# Impact of Training Data Quality on YOLOv8n Performance for Crack Detection in Concrete Structures

By:
Weisberg Ran
Tumanov Artemiy

# Contents

**Abstract**

This study evaluates the performance of the YOLOv8n model in detecting cracks in concrete structures, focusing on the impact of training data quality. We compared models trained on clean datasets to those trained on datasets augmented with noise and blur. The evaluation involved testing these models on various image quality conditions, including clean, noisy, and blurred images. Our results demonstrated that the model trained on degraded data exhibited higher robustness and generalization capabilities, achieving consistently higher recall and precision values across all test sets. These findings highlight the importance of incorporating realistic data variations during training to enhance the model's ability to handle real-world scenarios. This research provides valuable insights for improving the reliability and effectiveness of machine learning applications in structural health monitoring.

# 1 Introduction

Machine learning has brought significant advancements to various fields, including structural health monitoring. The ability to automatically detect and analyze cracks in concrete structures is crucial for maintaining safety and extending the lifespan of these structures. Traditional manual inspection methods are being increasingly augmented or replaced by automated, machine learning-based approaches due to their efficiency and accuracy. YOLOv8n (You Only Look Once version 8 nano) stands out as a state-of-the-art object detection algorithm capable of high-speed and accurate performance. This project investigates the application of YOLOv8n for detecting cracks in concrete surfaces, leveraging the capabilities of deep learning to address this critical task.

## 1.1 Motivation

The motivation for this project stems from the need to enhance the efficiency and reliability of crack detection in concrete structures. Real-world applications often involve images with varying resolutions and noise levels, which can significantly impact the performance of automated detection systems. By exploring how YOLOv8n performs under different image quality conditions, we aim to understand its practical limits and optimize its deployment in real-world scenarios. This investigation is driven by the necessity to ensure that machine learning models remain robust and effective, even when dealing with suboptimal input data.

## 1.2 The Goal Of The Project

The primary goals of this study are to:

1. Assess the performance of YOLOv8n in detecting cracks in concrete structures under varying image quality conditions.
2. Compare the effectiveness of the model when trained on a clean dataset versus a modified dataset with added noise and blur.
3. Determine the impact of training data quality on the model's accuracy and reliability.
4. Provide insights into the practical deployment of YOLOv8n for structural health monitoring in diverse real-world scenarios.

Through these objectives, we aim to highlight the strengths and limitations of using YOLOv8n for crack detection in concrete, ensuring robust and effective performance even with suboptimal input data. In the following sections, we will delve into the methods and computational models employed in this study, describe the problem in detail, present our findings, and discuss their implications. Finally, we will conclude with a summary of our contributions and their significance to the field of structural health monitoring.

## 1.3 Literature Review

**YOLOv8n** YOLO8 (You Only Look Once version 8) Jocher et al. (2023) is a state-of-the-art object detection algorithm that builds on the success of its predecessors. YOLOv8n (nano version) is designed for high-speed, real-time detection with high accuracy. Unlike traditional object detection methods that use a sliding window approach, YOLOv8 predicts bounding boxes and class probabilities directly from full images in a single evaluation, making it significantly faster.

YOLOv8n differs from earlier versions by incorporating improvements that enhance its performance and efficiency. The model features a backbone network with 29 convolutional layers,

optimized for feature extraction. It has approximately 3 million parameters (weights), making it lightweight yet powerful. The architecture includes:

1. **Backbone:** 29 convolutional layers that extract feature maps from input images.
2. **Neck:** Combines features at different scales using additional convolutional layers.
3. **Head:** Outputs the final predictions, including bounding boxes and class probabilities.

These enhancements make YOLOv8n a powerful tool for real-time object detection. It is widely used in various fields, including autonomous driving, surveillance, and medical imaging, due to its efficiency and effectiveness.

**Recall** Recall, also known as sensitivity or true positive rate, is a critical metric in the evaluation of machine learning models, particularly in the context of object detection tasks like crack detection in concrete structures. Recall measures the ability of a model to correctly identify all relevant instances within a dataset. It is defined as:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{1.1}$$

where TP (True Positives) represents the number of correctly predicted positive instances, and FN (False Negatives) represents the number of actual positive instances that were not identified by the model. High recall indicates that the model successfully captures most of the actual positive instances, making it particularly important in scenarios where missing a true positive (such as an undetected crack) could have severe consequences. In structural health monitoring, achieving high recall ensures that most cracks are detected, thereby reducing the risk of structural failures.

**Precision** Precision, also known as positive predictive value, is another vital metric for evaluating machine learning models, particularly in object detection. Precision measures the accuracy of the positive predictions made by the model. It is defined as:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{1.2}$$

where TP (True Positives) represents the number of correctly predicted positive instances, and FP (False Positives) represents the number of instances incorrectly predicted as positive. High precision indicates that the model has a low rate of false positive predictions, which is crucial in reducing unnecessary interventions and costs. In the context of crack detection in concrete structures, high precision ensures that the identified cracks are actual cracks, minimizing the chances of false alarms and ensuring efficient allocation of maintenance resources.

**DawgSurfaceCracks** The DawgSurfaceCracks Image Dataset, available on the Roboflow platform, is a comprehensive dataset specifically designed for training models to detect cracks in concrete surfaces. This dataset includes high-resolution images of concrete with annotated cracks, providing a valuable resource for developing and testing machine learning models. The dataset's quality and diversity make it an excellent choice for training YOLOv5, ensuring that the model learns to identify cracks under various conditions and textures.

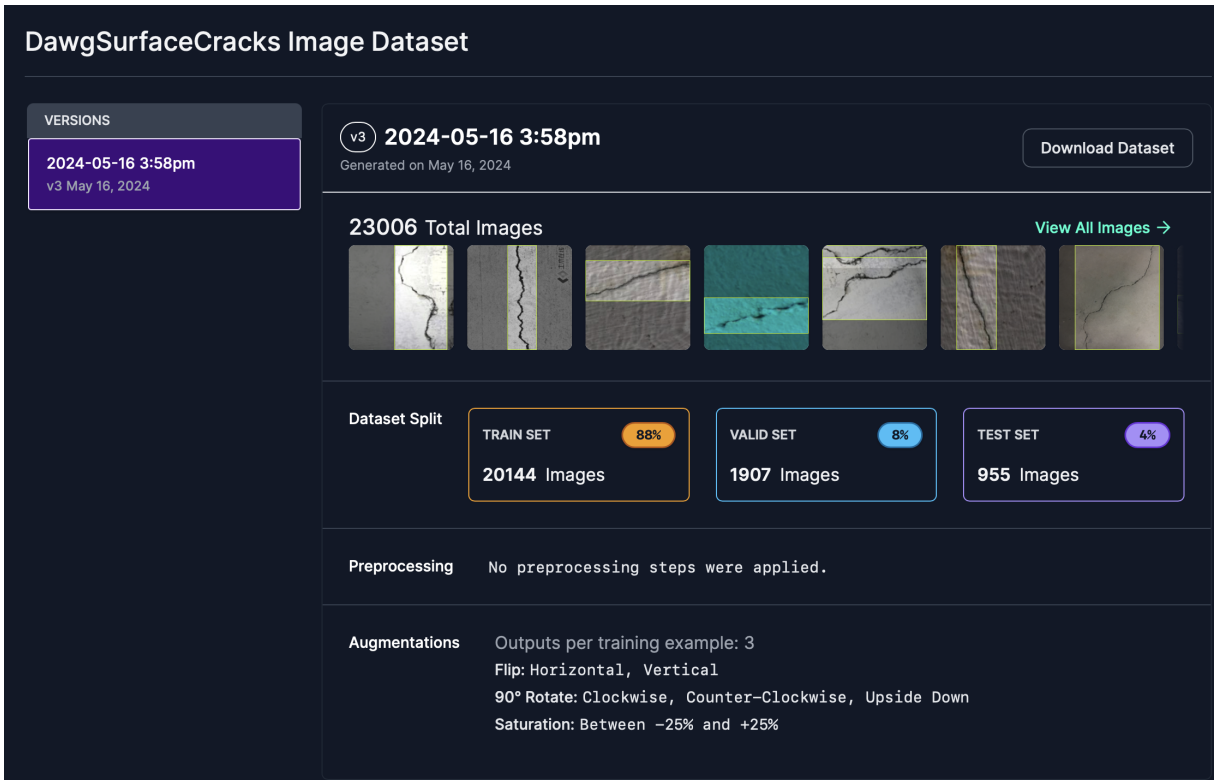


Figure 1.1: DawgSurfaceCracks Image Dataset page at Roboflow

## 2 Methods

To achieve the goals of this project, we divided the work into three main parts: training the model, creating a dataset for testing, and evaluating the model's performance. Each part is crucial for understanding the capabilities and limitations of YOLOv8n in detecting cracks in concrete under varying image quality conditions.

### 2.1 Training the Model

To train the model, we first used the DawgSurfaceCracks Image Dataset, which provides a comprehensive collection of annotated images specifically designed for this task. The dataset consists of 20,144 training images and 1,907 validation images.

We trained the models for 50 epochs with a batch size of 16. The initial training process used the YOLOv8n pre-trained weights as a starting point, allowing the model to refine its ability to detect cracks in concrete.

Next, we trained a second model using a modified version of the training dataset. To create this training set, we wrote a script that randomly applied a predetermined amount of blur to one-third of the data and noise to another third, while the remaining third of the data remained unchanged. This new training set aimed to simulate real-world conditions with varying image quality.

We then used this modified training set to train the second model under the same conditions, starting with the YOLOv8n pre-trained weights. This approach allowed us to compare the performance of the model trained on clean data with the model trained on partially degraded data, providing insights into the impact of training data quality on the model's accuracy and reliability.

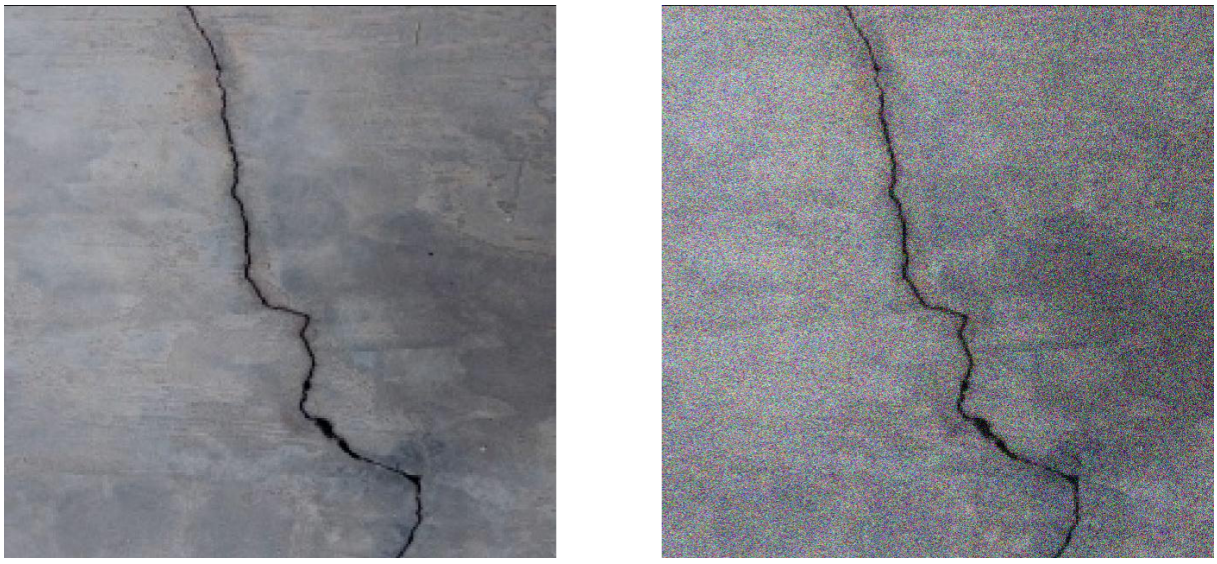

Figure 2.1: Comparison of training images: (Left) clean image, (Right) image with added noise.

### 2.2 Creating the Test Sets

We found 100 images of various cracks online and used Roboflow to add annotations to the set. To test the performance of the models on images with degraded quality, we created a script that can add noise or blur to a given set and generate as many new sets as required. The user specifies the percentage of noise or blur to add, which serves as a scale for the amount of distortion applied.

Using this script, we created 7 sets with varying levels of blur and another 7 sets with varying levels of noise. For both types of distortion, we generated sets with 2.5%, 5%, 10%, 15%, 20%,

25%, and 30% distortion. These test sets allowed us to systematically evaluate the robustness and accuracy of the YOLOv8n models under different levels of image degradation, providing a controlled environment to assess the practical limits of the models' performance.

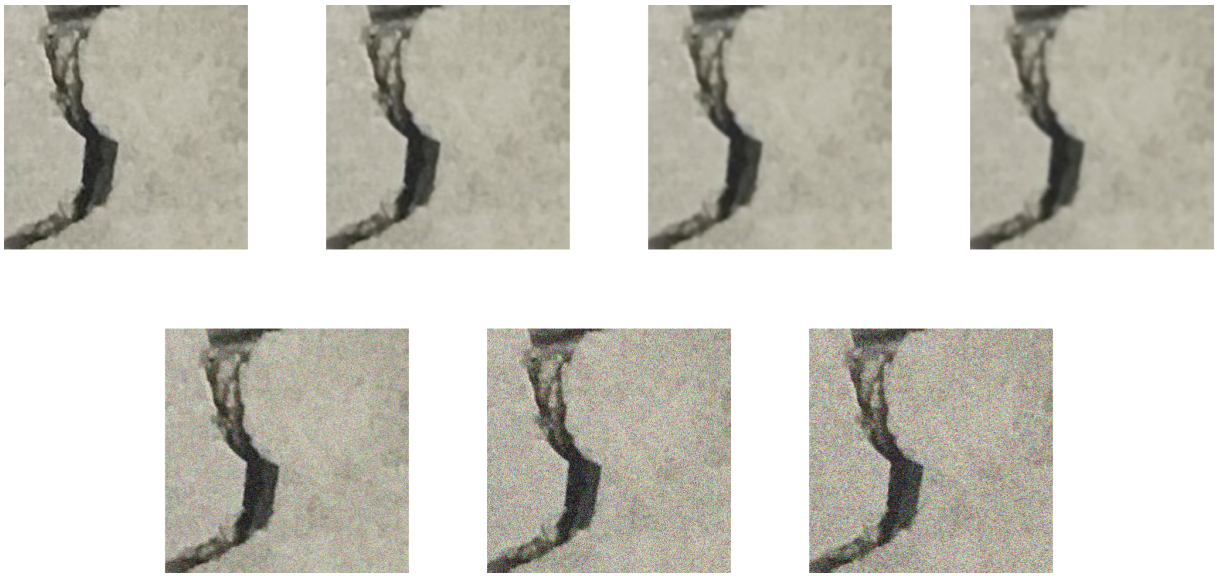

Figure 2.2: Example images from the test sets: (Top row, left to right) Clean image, 10% blur, 20% blur, 30% blur. (Bottom row, left to right) 10% noise, 20% noise, 30% noise.

## 2.3 Testing and evaluating the model

We used the built-in `val` function of YOLO to evaluate each trained model. For each model, we tested it on all 15 sets of data, which included varying levels of noise and blur.

For each test set, we recorded the recall and precision metrics to evaluate the performance of the models. Recall measures how many actual positive instances (i.e., true cracks) the model correctly identifies, while precision indicates how many of the identified positive instances are actually correct. These metrics provide a comprehensive evaluation of the model's accuracy and reliability in detecting cracks under different conditions.

By systematically running the evaluation on all the test sets, we assessed the robustness and effectiveness of the YOLOv8n models under varying image quality conditions.

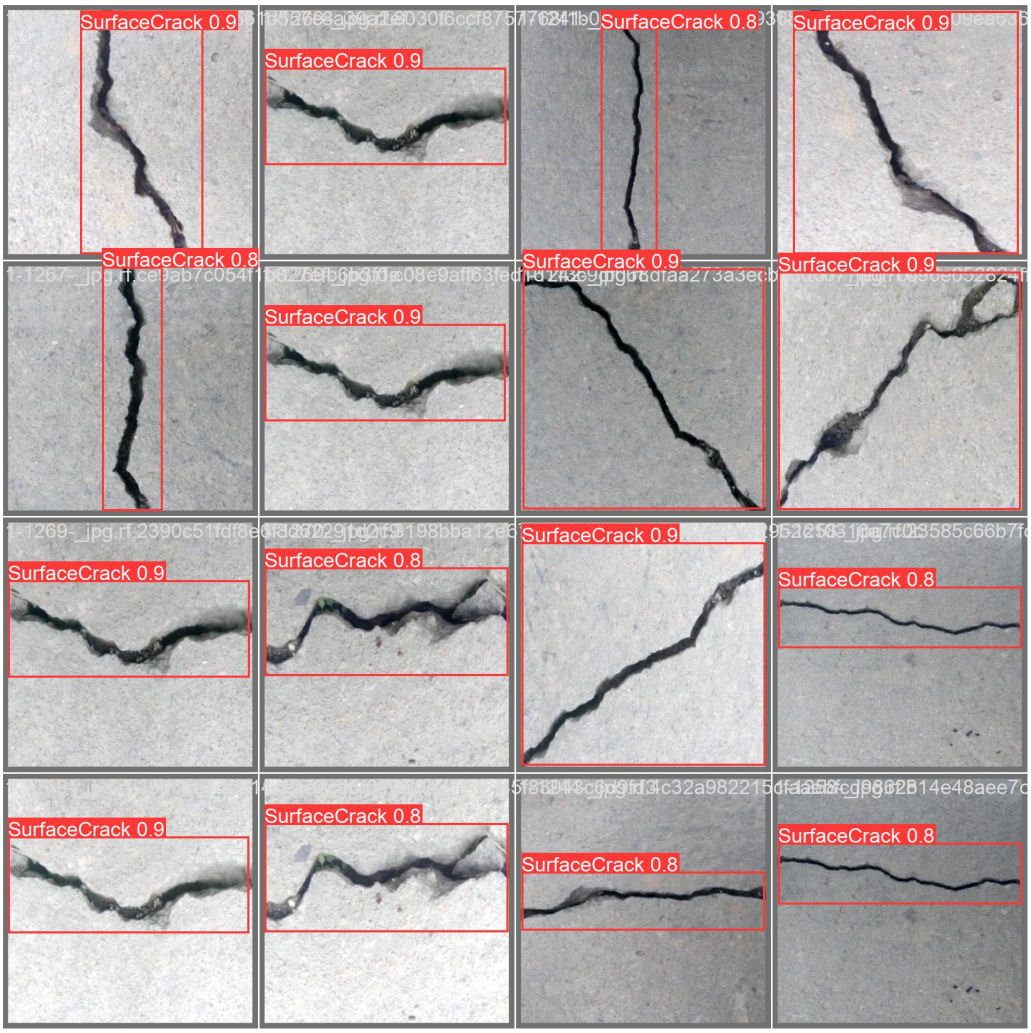


Figure 2.3: Example of YOLOv8n model detections on test images with varying levels of noise and blur.

## 3 Problem description

Many existing neural networks are designed to detect cracks in concrete and other building materials. However, these models often exhibit high sensitivity to noise and poor resolution in images. This sensitivity can result in two major issues: false positive identifications of cracks, leading to unnecessary expenditures and wasted labor, and false negative identifications, where actual cracks are not detected, potentially causing structural failures with severe consequences.

Given these challenges, it is crucial to investigate the performance limits of current models under varying image quality conditions. Our study aims to identify the thresholds for image resolution and levels of noise and blur beyond which the accuracy of these models significantly deteriorates. By training models on both clean and partially degraded datasets, and then testing them on images with varying degrees of noise and blur, we aim to define the operational boundaries within which existing neural networks can reliably function. This approach enhances their practical application in structural health monitoring by ensuring robust performance even under suboptimal conditions.

# 4 Results

In this chapter, we present the results of training process and our experiments. The performance of the YOLOv8n models was evaluated on a series of test sets with varying levels of noise and blur. We compare the models trained on clean data with those trained on partially degraded data to understand the impact of image quality on model accuracy and reliability. The key metrics analyzed include precision and recall.

## 4.1 Training Results

In this chapter, we present the training results for the YOLOv8n models on both clean and partially degraded datasets. The training process was monitored using various metrics, including box loss and precision-recall curves, to evaluate the performance and stability of the models during training.

The following figures summarize the training and validation box loss, as well as the precision and recall values, across all epochs.

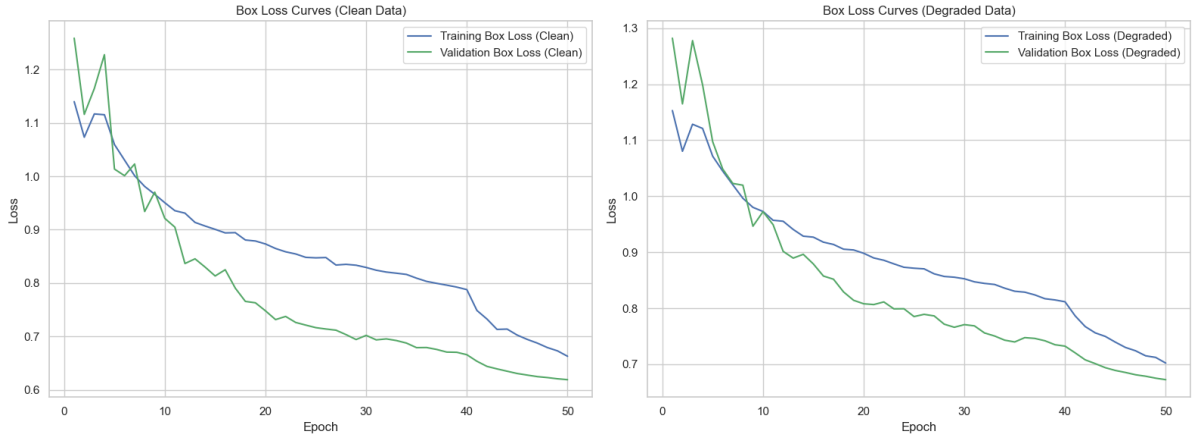


Figure 4.1: Box Loss Curves of YOLOv8n Models on Clean and Degraded Datasets

Figure 4.1 shows the box loss curves for both the clean and degraded datasets. The training and validation box loss are plotted over 50 epochs. The box loss measures the difference between the predicted bounding boxes and the ground truth bounding boxes.

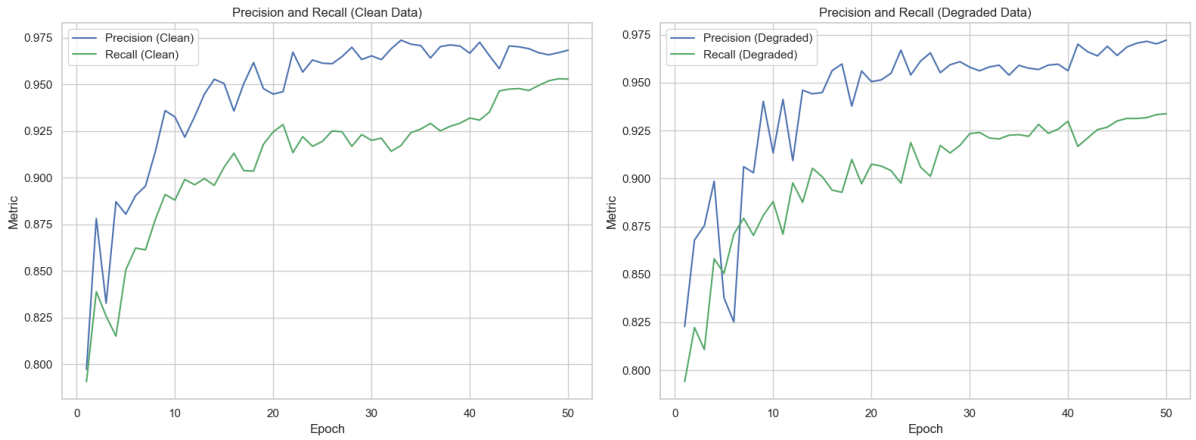


Figure 4.2: Precision and Recall Curves of YOLOv8n Models on Clean and Degraded Datasets

Figure 4.2 presents the precision and recall curves for both the clean and degraded datasets.

Precision is the ratio of true positive detections to the total number of positive detections made by the model, while recall is the ratio of true positive detections to the total number of actual positives in the dataset.

The box loss curves indicate the model's learning progression and stability during training, while the precision and recall curves illustrate the model's accuracy in detecting and classifying cracks. By comparing these metrics for the clean and degraded datasets, we can assess the impact of data quality on the model's performance and generalization capability.

## 4.2 Test Results

In this chapter, we present the performance results of the YOLOv8n models trained on both clean and partially degraded datasets. The models were evaluated on a series of test sets with varying levels of noise and blur. The key metrics analyzed include recall and precision, which provide insights into the model's ability to accurately detect cracks in concrete under different conditions.

The following table summarizes the recall and precision values for both the clean and degraded training sets across all test conditions. Each test set represents a different level of image quality degradation, allowing us to assess the robustness and accuracy of the models in various real-world scenarios.

| **Test Set** | **Recall (Clean)** | **Recall (Degraded)** | **Precision (Clean)** | **Precision (Degraded)** |
|---|---|---|---|---|
| clean data | 0.6883 | 0.7456 | 0.7903 | 0.8863 |
| data with 2.5% noise | 0.7611 | 0.7826 | 0.8537 | 0.8503 |
| data with 5% noise | 0.7304 | 0.7347 | 0.8204 | 0.8756 |
| data with 10% noise | 0.7739 | 0.7360 | 0.7938 | 0.8758 |
| data with 15% noise | 0.7391 | 0.7043 | 0.7392 | 0.8666 |
| data with 20% noise | 0.7383 | 0.7174 | 0.7081 | 0.8099 |
| data with 25% noise | 0.7391 | 0.7174 | 0.7287 | 0.8099 |
| data with 30% noise | 0.7130 | 0.7087 | 0.7220 | 0.7969 |
| data with 2.5% blur | 0.7391 | 0.7478 | 0.7512 | 0.8865 |
| data with 5% blur | 0.7174 | 0.7130 | 0.7904 | 0.8760 |
| data with 10% blur | 0.7130 | 0.7130 | 0.7263 | 0.8760 |
| data with 15% blur | 0.7043 | 0.7043 | 0.7359 | 0.8666 |
| data with 20% blur | 0.7174 | 0.7174 | 0.7845 | 0.8031 |
| data with 25% blur | 0.7087 | 0.7043 | 0.7309 | 0.7935 |
| data with 30% blur | 0.7174 | 0.7087 | 0.7927 | 0.7811 |

Table 4.1: Performance of YOLOv8n Models on Test Sets with Varying Levels of Noise and Blur

Following the table, we provide a detailed comparison of the recall metrics between the clean and degraded training sets. The graph below illustrates the recall performance of the models across the various test sets, highlighting the differences in their ability to correctly identify cracks in concrete.

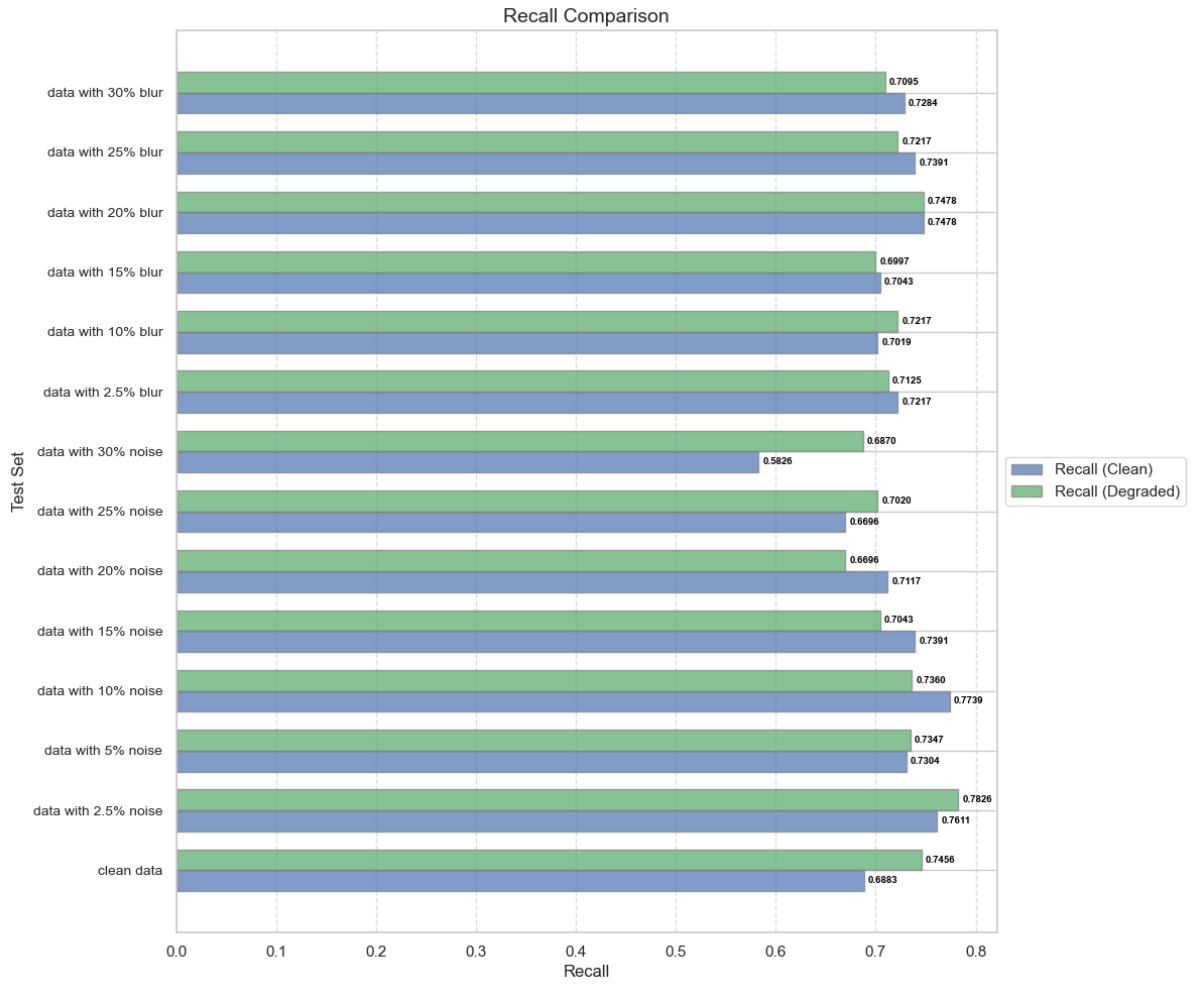


Figure 4.3: Recall Comparison of YOLOv8n Models on Test Sets with Varying Levels of Noise and Blur

Next, we compare the precision metrics between the clean and degraded training sets. The following graph presents the precision performance, indicating the accuracy of the models' predictions in identifying true cracks without misclassifying other features as cracks.

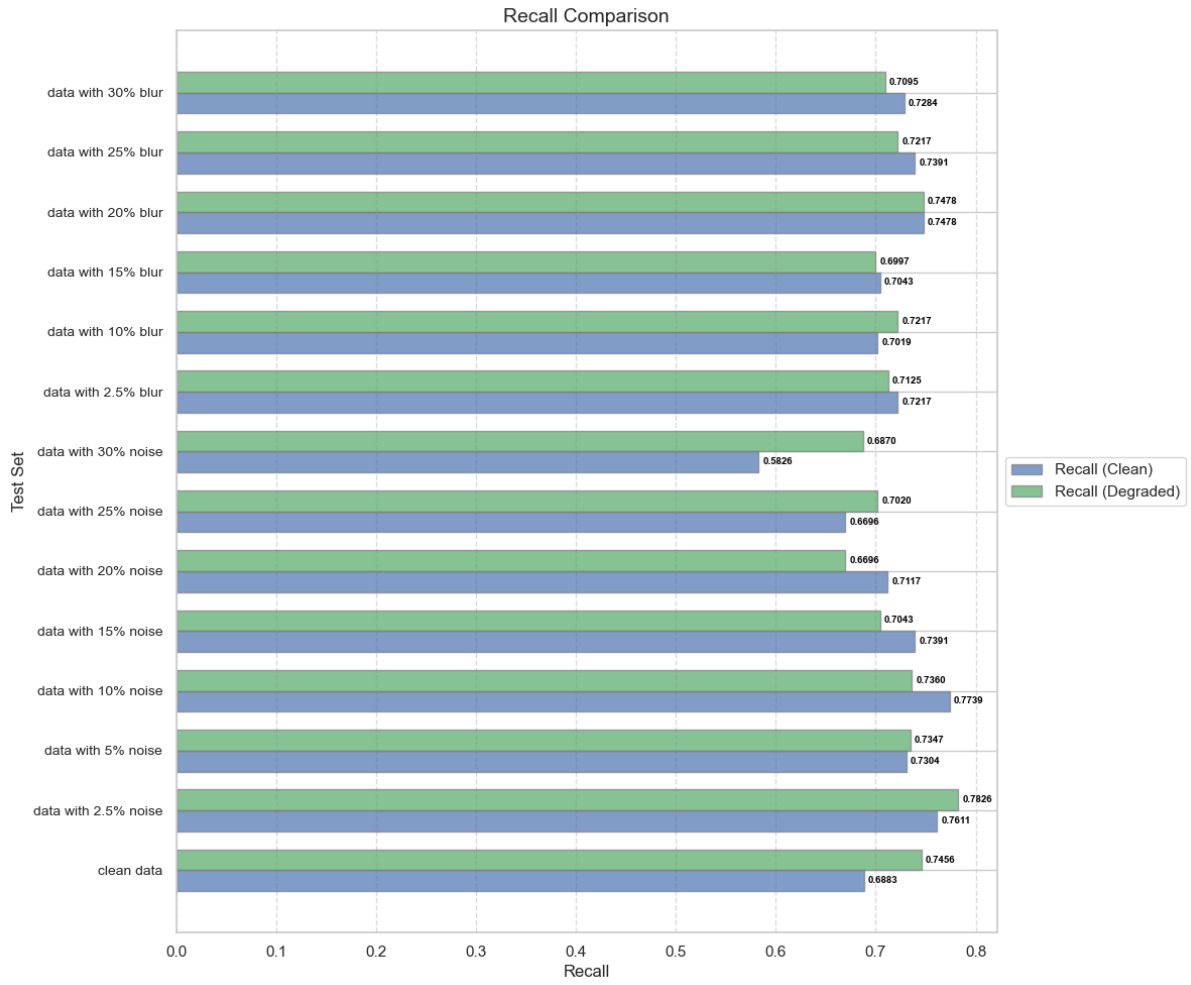


Figure 4.4: Precision Comparison of YOLOv8n Models on Test Sets with Varying Levels of Noise and Blur

# 5 Discussion

In this chapter, we interpret and discuss the results presented in the previous chapter. We compare the performance of the YOLOv8n models trained on clean and degraded datasets and also discuss the training process.

## 5.1 Training Process Analysis

During the training process, we were limited to 50 epochs for each model due to constraints in computational resources. The box loss plots from the previous chapter (Figure 4.1) indicate that, in both cases, the models had not yet begun to converge. This suggests that extending the training to more epochs could potentially improve the models' performance.

The box loss plots reveal that the model trained on clean data achieved a lower loss for both the training and validation datasets compared to the model trained on degraded data. This indicates better learning and generalization for the model trained on clean data. Additionally, after approximately 10 epochs, the training loss for the clean data model becomes higher than the validation loss and remains so throughout the training process. This pattern suggests that the model is not overfitting, as overfitting would typically result in higher validation loss compared to training loss.

In terms of recall and precision, as shown in Figure 4.2, there is no significant difference in precision between the models. However, the model trained on the degraded data shows an improvement in recall. This indicates that, despite the challenges presented by degraded data, the model can still improve its ability to correctly identify positive instances, even if the precision does not significantly differ from the clean data model.

Overall, the loss plots suggest that there is potential for further training to achieve convergence, and the recall and precision curves provide valuable insights into the models' performance under different training conditions.

## 5.2 Test Results Analysis

The performance of the YOLOv8n models on various test sets with different levels of noise and blur is compared in terms of recall and precision. Figures 4.3 and 4.4 present the recall and precision comparisons, respectively.

### 5.2.1 Recall Comparison

Figure 4.3 illustrates the recall performance of the models across the various test sets, highlighting the differences in their ability to correctly identify cracks in concrete.

- **General Performance**: The model trained on clean data shows a lower recall on the clean test set compared to the model trained on degraded data. Specifically, the recall for the clean model is around 0.6883, while for the degraded model it is about 0.7456. This suggests that the model trained on degraded data is more robust in detecting cracks, even on clean data.
- **Impact of Noise**: For test sets with varying levels of noise, the recall for the degraded model is generally higher than the clean model. For example, with 2.5% noise, the recall for the clean model is 0.7611, while for the degraded model it is 0.7826. As the noise level increases, the recall for both models decreases, but the model trained on degraded data consistently outperforms the clean model.
- **Impact of Blur**: Similarly, for test sets with varying levels of blur, the recall for the degraded model is generally higher. For instance, with 2.5% blur, the recall for the clean

model is 0.7391, while for the degraded model it is 0.7478. The recall decreases as the blur level increases, but again, the degraded model maintains a higher recall across all levels.

### 5.2.2 Precision Comparison

Figure 4.4 presents the precision performance, indicating the accuracy of the models' predictions in identifying true cracks without misclassifying other features as cracks.

- **General Performance**: The precision for the clean model on the clean test set is 0.7903, while for the degraded model, it is higher at 0.8863. This suggests that the model trained on degraded data is also more accurate in identifying true positives without many false positives.
- **Impact of Noise**: The precision for the test sets with noise shows that the clean model has slightly lower precision compared to the degraded model. For instance, with 2.5% noise, the precision for the clean model is 0.8537, while for the degraded model, it is 0.8503. As the noise level increases, the precision for both models fluctuates but remains higher for the degraded model.
- **Impact of Blur**: For test sets with blur, the precision for the clean model is lower compared to the degraded model. For example, with 2.5% blur, the precision for the clean model is 0.7512, while for the degraded model, it is 0.8865. The precision decreases as the blur level increases, but the degraded model generally maintains a higher precision.

### 5.2.3 Key Insights

- **Robustness**: The model trained on degraded data (with added noise and blur) demonstrates higher robustness and performs better across all test sets, including clean, noisy, and blurred data. This is evident from the consistently higher recall and precision values.
- **Generalization**: Training on a dataset with variations (noise and blur) helps the model generalize better to unseen data with similar variations. This results in better overall performance.
- **Data Quality**: The higher recall and precision values for the model trained on degraded data suggest that incorporating noise and blur into the training process can improve the model's ability to handle real-world conditions where such imperfections are common.
- **Model Performance**: The performance difference between the clean and degraded models indicates that training with a more diverse dataset can lead to better model performance, even when tested on clean data.

## 6 Conclusions

In this study, we investigated the performance of the YOLOv8n model in detecting cracks in concrete structures under various image quality conditions by comparing models trained on clean data and those trained on datasets augmented with noise and blur. We found that the model trained on degraded data demonstrated higher robustness and better generalization capabilities, with consistently higher recall and precision values across all test sets. This suggests that training with diverse and realistic data variations improves the model's ability to handle real-world conditions. These findings are relevant for practical applications in structural health monitoring, highlighting the importance of training data quality and variation in developing reliable machine learning models for detecting cracks in concrete.

## References

Jocher, G., Chaurasia, A., Qiu, J., and LeGrand, A. (2023). Yolov8: You only look once version 8. https://github.com/ultralytics/yolov8. Accessed: 2024-06-08.