different noise types in terms of test accuracy.

4.2.1 Gaussian Noise

Figure 5 shows the average test accuracy of the model trained on CIFAR-10 data with varying intensities of Gaussian noise. Figure 5a presents the test performance on the clean test set for these three noise levels, revealing that the model trained with the lowest noise intensity ($\sigma = 0.1$) maintains a relatively stable performance. In contrast, higher noise intensities ($\sigma = 0.3$ and 0.5) begin to degrade accuracy, particularly as the proportion of noisy training data increases.



(a) Effect of training noise on fully clean test performance.



(b) Effect of training noise on fully polluted test performance.

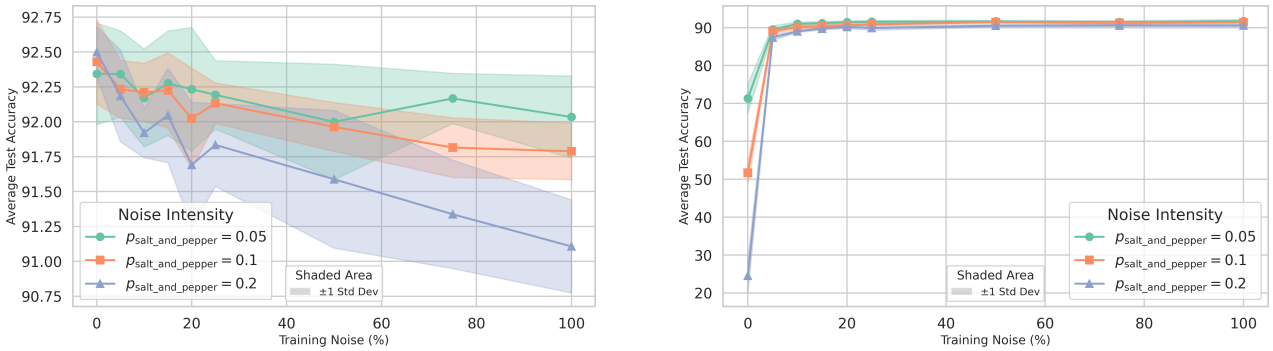
Figure 5: Average test accuracy across different Gaussian noise intensities and varying training noise percentages. Shaded areas indicate ±1 standard deviation over 10 runs.

Figure 5b demonstrates the model's performance with the respective fully polluted test set. Here, a small amount of noise augmentation during training (as low as 5%) leads to a substantial improvement in the accuracy,

showing that robustness improves with moderate exposure. Additionally, training with excessively strong noise ($\sigma = 0.5$) leads to a reduction in the average accuracy of the fully polluted (with the respective intensity) test set. Here the results also suggest that moderate augmentation with low-intensity noise ($\sigma = 0.1$ or 0.3) achieves the best balance as shown in Figure 2. For the strongest intensity ($\sigma = 0.5$), and also considering the results in Sect. 4.1.1, the trained model could be useful for extreme polluted test sets, as it delivers a better performance when compared to the model trained with only clean data.

4.2.2 Salt-and-Pepper Noise

Figure 6 shows the impact of Salt-and-Pepper noise across varying noise intensities and proportions of corrupted training data, ranging from 0% to 100%. Figure 6a presents the average test accuracy on a clean test set. As the proportion of noisy training data increases, performance on clean data remains relatively stable, with only a slight drop at higher noise intensities. Models trained with low-intensity noise ($p_{\text{salt\_and\_pepper}} = 0.05$) maintain an accuracy around 92%, while those trained with higher-intensity noise ($p_{\text{salt\_and\_pepper}} = 0.2$) decrease modestly to about 91%. In contrast, Figure 6b illustrates performance on a fully corrupted test set, where models trained without noise perform poorly, particularly for $p_{\text{salt\_and\_pepper}} = 0.2$, achieving only about 25% accuracy. However, introducing even a small proportion of noisy training data (10 to 20%) leads to a considerable improvement in robustness, with all models reaching approximately 91% accuracy. These results demonstrate that noise augmentation can substantially enhance robustness to test-time corruption, with minimal impact on clean-data performance.



(a) Effect of training noise on fully clean test performance.

(b) Effect of training noise on fully polluted test performance.

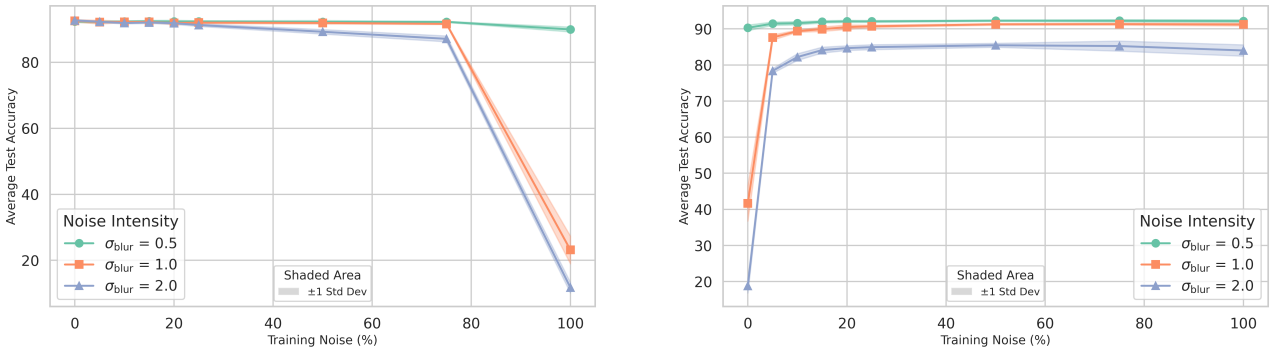
Figure 6: Average test accuracy across different Salt-and-Pepper noise intensities and varying training noise percentages. Shaded areas indicate ±1 standard deviation over 10 runs.

4.2.3 Gaussian Blur

Figure 7 shows the impact of Gaussian blur across varying noise intensities and proportions of corrupted training data, ranging from 0% to 100%. Figure 7a presents the average test accuracy on a clean test set. As the proportion of noisy training data increases, performance on clean data degrades, particularly for higher noise intensities. Models trained with low-intensity blur ($\sigma_{\text{blur}} = 0.5$) maintain relatively stable accuracy around 92%, while those trained with higher-intensity blur ($\sigma_{\text{blur}} = 1.0$ and 2.0) drop around 20%. In contrast, Figure 7b illustrates performance on a fully corrupted test set, where models trained without noise perform poorly, especially for $\sigma_{\text{blur}} = 2.0$, achieving only about 20% accuracy. However, introducing even a small proportion of noisy training data (10 to 20%) leads to a substantial improvement in robustness, as for moderate levels of blur ($\sigma_{\text{blur}} = 0.5$ and 1.0) the accuracy reaches approximately 90% and for the strongest one ($\sigma_{\text{blur}} = 2.0$) the accuracy is above 80%. These results show that training with the three different levels of blur can significantly enhance robustness.

5. DISCUSSION

To analyze the impact of different noise types on model robustness, we examine the average test loss on CIFAR-10 as the training inputs are progressively corrupted. For this analysis, we select the highest tested intensity



(a) Effect of training noise on fully clean test performance.

(b) Effect of training noise on fully polluted test performance.

Figure 7: Average test accuracy across different Gaussian blur intensities and varying training noise percentages. Shaded areas indicate ±1 standard deviation over 10 runs.

for each noise type: Gaussian noise ($\sigma = 0.5$), Salt-and-Pepper noise ($p_{\text{salt\_and\_pepper}} = 0.2$), and Gaussian blur ($\sigma_{\text{blur}} = 2.0$).

5.1 Loss

As shown in Figure 8, all three noise types, for the highest intensity respectively, exhibit a pronounced drop in test loss with as little as 5% of the training data being polluted. This early drop suggests that exposure to small amounts of input noise can act as an implicit regularizer, improving generalization.

Among the three perturbations, Salt-and-Pepper noise consistently yields the lowest test loss across all levels of input corruption, indicating greater robustness to this type of discrete noise. Gaussian noise performs similarly well, whereas Gaussian blur results in slightly higher test loss, particularly at lower corruption levels. However, the differences between the methods diminish beyond 10% corruption, where all curves flatten, suggesting the model's ability to adapt to noisy training data once a certain exposure threshold is reached.

5.2 Accuracy

In Figure 9 all three noise types exhibit a pronounced enhancement in accuracy loss with as little as 5% of the training data being polluted. This suggests again that exposure to small amounts of input noise can act as an implicit regularizer, improving generalization. Among the noise types, Salt-and-Pepper noise at highest intensity yielded the most robust performance across pollution levels, while Gaussian blur showed the least benefit, particularly at low exposure. Nevertheless, all models exhibited a convergence in performance after about 10% of noisy training data, highlighting this threshold as sufficient to trigger meaningful robustness without compromising clean-data accuracy.

5.3 Additional experiments on pollution intensity

Although not reported in the main paper, we are conducting new experiments with stronger intensity values of Salt-and-Pepper noise and Gaussian blur (i.e., $p_{\text{salt\_and\_pepper}} = 0.4$, $\sigma_{\text{blur}} = 4.0$). The preliminary results for loss and accuracy on both the clean and fully polluted test sets remained unchanged. On the clean test set, the loss tends to increase with higher training noise. Accuracy continues to decrease slightly as training noise increases, but the effect is not significant. For the fully polluted test set, the loss is reduced at higher training noise levels, and accuracy improves starting from approximately 5–10% training noise. Overall, the observed behavior is consistent with the results presented in Section 4.2.

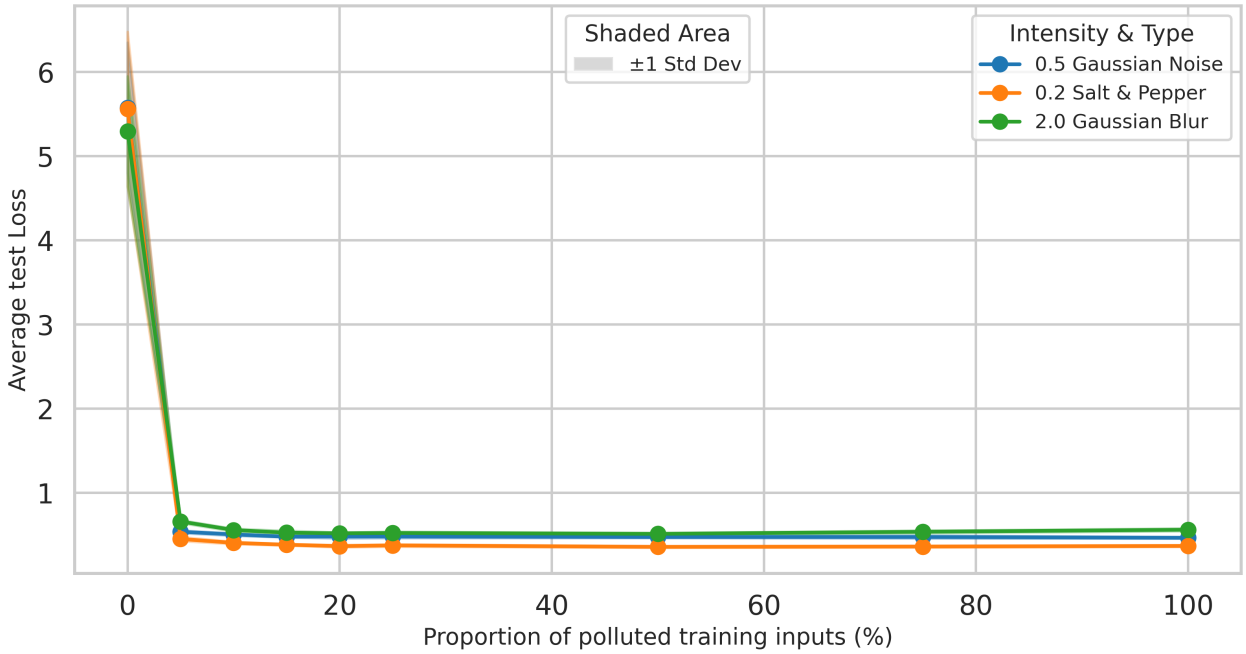


Figure 8: Impact of training noise on all pollution types on highest intensity values in fully polluted testing conditions for the average test Loss.

5.4 Comparison to SOTA

We build on prior work that uses data augmentation to improve model robustness. De Vries et al.[31] proposed Cutout, a regularization technique that removes rectangular regions from training images. Although not specifically designed for robustness against corruptions, Cutout improves generalization and enhances clean-data accuracy. Later, Lopes et al.[16] introduced Patch Gaussian augmentation, which injects Gaussian noise into localized patches rather than the entire image. Their method preserves clean-image accuracy while improving robustness to localized corruptions. In contrast, we apply global noise to entire images and evaluate different noise types across a range of intensities and proportions. Our results confirm that even small amounts of noise whether structured or unstructured help CNNs generalize to noisy conditions. While Cutout and Patch Gaussian focus on structural occlusion or local noise, our approach shows that typical sensor degradations like blur and impulse noise can be just as effective for improving robustness. Although our experiments show a degradation in accuracy for higher noise intensities, the findings collectively underscore the value of training with imperfect inputs.

5.5 Balancing Clean Data and Real-World Robustness

This experimental setup indicate that incorporating a degree of noise into the training data can significantly enhance model robustness, especially when test images are subject to similar forms of degradation. These findings suggest that strategically managing data quality by balancing clean and noisy data can lead to more reliable and resilient models suitable for deployment in real-world conditions.

These findings invite a critical reevaluation of classical definitions of data quality, which have traditionally emphasized cleanliness, consistency, and minimal noise. While such criteria aim to optimize model training under ideal conditions, our results suggest that strict adherence to these standards may limit model robustness in real-world scenarios. Introducing controlled amounts of representative noise may slightly deviate from conventional data quality metrics, but it offers clear benefits for generalization to noisy, imperfect inputs. This approach is particularly valuable in safety-critical applications, where operational environments are often unpredictable and robustness is essential.

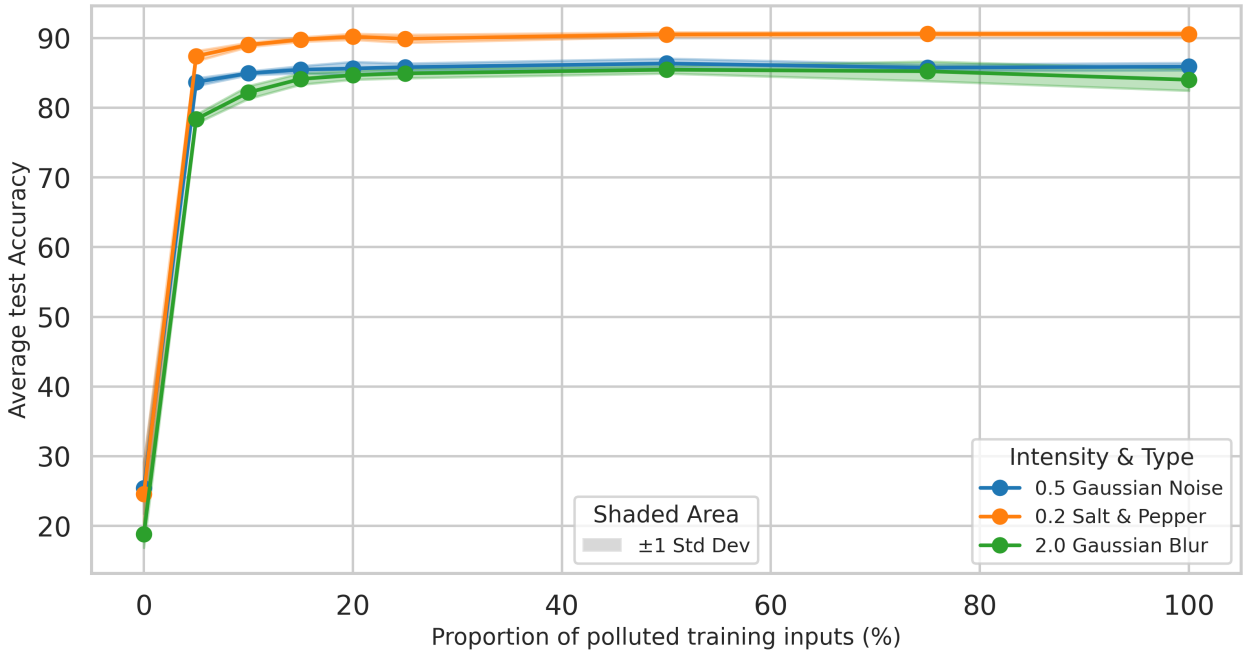


Figure 9: Impact of training noise on all pollution types on highest intensity values in fully polluted testing conditions for the average test accuracy.

6. CONCLUSION

This study presents an empirical investigation into how the inclusion of noisy data during training influences the robustness of CNNs in image classification tasks. To this end, Gaussian noise, Salt-and-Pepper noise, and Gaussian blur were systematically introduced into varying proportions of the CIFAR-10 training dataset. Model performance was then evaluated on both clean and degraded test inputs. The findings consistently indicate that even limited exposure to noise during training, typically as low as 5%, can substantially reduce test loss. Notably, incorporating just 10% noisy data during training significantly enhances robustness, leading to improved accuracy on corrupted test samples.

Overall, the results challenge traditional assumptions regarding the necessity of purely clean datasets for training high-performing models. Instead, they support a more nuanced perspective in which moderate and representative imperfections during training act as a form of regularization, thereby improving model resilience under real-world conditions. These insights highlight the importance of reevaluating data quality criteria in the development of AI systems, particularly for safety-critical or real-world applications where input imperfections are the norm.

7. ACKNOWLEDGMENTS

I would like to thank the Data Quality team at the DLR Institute for AI Safety and Security for their valuable feedback. In particular, I am grateful to Leonie Etzold for the constructive discussions, which have significantly contributed to improving this work.

APPENDIX A. CIFAR-10

A.1 Loss clean dataset

To assess the impact of non-determinism during training, we trained the model on CIFAR-10 over 10 independent runs and report the mean and standard deviation of the training and validation loss across 200 epochs (see

Figure 10). The training loss consistently decreases across all runs, exhibiting a smooth exponential decay with negligible variance, and converges near zero. This indicates that the model is able to fit the training data well, and that training randomness, such as random initialization, data shuffling, and dropout, has minimal impact on the convergence of the training loss.

In contrast, the validation loss decreases rapidly during the initial 20–30 epochs, then continues to decrease at a slower rate before plateauing around epoch 100. The final validation loss stabilizes around 0.25 with a noticeably higher variance compared to the training loss, especially in the early and intermediate epochs. This behavior reflects the model's generalization capacity and highlights a moderate sensitivity to non-deterministic factors, which primarily affect validation performance rather than training convergence. The increasing gap between training and validation loss over time suggests a degree of overfitting, which is typical for deep models trained on CIFAR-10 without aggressive regularization. These results illustrate that while the optimization dynamics are largely robust to stochastic effects, the model's generalization is more susceptible to variability introduced by non-determinism in the training pipeline.

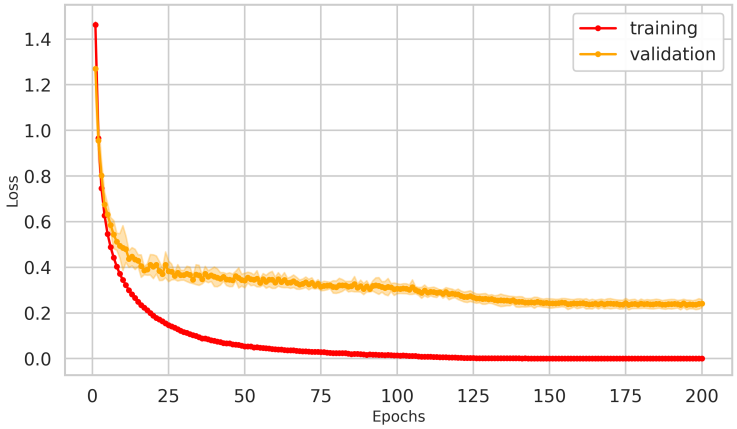


Figure 10: Loss single clean dataset.

A.2 Accuracy clean dataset

Figure 11 illustrates the training and validation accuracy over 200 epochs, averaged across 10 independent runs, with shaded regions indicating the standard deviation. The training accuracy exhibits a steady increase throughout training, rapidly approaching near-perfect performance (>99%) and plateauing after approximately 100 epochs. This reflects the model's strong capacity to fit the CIFAR-10 training data, with minimal variability across runs, confirming the robustness of the optimization process under stochastic conditions.

The validation accuracy also improves sharply during the early epochs, reaching approximately 90% within the first 30 epochs, and continues to increase more gradually thereafter. It ultimately stabilizes around 93%, with a standard deviation slightly larger than that of the training accuracy. This suggests that while the model generalizes well, its performance on unseen data is more sensitive to random factors such as initialization and data ordering. The persistent gap between training and validation accuracy indicates mild overfitting, consistent with the trends observed in the loss curves. Nonetheless, the relatively narrow variance bands demonstrate that the generalization behavior remains stable across repeated runs, highlighting that while non-determinism introduces some variability, it does not significantly undermine the model's overall performance on the validation set.

Finally, Table 2 shows the final test accuracy for the 10 different classes of CIFAR-10 for 100 and 200 epochs, respectively. Given the marginal improvement between the two configurations, we fixed the number of training epochs to 100 epochs for the experiments presented in this paper.

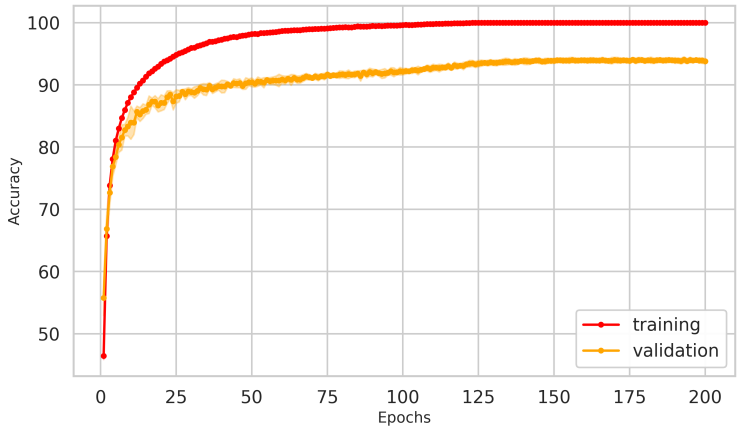


Figure 11: Accuracy single clean dataset.

Table 2: Final classification accuracy per class on the CIFAR-10 for 100 and 200 epcohs.

Class	**Accuracy (%)** (100 epochs)	**Accuracy (%)** (200 epochs)
Plane	94.30	95.60
Car	96.50	98.10
Bird	89.80	91.40
Cat	83.70	87.70
Deer	93.80	95.00
Dog	88.60	89.60
Frog	95.60	96.40
Horse	95.80	95.80
Ship	95.00	96.60
Truck	95.80	96.00
Final test accuracy	**92.89**	**94.22**

REFERENCES

[1] Krizhevsky, A., Sutskever, I., and Hinton, G. E., "Imagenet classification with deep convolutional neural networks," *Communications of the ACM* **60**(6), 84–90 (2017).

[2] He, K., Zhang, X., Ren, S., and Sun, J., "Deep residual learning for image recognition," in [*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*], (2016).

[3] LeCun, Y., Bengio, Y., and Hinton, G., "Deep learning," *Nature* **521**(7553), 436–444 (2015).

[4] Rawat, W. and Wang, Z., "Deep convolutional neural networks for image classification: A comprehensive review," *Neural Computation* **29**(9), 2352–2449 (2017).

[5] Bengio, Y., Courville, A., and Vincent, P., "Representation learning: A review and new perspectives," *IEEE TPAMI* **35**(8), 1798–1828 (2013).

[6] Sun, C., Shrivastava, A., Singh, S., and Gupta, A., "Revisiting unreasonable effectiveness of data in deep learning era," *ICCV* (2017).

[7] Recht, B., Roelofs, R., Schmidt, L., and Shankar, V., "Do imagenet classifiers generalize to imagenet?," *ICML* (2019).

[8] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L., "Imagenet: A large-scale hierarchical image database," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* , 248–255 (2009).

[9] Krizhevsky, A. and Hinton, G., "Learning multiple layers of features from tiny images," tech. rep., University of Toronto (2009).

[10] Dodge, S. and Karam, L., "Understanding how image quality affects deep neural networks," in [*International Conference on Quality of Multimedia Experience (QoMEX)*], (2016).
[11] Geirhos, R., Temme, C. R. M., Rauber, J., Schütt, H. H., Bethge, M., and Wichmann, F. A., "Generalisation in humans and deep neural networks," *NeurIPS* (2018).
[12] Karahan, O. T. and et al., "Image quality assessment on face recognition systems," in [*SIU*], (2016).
[13] Hendrycks, D. and Dietterich, T., "Benchmarking neural network robustness to common corruptions and perturbations," *International Conference on Learning Representations (ICLR)* (2019).
[14] Ford, J. and et al., "Adversarial examples are a natural consequence of test error in noise," *arXiv preprint arXiv:1901.10513* (2019).
[15] Taori, R., Dave, A., Shankar, V., and et al., "Measuring robustness to natural distribution shifts in image classification," *NeurIPS* (2020).
[16] Lopes, R. G., Yin, D., Poole, B., Gilmer, J., and Cubuk, E. D., "Improving robustness without sacrificing accuracy with patch gaussian augmentation," (2019).
[17] Shorten, C. and Khoshgoftaar, T. M., "A survey on image data augmentation for deep learning," *Journal of Big Data* **6**(1), 60 (2019).
[18] Zhang, H., Cisse, M., Dauphin, Y., and Lopez-Paz, D., "mixup: Beyond empirical risk minimization," in [*ICLR*], (2018).
[19] Xie, C. and et al., "Adversarial examples improve image recognition," in [*CVPR*], (2020).
[20] Liu, T. C. and et al., "A simple framework for contrastive learning of visual representations," in [*ICML*], (2020).
[21] Rusak, E. and et al., "A simple way to make neural networks robust against diverse image corruptions," in [*ECCV*], (2020).
[22] Madry, A. and et al., "Towards deep learning models resistant to adversarial attacks," in [*ICLR*], (2018).
[23] Northcutt, C. G. and et al., "Pervasive label errors in test sets destabilize machine learning benchmarks," *arXiv preprint arXiv:2103.14749* (2021).
[24] Zhang, B. and et al., "Understanding robustness and accuracy of data augmentation," in [*ICML*], (2021).
[25] Beery, S., Van Horn, G., and Perona, P., "Recognition in terra incognita," in [*ECCV*], (2018).
[26] A., A. and B., A., "Dataset curation and quality assurance in machine learning," *Journal of ML Systems* **10**(2), 123–134 (2020).
[27] C., A. and D., A., "Impact of data quality on machine learning models," *International Journal of Data Science* **5**(1), 45–56 (2019).
[28] E., A. and F., A., "Annotation consistency in image datasets," *Computer Vision Journal* **15**(4), 200–210 (2018).
[29] I., A. and J., A., "Noise robustness in convolutional neural networks," *Neural Networks* **134**, 123–135 (2021).
[30] K., A. and L., A., "Image degradation and classification performance," *Journal of Visual Computing* **28**(6), 450–460 (2020).
[31] DeVries, T. and Taylor, G. W., "Improved regularization of convolutional neural networks with cutout," (2017).
[32] M., A. and N., A., "Data augmentation for robustness in image classification," *Pattern Recognition Letters* **120**, 45–53 (2019).
[33] Goodfellow, I. J., Shlens, J., and Szegedy, C., "Explaining and harnessing adversarial examples," *International Conference on Learning Representations (ICLR)* (2015).
[34] Fawzi, A., Fawzi, O., and Frossard, P., "Analysis of classifiers' robustness to adversarial perturbations," *Machine Learning* **107**, 481–508 (2018).
[35] Hendrycks, D. and Gimpel, K., "A baseline for detecting misclassified and out-of-distribution examples in neural networks," *International Conference on Learning Representations (ICLR)* (2017).
[36] Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D., "mixup: Beyond empirical risk minimization," *International Conference on Learning Representations (ICLR)* (2018).
[37] S., A. and T., A., "Curriculum learning strategies for noisy data," *International Journal of Machine Learning* **11**(2), 200–215 (2020).

[38] U., A. and V., A., "Noise-aware training objectives for robust deep learning," *Journal of AI Research* **19**(3), 333–345 (2022).

[39] W., A. and X., A., "Adaptive noise filtering techniques in cnn training," *Signal Processing Letters* **28**(7), 600–610 (2021).

[40] O., A. and P., A., "Domain adaptation methods for image classification," *Machine Learning Review* **14**(1), 12–25 (2022).

[41] Sontakke, M. D. and Kulkarni, M. S., "Different types of noises in images and noise removing technique," *International Journal of Advanced Technology in Engineering and Science* **3**, 102–115 (jan 2015).

[42] Wan, B., Zhou, X., Sun, Y., Wang, T., Lv, C., Wang, S., Yin, H., and Yan, C., "Adnet: Anti-noise dual-branch network for road defect detection," *Engineering Applications of Artificial Intelligence* **132**, 107963 (jun 2024).

[43] Yoshihara, S., Fukiage, T., and Nishida, S., "Does training with blurred images bring convolutional neural networks closer to humans with respect to robust object recognition and internal representations?," *Frontiers in Psychology* **14**, 1047694 (feb 2023).