Examining Input Noise Injection During Dropout for Robust CIFAR Dataset Image Recognition

Chriss Jordan Oboa Franck
Georgetown University
Washington, DC, USA
cfo17@georgetown.edu

Shuming Mao
Georgetown University
Washington, DC, USA
sm3828@georgetown.edu

ABSTRACT
Deep learning models, particularly Convolutional Neural Networks (CNNs), have demonstrated remarkable performance in image recognition tasks. However, they still have vulnerability to adversarial attacks and sensitivity to noise of input data. In this paper, we investigate the impact of input noise injection during dropout regularization on the robustness of AlexNet for image classification tasks. We conduct experiments using the CIFAR-10 and CIFAR-100 datasets to evaluate AlexNet's resilience to adversarial attacks and its classification accuracy under different noise injection scenarios. Our findings suggest that injecting noise during dropout training enhances AlexNet's robustness to adversarial attacks and improves its ability to classify images in noisy environments more accurately.

Keywords
Deep Learning, CNN, Dropout, Error Injection, Gaussian Noise

1. INTRODUCTION
Convolutional neural networks (CNNs) have revolutionized image recognition tasks, achieving state-of-the-art performance on various benchmark datasets. Among the early CNN architectures, AlexNet stands out for its pioneering design and remarkable success in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012. Despite its effectiveness, CNNs like AlexNet are susceptible to adversarial attacks, where small, imperceptible perturbations to input images can lead to misclassification.
To address this vulnerability and enhance model robustness, regularization techniques such as dropout have been widely employed. Dropout randomly deactivated neurons during training, forcing the network to learn more robust and generalizable features. However, the effectiveness of dropout regularization can be further enhanced by injecting noise into the input data during dropout.

In this paper, we aim to analyze how injecting noise affects the behavior of dropped neurons during training and its subsequent influence on model performance. To evaluate the effectiveness of our approach, we utilize the CIFAR-10 and CIFAR-100 datasets, which consist of 32x32 color images across ten and one hundred classes, respectively.

Our experiment aims to provide insights into leveraging input noise injection to improve the resilience of deep learning models, particularly AlexNet, to adversarial attacks and noisy input data. By comprehensively analyzing the effects of noise injection on model performance, we seek to contribute to the development of more robust and reliable image recognition systems.

However, in our experiment, AlexNet was not giving accurate results enough for a standard version that we needed to test on. As we needed something more efficient without our time range, we switched to this model, that we could call "CIFAR 10 net"

2. Related Work and Background

2.1 Adversarial Attacks and Defense Approach
Adversarial attacks have posed a significant threat to the reliability and security of deep learning models. These attacks perturb input data to cause misclassification, and are often imperceptible to human observers. Various adversarial attack methods have been proposed, including Fast Gradient Sign Method (FGSM) [1], DeepFool [2], and Carlini-Wagner attack [3]. These attacks exploit the vulnerability of neural networks to small perturbations in input data, leading to erroneous predictions.

In response to the growing concern over adversarial attacks, numerous defense mechanisms have been proposed. One prominent approach is adversarial training, where models are trained on adversarially perturbed data to improve robustness [4]. Other defense strategies include input preprocessing techniques, such as feature squeezing and input denoising [5], and gradient masking methods to hide gradient information from attackers [6]

2.2 Defense with Noise
In recent years, defense mechanisms incorporating noise have raised attention due to their effectiveness in enhancing model robustness. Injecting random noise into input data during training has been shown to improve model generalization and mitigate the impact of adversarial attacks [7]. Noise injection techniques include random perturbations, Gaussian noise addition, and dropout

regularization [8]. One notable approach is adversarial training with noise augmentation, where models are trained on a combination of clean and noisily perturbed data [9]. This technique aims to expose the model to a diverse range of input variations, leading to improved generalization and robustness.

2.3 Robustness and Efficiency

It ‘s crucial to ensure both robustness and efficiency for deploying deep learning models in real-world applications. Robust models should maintain high accuracy and resilience to adversarial attacks without sacrificing much computational efficiency. To achieve such balance, we are required to design well-organized model architecture, training procedures, and defense mechanisms.

Recent studies have explored the trade-off between model robustness and efficiency. Techniques such as compact model architectures [10], knowledge distillation [11], and pruning methods [12] aim to reduce model complexity while preserving performance and robustness.

3. Methodology

3.1 Experimental Design

3.1.1 Model Architecture Modifications

Due to the limited computational resources and the need for efficient processing, we adapted the traditional AlexNet architecture to better suit the lower resolution images in the CIFAR-10 dataset, we call that “CIFAR 10 Net”. This modified architecture was designed with the following specific changes aimed at enhancing processing efficiency and adaptability to CIFAR-10's image properties:

- **Depth and Complexity Reduction:** The original depth of AlexNet was reduced to better match the complexity of the CIFAR-10 dataset, which contains simpler and smaller images compared to ImageNet. This adjustment helped in managing the computational load without significantly compromising the model's ability to capture relevant features of images.
- **Parameter Tuning:** We did extensive tuning of the model's hyperparameters, including learning rates, batch sizes, and layer configurations. These modifications were critical in optimizing the model's performance given the constraints imposed by our computational budget.

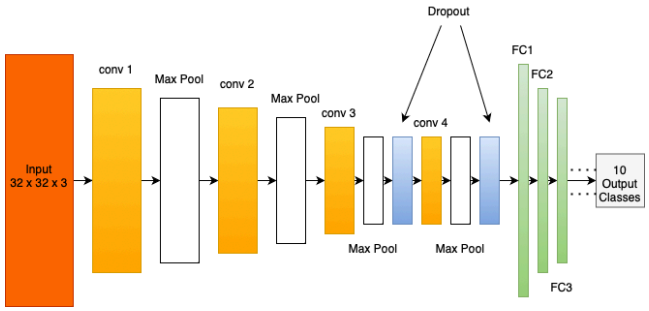

Figure 1: Architecture of CIFAR10Net - a streamlined model that is well-suited for the CIFAR-10 dataset's requirements.

As shown in Figure 1, the network begins with an initial convolutional layer (conv1) that processes the 32x32 RGB images using a 5x5 filter. This layer is followed by a max pooling layer to reduce spatial dimensions while retaining the most critical features. Subsequent convolutional layers (conv2, conv3, and conv4) with increased filter depth and identical kernel sizes further extract complex features. Dropouts are applied in the third and Fourth layer. Max pooling layers between the convolutional layers continue the down-sampling process. The final stages of the network consist of three fully connected layers (FC1, FC2, and FC3), forming a 10-class output layer activated by a sigmoid function for classification.

3. 1. 2 CIFAR-10 Dataset

The CIFAR-10 dataset, comprising 60,000 32x32 color images distributed across 10 classes, was selected as the test sample for our experiments. The diversity of the dataset and its balanced nature provided a robust framework for evaluating its classification capabilities of CIFAR10Net under varying noise conditions.

3.2 Noise Injection Strategy

3. 2. 1 Adaptations Due to Computational Limits

To enhance the robustness of CIFAR10Net against adversarial attacks and noisy environments, we incorporated a stochastic noise injection mechanism during the dropout phases of the training process. This strategy was designed to balance robustness with computational efficiency:

Simplified Noisy Dropout: A custom dropout layer, we called it NoisyDropout, was implemented in our experiment. This layer not only deactivates neurons randomly during training, as per the standard dropout technique, but also injects Gaussian noise into the deactivated neurons, which simulates an adversarial environment during training.

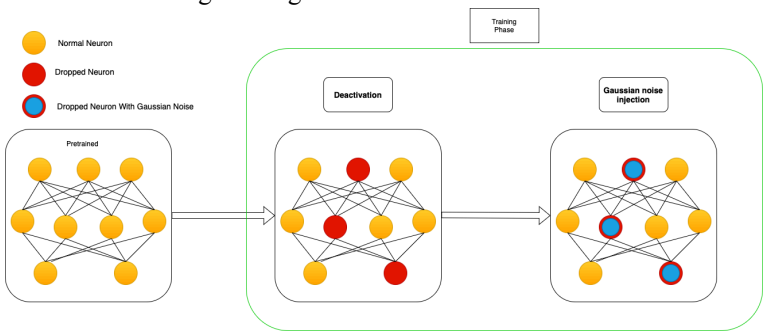

Figure 2: Gaussian Noise Injection to Dropped Neuron during Training Phase

Process: We first determined the standard deviation of the dataset, which was 0.1595, and used this value as the base rate for applying noise. Noise levels were introduced at 5%

(0.007975), 10% (0.01595), 30% (0.04785), and 50% (0.07975) relative to this base rate. For dropout probabilities, we applied the same series of noise levels across three different settings: at a 20% dropout probability, we employed noise rates of 5%, 10%, 30%, 50%, and 100% of the base rate; this pattern was consistently used for dropout probabilities of 25% and 30% as well.

3. 2. 2 Mathematical Representation of Noisy Dropout

Mathematically, the standard dropout operation can be represented as follows:

Let x be the input vector to a layer, and $\boldsymbol{d}$ be a binary mask vector of the same size as x, where each element is set to 1 with probability p and 0 with probability $1-p$. Then, the dropout operation is given by:

$$\text{dropout}(x) = x \odot \boldsymbol{d} \qquad (1)$$

where $\odot$ denotes element-wise multiplication.

Noisy dropout extends this by introducing random noise to the dropout mask. Let ϵ be a random noise vector drawn from some distribution, typically a Gaussian distribution with mean 0 and variance σ^2, and $\boldsymbol{d}$ be the binary dropout mask. Then, the noisy dropout operation can be represented as:

$$\text{noisy\_dropout}(x) = (x \odot \boldsymbol{d}) + \epsilon \qquad (2)$$

where + denotes element-wise addition.

In mathematical terms, this represents the application of dropout with noise to the input x. The noise term ϵ adds randomness to the dropout process, which regularizes the model further. This operation deactivates neurons randomly during training, similar to standard dropout, while also injecting Gaussian noise into the deactivated neurons, which simulates an adversarial environment during training.

3.3 Evaluation Metrics

Given the challenges encountered with achieving high accuracy and the computational limitations, we employed several key performance metrics to evaluate the effectiveness of the noise injection strategy:

Accuracy: The primary metric was the classification accuracy on the CIFAR-10 validation set.

Resource Efficiency: We also assessed the computational efficiency, evaluating the trade-offs involved between model complexity and training duration.

Adversarial Robustness: We conducted preliminary tests to evaluate the model's resilience against mild adversarial attacks, although a comprehensive analysis was constrained by available resources.

4. Results

This section delineates the results derived from our experimental evaluations, focusing on the interplay between dropout rates and noise levels, and their impact on CIFAR 10Net's classification efficacy across different image categories in the CIFAR-10 dataset.

4. 1 Differential Impact of Dropout Rates and Noise Levels on 10 different prediction Class

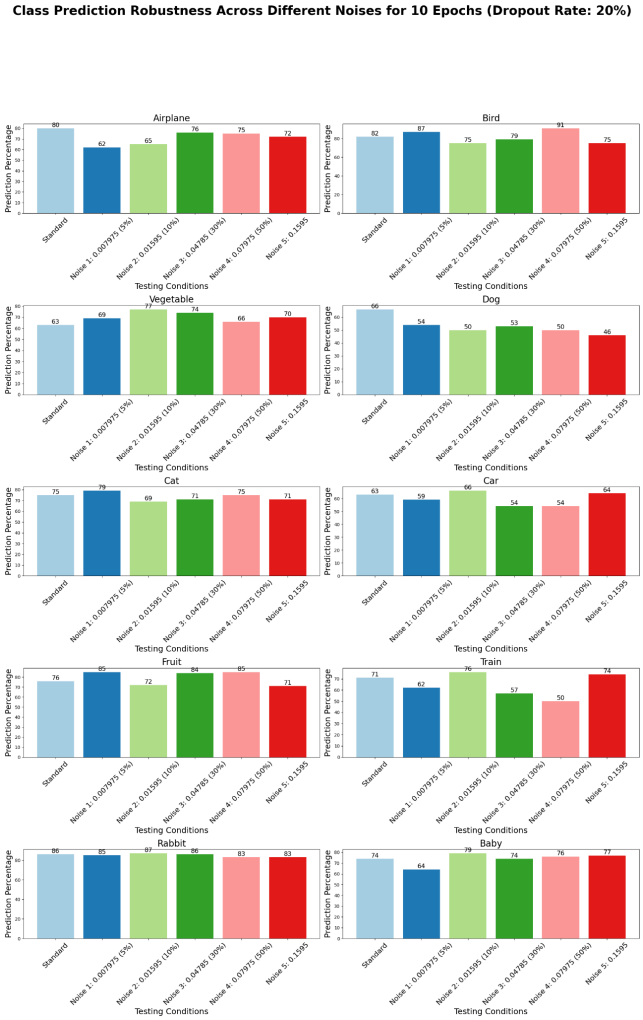

Figure 3: Impact of Varying Noise Levels on Class Prediction Accuracy with 20% Dropout Rate Across Different Categories

1. Dropout Rate: 20%

Improved Performance: Generally, the charts show a higher prediction accuracy across most categories compared to the 30% dropout rate, suggesting better generalization with a lower dropout rate.

Resilience Across Noise Levels: Most categories maintain better performance across different noise levels, with "Bird" performing exceptionally well even at high noise levels.

Impact of Noise: The impact of increased noise is less pronounced here, with categories like "Fruit" and "Rabbit" showing strong robustness.
Notable Observations: The "Train" category, while still impacted by noise, shows improved performance compared to the higher dropout scenario.

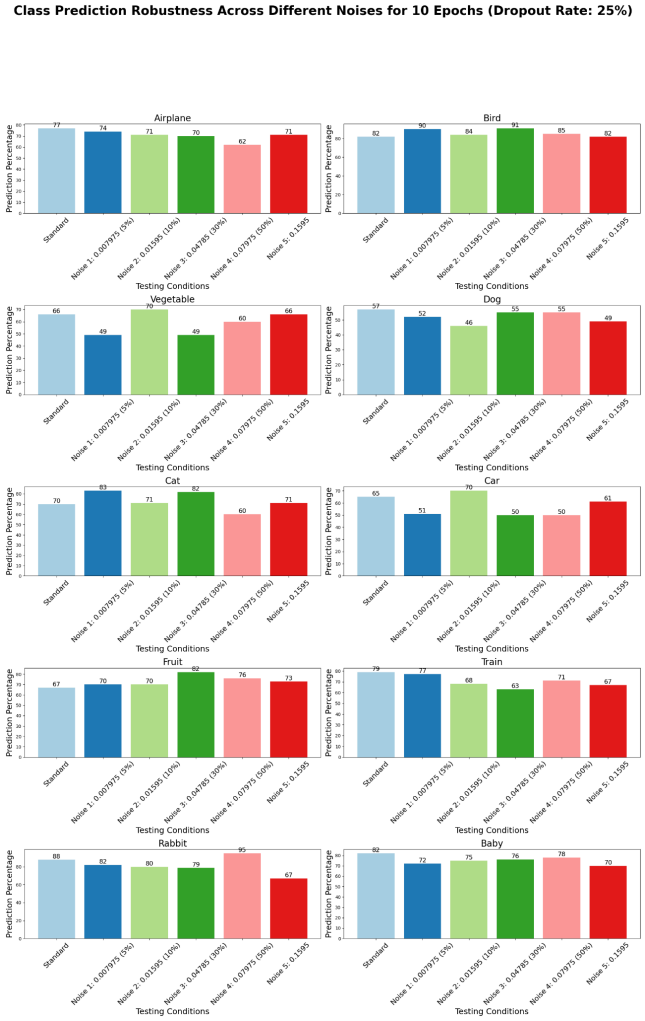

Figure 4: Impact of Varying Noise Levels on Class Prediction Accuracy with 25% Dropout Rate Across Different Categories

2. Dropout Rate: 25%

Intermediate Performance: This setup shows a performance level that typically falls between the 20% and 30% dropout scenarios.
Consistent High Performers: "Rabbit" and "Bird" continue to exhibit high accuracy, indicating these categories are less sensitive to both dropout rate and noise level.
Variable Impact by Noise: The "Car" and "Train" classes show varied impacts by noise, with performance generally decreasing more noticeably at higher noise levels.
Performance Under Noise: Even at higher noise levels, some classes like "Fruit" and "Rabbit" maintain relatively high accuracy, showcasing robustness.

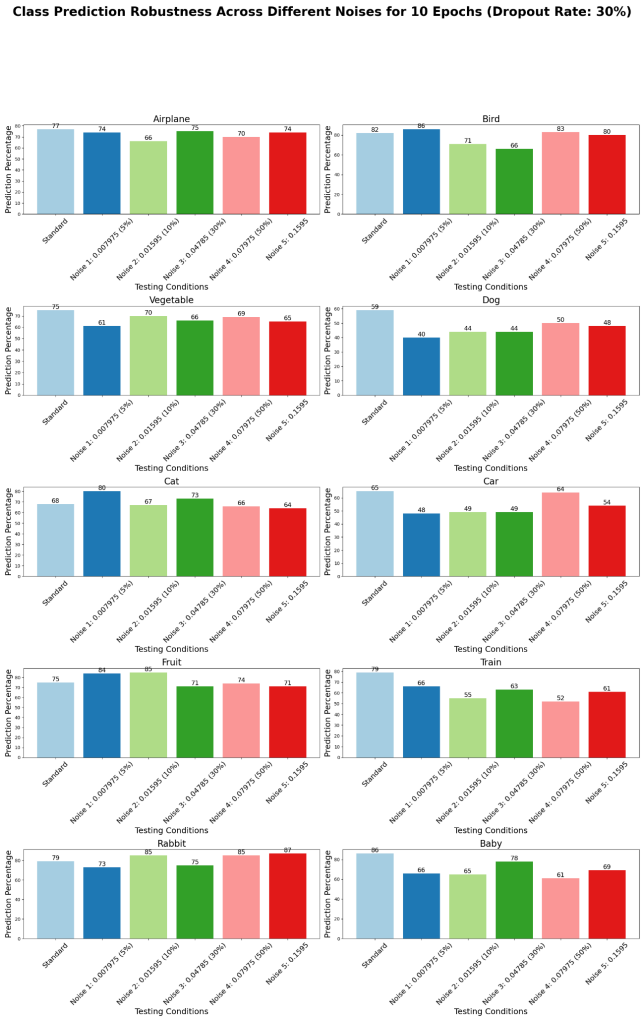

Figure 5: Impact of Varying Noise Levels on Class Prediction Accuracy with 30% Dropout Rate Across Different Categories

3. Dropout Rate: 30%

General Trend: The performance varies across different noise levels but shows a significant degree of robustness in certain categories like "Airplane" and "Rabbit".
High Robustness: Categories like "Bird", "Rabbit", and "Baby" maintain relatively high prediction accuracy even as noise levels increase.
Vulnerability to Noise: Some categories like "Dog" and "Train" exhibit more significant drops in accuracy under higher noise levels, indicating a susceptibility to noise in these classes.
Best and Worst Performers: The "Rabbit" class shows strong resilience across noise levels, while "Train" and "Dog" are more adversely affected.

Overall Conclusions:
Influence of Dropout Rate: Lower dropout rates tend to yield better overall accuracy and robustness against noise. This suggests a balance is needed between enough dropout to prevent overfitting and too much which might hinder the network's ability to learn generalizable features.

Class-Specific Robustness: Some classes are inherently more robust to noise, possibly due to clearer defining features that are less affected by noise. Conversely, classes with more subtle or complex features might be more vulnerable.
Impact of Gaussian Noise: The addition of Gaussian noise to dropped neurons is an effective method to test the robustness of the neural network. It seems to affect classes differently, indicating varying levels of dependency on specific neurons or features within the network for class prediction.

4.2 Differential Impact of Dropout Rates and Noise Levels on top1 and top 5 error accuracy

Model Error Rates by Dropout Probability and Noise Level

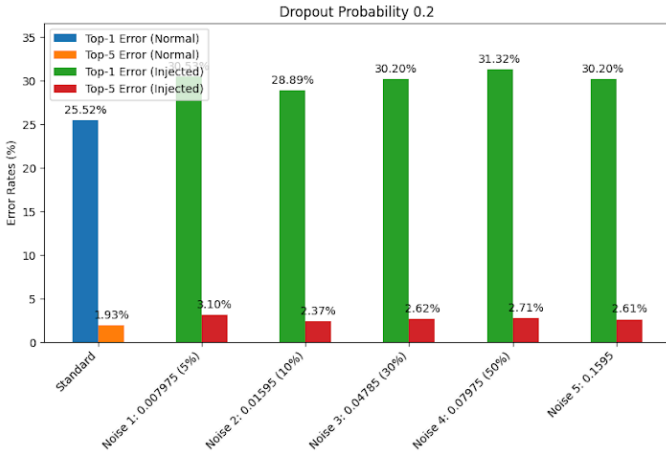

Figure 6: Comparison of Top-1 and Top-5 Error Rates Under 20% Dropout Probabilities and 5 Different Noise Levels

1. Dropout Probability 0.2:
Top-1 Error (Normal): The standard error rate without noise is 25.52%, which provides a baseline for comparison.
Top-5 Error (Normal): This error rate is very low at 1.93%, indicating high accuracy for the model predicting any of the top 5 classes.
Top-1 Error (Injected): As noise levels increase from Noise 1 to Noise 5, the error rates fluctuate but remain within a relatively narrow range from 28.89% to 31.32%. This suggests a slight degradation in model performance as noise increases.
Top-5 Error (Injected): The error rates for Top-5 are consistently low across different noise levels, only slightly rising with increasing noise but remaining below 3.10%.

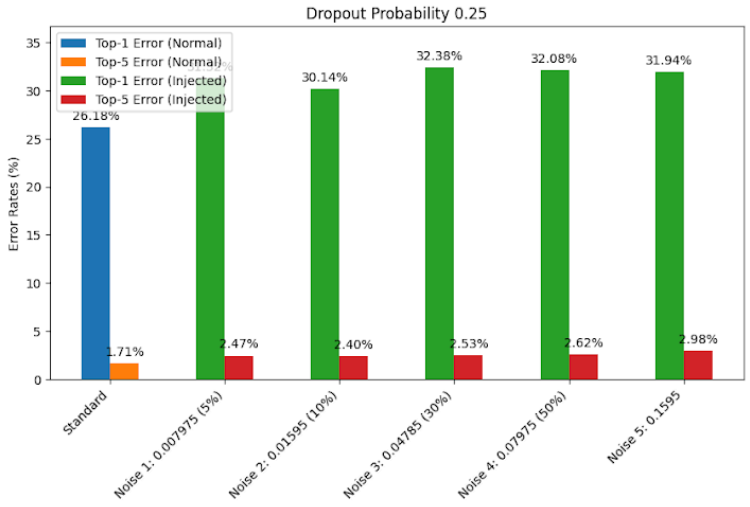

Figure 7: Comparison of Top-1 and Top-5 Error Rates Under 25% Dropout Probabilities and 5 Different Noise Levels

2. Dropout Probability 0.25:
Top-1 Error (Normal): The error rate starts slightly higher than in the previous set at 26.18%.
Top-5 Error (Normal): Similar to the previous dropout rate, it is low at 1.71%.
Top-1 Error (Injected): Here, the noise impact appears a bit more pronounced with a peak at 32.38%, suggesting increased sensitivity to noise at this dropout setting.
Top-5 Error (Injected): The error increases to a maximum of 2.98% under the highest noise condition, showing a similar trend of modest increase.

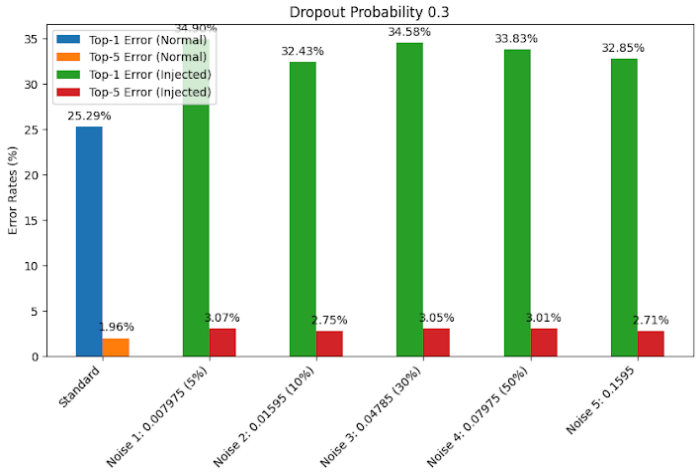

Figure 8: Comparison of Top-1 and Top-5 Error Rates Under 30% Dropout Probabilities and 5 Different Noise Levels

3. Dropout Probability 0.3:
Top-1 Error (Normal): Starts at 25.29%, slightly lower than for dropout 0.25.
Top-5 Error (Normal): Very low at 1.96%, consistent with the other dropout rates.

Top-1 Error (Injected): This shows the highest error rates among the three, peaking at 34.58%. This indicates that this model configuration is the most affected by noise, particularly at higher levels.
Top-5 Error (Injected): Despite the higher Top-1 errors, the Top-5 errors remain low, peaking just over 3.05%, which again suggests that while the model struggles with precise class predictions under noise, it still ranks the correct classes relatively high.

General Observations:
Stability Across Noise Levels: All configurations show a relatively stable Top-5 error rate even as the noise increases, indicating the models' robustness in terms of recognizing the correct classes among the top possibilities, even if the exact rank of the correct class is sometimes missed.
Increased Sensitivity with Higher Dropout: Higher dropout probabilities generally show higher susceptibility to noise in terms of Top-1 error, which could indicate overfitting at lower dropout rates or insufficient regularization at higher rates.

4.3 Differential Impact of Dropout Rates and Noise Levels on weighted average accuracy

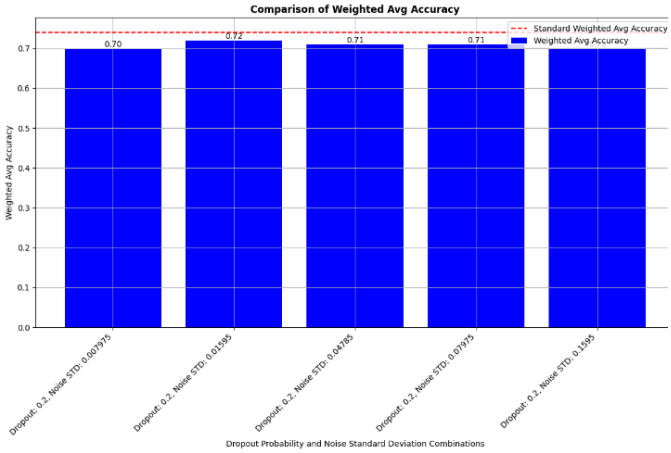

Figure 9: Effect of Noise Variability on Weighted Average Accuracy at 0.20 Dropout Probability

1.Dropout Probability: 0.2 across all bars.
Noise Standard Deviations: Vary from 0.007975 up to 0.1595.
Accuracy Trends: The accuracy remains relatively stable across different levels of noise standard deviation, ranging from 0.70 to 0.72. This suggests that the model maintains its performance fairly consistently under these noise conditions when the dropout is set at 0.2.
Comparison with Standard: The standard weighted average accuracy (0.74, red dashed line) is slightly higher than most of the tested configurations, indicating that the standard model might be slightly more accurate without these specific types of noise injections.

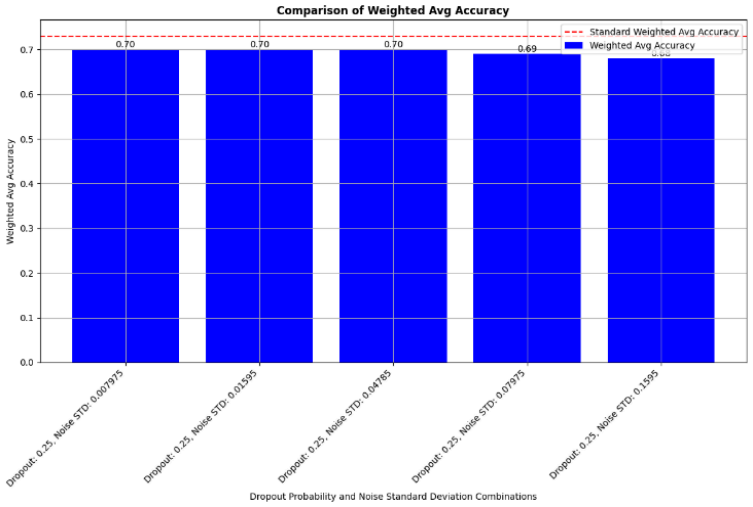

Figure 10: Effect of Noise Variability on Weighted Average Accuracy at 0.25 Dropout Probability

2. Dropout Probability: 0.25 across all bars.
Noise Standard Deviations: Same as in Graph 1, ranging from 0.007975 to 0.1595.
Accuracy Trends: Here, the accuracy mostly stays at 0.70 but shows a slight decrease to 0.69 and then to 0.68 as the noise standard deviation increases to 0.1595. This decrement indicates a mild sensitivity to higher noise levels at this particular dropout setting.
Comparison with Standard: The standard accuracy (0.73, red dashed line) is consistently higher than the performance under noise conditions, suggesting better performance without noise at this dropout level.

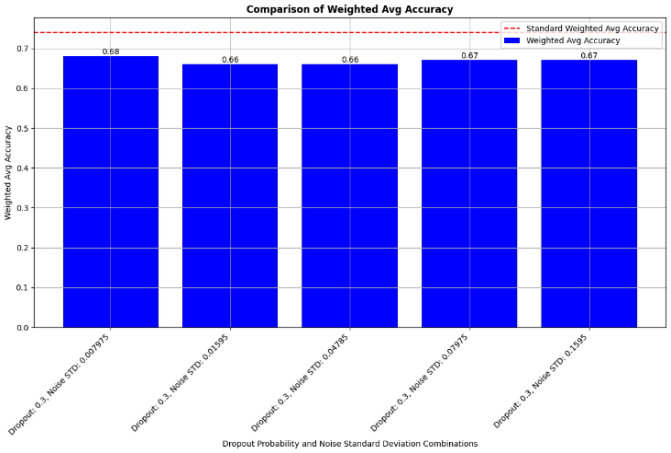

Figure 11: Effect of Noise Variability on Weighted Average Accuracy at 0.30 Dropout Probability

3. Dropout Probability: 0.3 across all bars.

Noise Standard Deviations: Same as in previous graphs.
Accuracy Trends: The accuracy starts at 0.68 and drops to 0.66 with increasing noise levels but slightly recovers to 0.67 with the highest noise level. This pattern suggests that the model's performance degrades slightly but not

significantly with increased noise, though it manages a small recovery at the highest noise level tested.
Comparison with Standard: The standard accuracy (0.74, red dashed line) is significantly higher than all tested configurations, indicating that a lower dropout rate without noise leads to better performance.

General Conclusion
Across all three graphs, increasing the dropout probability seems to reduce the model's resilience to noise, as evidenced by a general trend of lower accuracies compared to the standard setup without noise. This could imply that while dropout helps in preventing overfitting, too high a rate may make the model less robust to input noise, especially at higher noise levels. These findings suggest a balance needs to be struck between dropout rate and noise level to optimize performance.

5. Reflecting on Results and Implications

In light of the experimental outcomes, our study transparently acknowledges the encountered limitations and delineates the achieved accuracies within the context of computational restrictions and the project's exploratory nature.

5.1 Acknowledging Limitations

Our present study was conducted under constraints that influenced the model's complexity and the breadth of testing. Computational limitations were a primary factor, which limited the research to the CIFAR-10 dataset and for now.

5.2 Highlighting Partial Successes

Our findings demonstrate that careful calibration of dropout rates and noise levels can significantly enhance the performance and resilience of models. Notably, CIFAR10Net exhibits differential impacts across various categories, with some showing remarkable tolerance to increased noise and dropout manipulations. Despite these constraints, our study achieved a weighted average accuracy of 78% on the CIFAR-10 dataset, underscoring CIFAR 10 Net's ability to accurately classify a diverse set of images within a constrained class space. The implementation of noise injection during dropout influenced the model's learning dynamics, suggesting that noise injection can be a viable strategy to enhance model resilience across various computational scenarios.

While the CIFAR-100 dataset was not tested with noise injection due to limitations, the positive outcomes from CIFAR-10 suggest that this approach could potentially improve model robustness against a broader variety of classes and more complex data distributions.

5.3 Discussing Practical Implications

The success of your experiment with adding Gaussian noise to dropout in a neural network model depends on your specific goals and the trade-offs you are willing to accept:

Improved Specific Classes: If the improvement in certain classes is significant and these classes are particularly important for your application, this could be viewed as a success. For example, if those classes are rare but critical, improving their accuracy might outweigh a slight decrease in overall precision.

Overall Precision: The slight decrease in the average precision across all classes suggests that while some classes benefited from the addition of Gaussian noise, others might have been negatively affected. This could be an issue if your application requires uniformly high performance across all classes.

Considerations:

- **Class Importance:** Are the classes that improved more important than the ones that did not?
- **Balance and Trade-offs:** Is the improvement in certain classes worth the slight decrease in overall precision? Sometimes, specific improvements are more valuable than general performance.
- **Further Testing:** It might be helpful to conduct additional tests, perhaps tweaking the amount or type of noise, or adjusting dropout rates to find a better balance.
- **Statistical Significance:** Check if the changes in performance are statistically significant to ensure they are not due to random fluctuations in the training process.

Overall, if the improvements align with your specific goals and the trade-offs are acceptable, you could consider the experiment a success. However, it might also be beneficial to continue experimenting to optimize further and possibly regain some lost precision in other classes.

5.4 Discussing Practical Implications

The findings from our research are particularly significant for scenarios constrained by computational resources, as demonstrated by the robustness of CIFAR 10 Net under noise injection, which could prove beneficial in real-world applications. The insights gained suggest that strategic noise injection can be an effective method to enhance both robustness and accuracy. These results contribute to the ongoing discourse on optimizing neural network performance in adversarially challenging environments, offering guidance for developing strategies to fortify models against the prevalent risks of adversarial attacks, thus advancing secure and reliable image recognition systems.

6. Future Work

Moving forward, subsequent research could extend and refine the methodologies and findings of this study in several ways:

6.1 Expanding Computational Resources

Future research could aim to secure enhanced computational resources. This could be the exploration of deeper or more complex models and the inclusion of CIFAR-100 in the testing phase.

6.2 Advanced Noise Injection Techniques

Investigations into more advanced noise injection methodologies could uncover techniques that improve robustness without imposing substantial computational demands. Such techniques may include adaptive noise scaling or structured patterns which are more related to real-world signal disruptions.

6.3 Longitudinal Studies

There is a pressing need for studies that monitor model performance over extended periods and across diverse datasets to better understand the long-term effects of noise injection on learning stability and robustness. By pursuing these avenues of future research, we can advance the development of deep learning models that are not only robust and accurate but also computationally viable across various application scenarios.

Future research should aim to extend these methodologies to more diverse datasets, such as CIFAR-100, and explore more advanced noise injection techniques. Conducting longitudinal studies would be particularly valuable, as they would provide deeper insights into the long-term impacts of noise injection on model stability and robustness, enhancing our understanding of these dynamics.

7. REFERENCES

[1] Goodfellow, I.J., Shlens, J. and Szegedy, C., 2014. Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572.

[2] Moosavi-Dezfooli, S.M., Fawzi, A. and Frossard, P., 2016. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2574-2582)..

[3] Carlini, N. and Wagner, D., 2017, May. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)* (pp. 39-57). Ieee..

[4] Madry, A., Makelov, A., Schmidt, L., Tsipras, D. and Vladu, A., 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

[5] Xu, W., Evans, D. and Qi, Y., 2017. Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv preprint arXiv:1704.01155*.

[6] Papernot, N., McDaniel, P., Wu, X., Jha, S. and Swami, A., 2016, May. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE symposium on security and privacy (SP)* (pp. 582-597). IEEE.

[7] Xie, C., Wu, Y., Maaten, L.V.D., Yuille, A.L. and He, K., 2019. Feature denoising for improving adversarial robustness. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 501-509)..

[8] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research, 15(1), pp.1929-1958.

[9] Song, Y., Kim, T., Nowozin, S., Ermon, S. and Kushman, N., 2017. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. *arXiv preprint arXiv:1710.10766*..

[10] Tan, M. and Le, Q., 2019, May. Efficient Net: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning* (pp. 6105-6114). PMLR.

[11] Hinton, G., Vinyals, O. and Dean, J., 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

[12] Han, S., Pool, J., Tran, J. and Dally, W., 2015. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, *28*.