Karadeniz Fen Bilimleri Dergisi, 16(2), 572-591, 2026

Karadeniz Fen Bilimleri Dergisi
The Black Sea Journal of Sciences

DOI: 10.31466/kfbd.1870662
ISSN (Online): 2564-7377
https://dergipark.org.tr/tr/pub/kfbd

Research Article / Araştırma Makalesi

A Systematic Evaluation of Photometric Data Augmentation Combinations in Medical Object Detection

Tıbbi Nesne Tespitinde Fotometrik Veri Artırma Kombinasyonlarının Sistematik Bir Değerlendirmesi

Onur Erdem KORKMAZ [1]*

Abstract

This study examines how different data augmentation strategies influence medical object detection performance on the Kvasir-SEG dataset by using Faster R-CNN X101-FPN and YOLOv7 as benchmark models. Augmentation is widely used to improve robustness in image classification. However, systematic analyses in object detection are still limited because bounding-box integrity must be preserved during every geometric transformation step. In this work, eight photometric augmentation techniques (Hue, Noise, Saturation, Grayscale, Blur, Brightness, Contrast, and Cutout) were applied independently and in multi-level combinations. Each augmentation was tested in single, double, triple, and full pipelines. Model performance was evaluated through mean Average Precision (mAP) using the COCO evaluation standard. The results show that color-based augmentations improve detection accuracy more than distortion-based augmentations in polyp detection tasks. The results also show that excessive augmentation depth slows model convergence and prevents accuracy gains. This study provides a structured analysis of augmentation depth and diversity on a medical object detection dataset and offers clear guidance for designing effective augmentation pipelines for medical object detection.

Keywords: Deep learning, Computer vision, Data augmentation, Medical object detection.

Öz

Bu çalışma, farklı veri artırma tekniklerinin tıbbi nesne tespiti performansı üzerindeki etkisini Kvasir-SEG veri seti kullanılarak incelemektedir. Faster R-CNN X101-FPN ve YOLOv7 modelleri temel alınmıştır. Veri artırma yöntemleri görüntü sınıflandırmada yaygın biçimde kullanılmaktadır. Ancak nesne tespiti için yapılan sistematik çalışmalar sınırlıdır. Bunun temel nedeni, her geometrik dönüşüm adımında bounding-box bütünlüğünün korunması gerekliliğidir. Bu çalışmada sekiz fotometrik veri artırma tekniği (Hue, Noise, Saturation, Grayscale, Blur, Brightness, Contrast ve Cutout) bağımsız olarak ve çok seviyeli kombinasyonlar şeklinde uygulanmıştır. Her bir yöntem tekli, ikili, üçlü ve tam kombinasyon yapıları içinde test edilmiştir. Model performansı, COCO değerlendirme standardına göre hesaplanan Ortalama Doğruluk (mAP) metriği ile ölçülmüştür. Elde edilen sonuçlar, renk odaklı artırmaların polip tespiti görevlerinde bozulma odaklı yöntemlere kıyasla daha yüksek doğruluk artışı sağladığını göstermektedir. Ayrıca artırma derinliğinin aşırı yükselmesi, öğrenme sürecini yavaşlatmakta ve doğruluk kazanımlarını engellemektedir. Bu çalışma, tıbbi nesne tespitinde veri artırma çeşitliliğinin etkisini sistematik biçimde ortaya koymakta ve etkili artırma yapılarına yönelik uygulayıcılar için net öneriler sunmaktadır.

Anahtar Kelimeler: Derin öğrenme, Bilgisayarlı görü, Veri artırma, Tıbbi nesne tespiti.

[1]Atatürk University, Engineering Faculty, Electrical and Electronic Engineering Department, Erzurum, Türkiye

***Corresponding Author/Sorumlu Yazar:** onurerdem.korkmaz@atauni.edu.tr

Received/Geliş: 24.01.2026 **Accepted/Kabul:** 25.03.2026 **Published/Yayın:** 26.03.2026

Korkmaz, O. E. (2026). A Systematic Evaluation of Photometric Data Augmentation Combinations in Medical Object Detection. *Karadeniz Fen Bilimleri Dergisi*, 16(2), 572-591. https://doi.org/10.31466/kfbd.1870662

1. Introduction

Deep learning has become one of the most widely adopted methodological frameworks across a broad range of application domains in recent years (Chen et al., 2024; Turay et al., 2025; Feng et al., 2021. Among these domains, computer vision (Chen et al., 2024) has occupied a particularly prominent position due to its direct relevance to real-world perception tasks and its rapid technological advancement. Applications such as image classification, object detection, and instance segmentation have significantly benefited from deep learning-based approaches, leading to substantial improvements in performance and generalization capability.

Improving the accuracy of object detection models largely depends on two fundamental factors: the internal properties of the detection algorithm and the overall quality of the dataset used for training (Lin et al., 2014). From the model perspective, properties such as architectural design, network depth and optimization choices play a decisive role in shaping detection performance (Ren et al., 2015). On the dataset side, aspects including data modality, image resolution, class count, intra-class diversity, and annotation reliability are equally important in achieving stable and generalizable results (Everingham et al., 2010).

Among dataset-centered enhancement methods, data augmentation has emerged as one of the most powerful strategies for improving detection accuracy. It encompasses a broad set of operations that artificially increase dataset diversity through transformations such as flipping, rotating, scaling, or altering visual appearance (Mumuni, 2022). Although augmentation has been widely applied in domains such as natural language processing (Pellicer et al., 2023) and other modalities (Iglesias et al., 2023), its role in computer vision has become especially important for reducing overfitting and improving generalization (Mumuni, 2022; Alomar et al., 2023). In vision tasks, commonly used augmentation approaches include geometric and photometric manipulations as well as more advanced region-level, pixel-level, and feature-space techniques (Mumuni, 2022). As illustrated in Figure 1, these methods can be organized into two major categories, data transformation (Alomar et al., 2023) and data synthesis (Mumuni et al., 2024), each contributing different mechanisms for expanding dataset variability. This hierarchical taxonomy provides a unified framework for understanding the scope of augmentation methods and clarifies how individual transformations or their combinations may influence model performance.

Despite its widespread use in computer vision, augmentation has been more comprehensively examined in image classification than in object detection (Cheung, Yeung, 2024). This gap primarily stems from the technical difficulty of applying certain transformations in detection tasks, where flipping, rotation, and scaling operations must be accompanied by consistent updates to bounding-box annotations. Such additional complexity has limited the number of systematic investigations in object detection, even though several studies (Alin et al., 2023; Zoph et al., 2019; Yuan et al., 2022; Yim et al., 2023) have demonstrated the benefits of augmentation in this context. Existing works have generally assessed augmentation methods individually, reporting how a single transformation (such as grayscale conversion or noise injection) affects detection accuracy.

The present study extends beyond these isolated evaluations by systematically analyzing both the individual and combined effects of augmentation strategies on object detection performance. This work focuses specifically on eight photometric augmentation techniques (Kumar et al., 2024a) that leave annotation coordinates unchanged: Hue adjustment, noise addition, saturation modification, grayscale conversion, blur, brightness adjustment, contrast manipulation, and cutout. By applying these techniques separately and in multiple combination schemes, the study investigates whether certain augmentation groupings can produce synergistic improvements in mAP during the training phase. For instance, a three-method augmentation set may outperform more extensive combinations, revealing new insights into the design of augmentation pipelines for object detection.

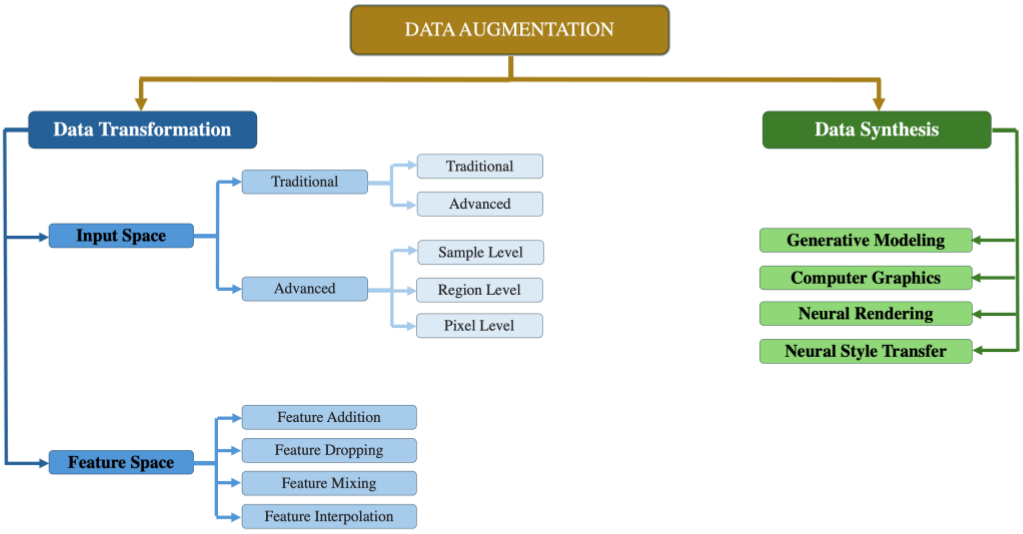


Figure 1. Comprehensive taxonomy of data augmentation techniques.

Data augmentation is widely used in computer vision to enhance model generalization, reduce overfitting, and increase dataset diversity without additional data collection. Augmentation strategies have been applied across various architectures, including Convolutional Neural Networks (CNNs) (Alin et al., 2023) and Vision Transformers (ViTs) (Gao et al., 2024), using geometric transformations such as flipping, rotation, scaling, and cropping, as well as photometric adjustments such as brightness, contrast, hue, and saturation (Mumuni, 2022; Alomar et al., 2023; Kumar et al., 2024a; Nanni et al., 2021). Although these methods have been extensively explored in image classification, their application in object detection is more challenging because geometric operations require consistent updates to bounding-box annotations.

In image classification, a comprehensive study (Nanni et al., 2021) examined more than ten augmentation techniques, including both conventional transforms and signal-processing-based approaches such as Wavelet and Gabor filters. The authors demonstrated that applying augmentations individually reduces overfitting and improves generalization across medical and natural image datasets. Another work (Goceri, 2023) analyzed multiple augmentation methods on medical images from diverse organs and modalities and showed that the effectiveness of each augmentation depends strongly on image characteristics. Both studies focus on isolated, single augmentation techniques. In contrast, one study (Korzhebin, Egorov, 2021) evaluated combinations of augmentations along with different transfer learning strategies across several architectures. The results showed that the best individual techniques did not necessarily achieve the highest accuracy when combined, highlighting the need for systematic studies that examine augmentation interactions.

In the context of object detection, augmentation has also been studied but still mostly at the level of isolated individual techniques. For instance, one work (Alin et al., 2023) applied fourteen augmentation methods on a drone dataset using YOLOv5. The best performance was obtained with mosaic method, while the analysis did not examine interactions between multiple augmentations. In another work, a broader taxonomy of augmentation approaches was introduced in (Kumar et al., 2024b), with experiments on object detection using Faster R-CNN (Ren et al., 2015) and EfficientDet (Tan et al., 2020). Despite providing valuable comparisons, the study still evaluated augmentation techniques individually.

Other works for object detection focused on specialized augmentation families: **(i)** a lightweight YOLO-based pest detection study (Yuan et al., 2022) examined flipping, scaling, HSV perturbation, and mosaic, reporting that geometric transforms were generally beneficial, whereas more complex strategies sometimes caused distribution mismatch. **(ii)** (Yim et al., 2023) introduced an object-oriented cutout method for tiny object detection on VisDrone and modest improvements were reported. **(iii)** On X-ray security imagery, (Luo, Zhu, 2020) tested a small set of augmentations including flips, rotation, brightness adjustment, and a simple hybrid combination. The findings indicated that geometric variations aligned with the dataset improved accuracy, while rotation provided limited benefit. The limitation of study did not incorporate broader photometric or distortion-based methods. **(iv)** Finally, a further domain-specific study (O’shea et al., 2025) proposed yellow-style, gray-style, and night-scene augmentations for aerial aircraft detection. Although these works reported moderate improvements, their methods were tightly coupled to the specific datasets and did not provide a systematic assessment of common geometric or photometric augmentations, nor did they examine combinations of such techniques.

2. Materials and Methods

This study adopts a controlled experimental design to analyze how photometric data augmentation strategies influence object detection performance in a medical imaging context. An online augmentation framework is employed to isolate the effect of augmentation diversity while maintaining a fixed augmentation intensity across all experiments. Eight photometric transformations are evaluated both individually and in structured multi-augmentation configurations, enabling a systematic examination of augmentation depth under consistent training conditions. All experiments are conducted on the Kvasir-SEG (Jha et al., 2019) dataset using two detection architectures with distinct learning paradigms, ensuring that observed performance differences can be attributed to augmentation strategy rather than model-specific or optimization-related factors.

2.1. Custom Dataset

The experiments were conducted using the Kvasir-SEG dataset (Jha et al., 2019), an open-access collection of 1k colonoscopy images and their corresponding pixel-wise polyp segmentation masks. The dataset contains clinically verified annotations produced by experienced gastroenterologists, and it is widely used as a benchmark for polyp segmentation, detection, and localization research. Image resolutions range from 332×487 to 1920×1072 pixels, and each image–mask pair is stored under the same filename structure. Figure 2 illustrates a representative example from the dataset.

Since the original dataset provides only segmentation masks, the ground-truth boxes required for object detection were derived through a mask-to-box extraction process. For each binary mask, all foreground pixels corresponding to polyp tissue were first identified. The minimum and maximum pixel coordinates along the horizontal axis were assigned as x_min and x_max, respectively, while the same procedure was applied along the vertical axis to obtain y_min and y_max. This procedure yields the minimal axis-aligned bounding rectangle that fully encloses the segmented region. All bounding boxes were automatically generated using a custom Python-based algorithm, ensuring a consistent and reproducible conversion from segmentation masks to detection annotations across the entire dataset.

To ensure robust evaluation, the dataset was partitioned into training (%80) and test (%20) subsets following standard practice in medical image analysis. All experiments in this study use these masks-derived bounding boxes as detection ground truth during model training and assessment.

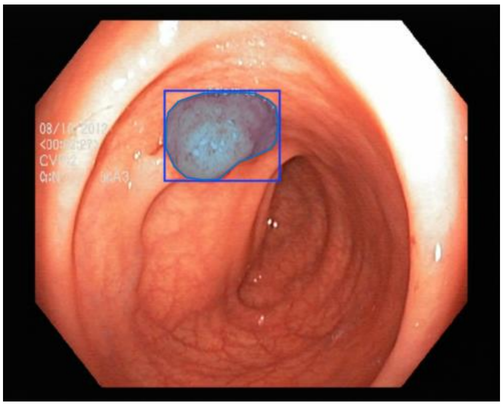


Figure 2. Example image–mask pair from the Kvasir-SEG (Jha et al., 2019) dataset and the corresponding bounding-box representation derived from the segmentation mask.

2.2. Augmentation Methods and Combination Strategy

The eight photometric data augmentation methods (Mumuni, 2022) were selected to systematically evaluate their individual and combined effects on medical object detection performance. These augmentation techniques were grouped into two main categories: color-based transformations and distortion-based transformations, as illustrated in Table 1. Representative examples of each augmentation applied to a sample Kvasir-SEG image are shown in Figure 3.

Color-based augmentations modify the appearance of the image by altering its color distribution while preserving geometric structure. The following five methods fall under this category: **(i)** Hue (H): Adjusts the hue channel within a limited range to simulate variations in illumination color and endoscopy lighting conditions. **(ii)** Saturation (S): Modulates the saturation level to imitate changes in tissue color intensity. **(iii)** Grayscale (GS): Converts the image to monochrome, forcing the model to rely more on texture and structural cues rather than color. **(iv)** Brightness (BR): Adjusts pixel intensity to mimic underexposed or overexposed imaging conditions frequently observed in clinical procedures. **(v)** Contrast (C): Enhances or suppresses local intensity differences, affecting the visibility of polyp boundaries and surrounding mucosal texture.

Table 1. Summary of single and multi-augmentation combinations used for evaluation, categorized into color-based and distortion-based groups.

ID	Augmentation Combinations	Description
1	Baseline (B)	No augmentation
2	Hue (H)	Single color-based augmentation
3	Saturation (S)	Single color-based augmentation
4	Grayscale (GS)	Single color-based augmentation
5	Brightness (BR)	Single color-based augmentation
6	Contrast (C)	Single color-based augmentation
7	Noise (N)	Single distortion-based augmentation
8	Blur (BL)	Single distortion-based augmentation
9	Cutout (CO)	Single distortion-based augmentation

10	Brightness+Contrast (BRC)	Double color-based combination
11	Noise + Blur (NBL)	Double distortion-based combination
12	Brightness+Cutout (BRCO)	Double cross-family combination
13	Noise+Blur+Cutout (NBLCO)	Triple full distortion-based
14	Hue+Saturation+Grayscale+ Brightness+Contrast (HSGSBRC)	Quintuple full color-based
15	All Augmentations Combined (ALL)	Complete set of eight augmentations

Distortion-based augmentations introduce stochastic perturbations or occlusions that increase robustness to real-world acquisition artifacts. These include: **(i)** Noise (N): Adds random pixel-level noise that simulates sensor imperfections or compression artifacts. **(ii)** Blur (BL): Applies Gaussian blurring to emulate motion blur or out-of-focus imaging typical in endoscopic video sequences. **(iii)** Cutout (CO): Masks out rectangular image regions to encourage the model to learn contextual cues and remain robust to partial occlusions.

Beyond single-method evaluations, multi-augmentation combinations were constructed to assess whether interacting transformations produce synergistic or antagonistic effects. Table 1 summarizes all augmentation scenarios considered in this study, ranging from isolated augmentations to double, triple, quintuple, and full eight-method combinations. These structured combinations enable an examination of how augmentation depth and diversity influence detection performance under a unified probabilistic online augmentation framework.

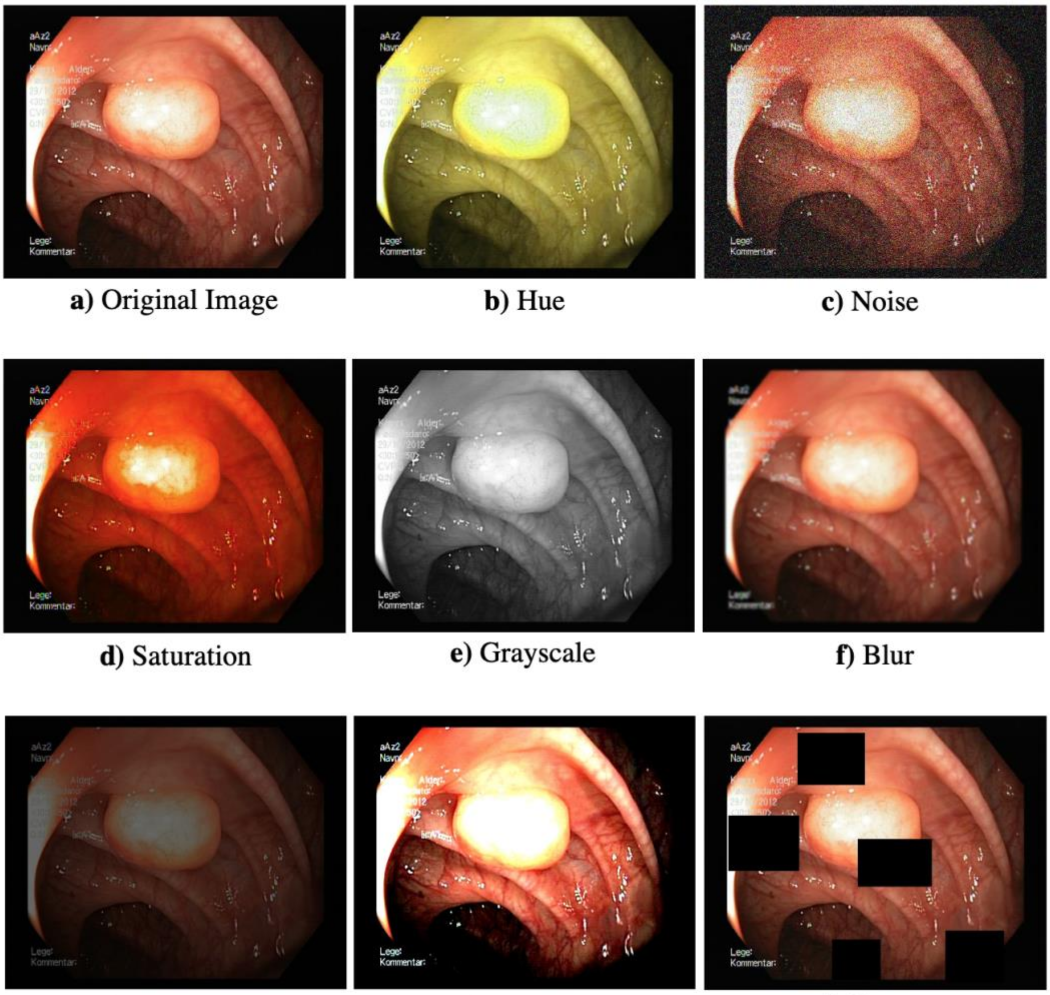
2.3. Methodological Framework

Unlike traditional offline augmentation, where augmented samples are generated once and stored, the online framework applies transformations dynamically during every training iteration. This ensures that each image can appear in many altered forms across epochs, increasing diversity without enlarging the dataset size.

Online augmentation is particularly advantageous for medium-scale datasets such as Kvasir-SEG, as it continuously exposes the model to new stochastic variants and prevents memorization of fixed augmented samples. All eight augmentation methods defined in Section 2.2 were therefore integrated into a unified online pipeline to allow controlled comparisons between single-method and multi-method augmentation scenarios.

2.3.1. Probability design and combination logic

To evaluate how augmentation depth and diversity affect detection accuracy, three augmentation configurations were used: **(i)** Single augmentations: Each augmentation is applied with probability $p = 0.50$ independently. **(ii)** Multi-augmentation combinations (double, triple, quintuple): For combinations, the total augmentation intensity was kept constant at 50%, shared equally among the included methods. For example, for a double combination, each method is applied with $p = 0.25$. Another, for a triple combination, each method is applied with $p \approx 0.17$. **(iii)** Full eight-method augmentation (ALL): The eight methods share the same 50% total transformation intensity. Each augmentation is applied with $p = \approx 0.06$ (50% ÷ 8). These probabilities follow independent Bernoulli (Ross, 2014) trials, meaning each method activates independently with probability 0.06.



a) Original Image

b) Hue

c) Noise

d) Saturation

e) Grayscale

f) Blur

g) Brightness

h) Contrast

i) Cutout

Figure 3. Representative examples of the eight photometric augmentation methods applied to a Kvasir-SEG image (Jha et al., 2019): (a) Original, (b) Hue, (c) Noise, (d) Saturation, (e) Grayscale, (f) Blur, (g) Brightness, (h) Contrast, and (i) Cutout.

Although each method is evaluated independently, the *expected number of transformations per image* remains stable around 0.5, ensuring a fair comparison across augmentation depths. Because augmentations are applied independently in an online setting, each batch statistically preserves the desired 50% augmentation rate, provided the batch size is ≥ 16.

2.4. Implementation using PyTorch Transforms

The augmentation pipeline was implemented using the torchvision.transforms (PyTorch, 2025) interface. Each augmentation was wrapped in a RandomApply module, which allows assigning an explicit activation probability. A typical transformation code block in PyTorch takes the form: RandomApply ([AugmentationFunction(parameters)], *p*=prob).

In the online framework, each augmentation is evaluated independently for every image, and if triggered, it modifies the current sample before it is passed to the model. This process preserves statistical independence across transformations and provides a transparent, reproducible implementation suitable for systematic analysis.

A representative implementation (illustrative code snippet) of the online augmentation pipeline for the full eight-method augmentation setting (ALL) is provided for reproducibility at Appendix. In this illustrative configuration, the activation probability for each method is set to $p = 0.06$, because the total augmentation intensity is evenly distributed across eight transformations. If a single augmentation method had been used instead, the activation probability would have been $p = 0.50$ to maintain an equivalent augmentation strength.

In all experiments, each augmentation technique was applied using fixed parameter settings. These settings were preserved across all single and combined augmentation scenarios to ensure methodological consistency. The hue adjustment (H) was implemented using ColorJitter(hue=0.2), which shifts the color tone of the image within a controlled range. This transformation was triggered with probability p_each and broadens the color spectrum encountered during training. Saturation modification (S) was performed with ColorJitter(saturation=0.3), enabling the model to remain robust under highly vivid or desaturated imaging conditions. Grayscale conversion (GS) used Grayscale(num_output_channels=3) to remove chromatic information while preserving the expected channel structure for the network. Brightness adjustment (BR) followed ColorJitter(brightness=0.3), introducing luminance variability that simulates both underexposed and overexposed endoscopic frames. Contrast modification (C) relied on ColorJitter(contrast=0.3), enhancing or reducing intensity differences to reflect fluctuations commonly observed on the colon surface. Image noise (N) was added through a custom Gaussian noise module (GaussianNoise(std=0.05)), which injects pixel-level perturbations similar to sensor noise or compression artifacts. Blurring (BL) was introduced using GaussianBlur(kernel_size=5), mimicking defocus, motion artifacts, or optical degradation. Finally, the cutout operation (CO) was implemented via custom RandomErasing(scale=(0.02, 0.15)) module, masking a randomly selected region of the image to reproduce occlusion effects such as fluids, specular highlights, or partially obstructing instruments.

Across all augmentation scenarios, these parameter values remained unchanged. Only the application probability differed depending on the experimental design, while the underlying transformations and their strengths were kept constant throughout the study.

2.5. Training Procedure and Deployed Models

Two state-of-the-art object detection frameworks were used in this study: Faster R-CNN X101–FPN implemented in Detectron2 (Detectron2, 2025) and YOLOv7 implemented in the Ultralytics framework (Wang et al., 2023). Both models were trained under a unified training schedule to ensure fair comparison across augmentation scenarios. Training was performed for 50 epochs with a batch size of 32. This batch size was selected as it provides a stable gradient estimate, maintains sufficient stochasticity for online augmentation, and fits comfortably within the memory constraints of the A40 GPU cluster.

The initial learning rate was set to 0.001 and decayed by a factor of 0.1 after 30 epochs using a step-based scheduler, which improved convergence stability in both architectures. Weight decay of 10^{-4} was applied throughout all experiments to mitigate overfitting, and optimization was performed using stochastic gradient descent (SGD) with a momentum coefficient of 0.9. For YOLOv7, hyperparameters were adapted from the official Ultralytics configuration with minimal adjustments to maintain comparability with the Faster R-CNN settings.

All experiments were executed on the High Performance Computing Center of Atatürk University using an 8×NVIDIA A40 GPU cluster, each with 48 GB VRAM. This high-performance infrastructure enabled efficient multi-scenario training and ensured full reproducibility across all augmentation configurations.

3. Findings and Discussion

During training, the optimization process was monitored using multiple loss components, including classification loss (loss_cls), bounding box regression loss (loss_box_reg), region proposal network (RPN) classification loss (loss_rpn_cls), and RPN localization loss (loss_rpn_loc). The sum of these terms is reported as the total loss, which was tracked across epochs to evaluate convergence behavior. However, the most effective and widely accepted evaluation criterion in object detection is the mAP, since it directly reflects both classification accuracy and localization precision on unseen data. In this study, results are therefore consistently reported using mAP (mAP@[.5:.95]), mAP@.5, and mAP@.75 following the COCO evaluation protocol (Lin et al., 2014).

Unlike validation loss, which primarily captures short-term fluctuations in optimization and may not strongly correlate with final detection accuracy, mAP provides a more reliable measure of model performance in practice. Indeed, recent benchmarks have emphasized that loss reduction does not necessarily translate into higher mAP scores, particularly when augmentation-induced regularization effects are present. To ensure robustness and comparability, the COCO Evaluator tool was employed for all evaluations (Detectron2, 2025b; Lin et al., 2014).

Table 2. Detection Performance of Faster R-CNN X101-FPN and YOLOv7 across Single and Combined Augmentation Scenarios.

ID	**Augmentation Combinations**	**Faster R-CNN X101-FPN** mAP* / mAP@.5 / mAP@.75	**YOLOv7** mAP / mAP@.5 / mAP@.75
1	Baseline (B)	0.612 / 0.823 / 0.571	0.598 / 0.844 / 0.512
2	Hue (H)	0.638 / 0.846 / 0.593	0.622 / 0.861 / 0.537
3	Saturation (S)	0.645 / 0.851 / 0.601	0.628 / 0.867 / 0.545
4	Grayscale (GS)	0.633 / 0.839 / 0.588	0.615 / 0.855 / 0.526
5	Brightness (BR)	**0.657** / 0.861 / 0.612	**0.642** / 0.874 / 0.553
6	Contrast (C)	0.651 / 0.858 / 0.607	0.637 / 0.872 / 0.548
7	Noise (N)	0.618 / 0.829 / 0.566	0.603 / 0.841 / 0.508
8	Blur (BL)	0.611 / 0.823 / 0.559	0.594 / 0.837 / 0.501
9	Cutout (CO)	0.624 / 0.831 / 0.574	0.609 / 0.846 / 0.518
10	Brightness+Contrast (BRC)	**0.668** / 0.872 / 0.624	**0.654** / 0.887 / 0.566
11	Noise + Blur (NBL)	0.602 / 0.815 / 0.547	0.589 / 0.828 / 0.492
12	Brightness+Cutout (BRCO)	0.661 / 0.867 / 0.619	0.648 / 0.882 / 0.561
13	Noise+Blur+Cutout (NBLCO)	0.595 / 0.809 / 0.538	0.582 / 0.822 / 0.487
14	Hue+Saturation+Grayscale+ Brightness+Contrast (HSGSBRC)	**0.675** / 0.879 / 0.631	**0.662** / 0.893 / 0.575
15	All Augmentations Combined (ALL)	0.608 / 0.821 / 0.563	0.595 / 0.838 / 0.505

*mAP@[.5, .95]

3.1. Comparative Analysis of Color-Based and Distortion-Based Augmentations on mAP

A systematic comparison of color-based and distortion-based augmentations reveals tentative yet observable trends regarding how appearance-level and structure-level transformations may influence object detection accuracy on the Kvasir-SEG dataset. As summarized in Table 2, color-based augmentations (IDs 2-6) generally tend to provide accuracy improvements for both Faster R-CNN X101-FPN and YOLOv7, whereas distortion-based methods (IDs 7-9) appear more sensitive and may lead to mixed or occasionally reduced performance, particularly at higher IoU thresholds.

The analysis is primarily conducted on mAP (mAP@[.5:.95]), which serves as the most stringent and informative metric. Supplementary results for mAP@.5 and mAP@.75 are included to offer additional perspective. These latter values, although widely reported in the literature, reflect performance at single IoU thresholds and therefore offer a more limited view. In our evaluations, the tendencies observed in mAP@[.5:.95] were also reflected in mAP@.5 and mAP@.75, which suggests consistency across metrics while still treating the single-threshold values as supportive rather than definitive indicators.

Among the eight individual augmentation methods, Brightness (BR) and Contrast (C) show the most notable improvements. For Faster R-CNN, BR increases mAP from 0.612 to 0.657, while YOLOv7 exhibits a similar gain from 0.598 to 0.642. These increases may imply that controlled illumination variation helps detectors better accommodate the diverse lighting patterns frequently encountered in endoscopic imaging. Saturation (S) and Hue (H) also produce positive changes, although at smaller magnitudes. These patterns suggest that enrichment of color variability can support generalization, provided that the underlying structural information of the polyps remains intact.

Compared to color-based augmentations, distortion-based transformations exhibit inconsistent or sometimes adverse effects on detection accuracy. Noise (N) and Blur (BL) tend to reduce performance to some extent. For instance, BL changes Faster R-CNN performance from 0.612 to 0.611 and YOLOv7 from 0.598 to 0.594, with more pronounced sensitivity at mAP@0.75. Cutout (CO) performs relatively better within the distortion group, increasing Faster R-CNN mAP to 0.624. This modest improvement may indicate that limited occlusion encourages models to rely on contextual cues, although excessive masking could also remove clinically relevant texture patterns and thereby hinder localization accuracy.

Two- and three-augmentation combinations highlight these differences more clearly. The Brightness+Contrast (BRC) combination achieves one of the strongest results (mAP = 0.668 for Faster R-CNN and 0.654 for YOLOv7), exceeding the performance of any individual method. This may indicate that complementary illumination variations produce a cumulative benefit. In comparison, Noise+Blur (NBL) decreases performance in both detectors, suggesting that concurrent structural distortions may amplify visibility challenges, particularly around polyp boundaries. Similarly, the triple combination NBLCO yields the lowest scores among the evaluated combinations, reinforcing the observation that aggressive distortion can interfere with fine-grained texture cues important for accurate detection.

Higher-dimensional color-based combinations produce the most favorable outcomes. The five-method combination HSGSBRC, which integrates Hue, Saturation, Grayscale, Brightness, and Contrast, achieves the strongest results for both detectors (mAP = 0.675 for Faster R-CNN and 0.662 for YOLOv7). These findings may suggest that broader color diversity contributes positively to model robustness, as long as structural fidelity is maintained throughout training.

Although the final configuration, ALL, applies all eight augmentation types simultaneously, its performance returns to values closer to the baseline. This behavior may arise from the simultaneous application of both beneficial color variations and disruptive structural distortions, where the latter may offset some of the potential gains offered by the former. Taken together, these outcomes imply that increasing augmentation diversity does not necessarily guarantee higher accuracy; rather, augmentation strategies may require careful balancing to preserve clinically meaningful visual information.

3.2. Convergence Trends

The convergence characteristics of the training process are illustrated through four figures. Figures 4 and 5 present total loss trajectories for Faster R-CNN and YOLOv7 under the baseline and eight single-augmentation scenarios. Figures 6 and 7 extend this analysis to multi-augmentation settings (Table 1), comparing color-based, distortion-based, cross-family, and full-combination approaches. Collectively, these visualizations help contextualize how augmentation strategies may influence optimization behavior, stability, and the eventual performance margins between detectors.

3.2.1. Under single-augmentation application

Both detectors display stable and interpretable convergence patterns under single-augmentation settings. A consistent observation across Figures 4 and 5 is that augmentation-enhanced training curves tend to lie below the baseline (depicted with a thicker red line in both plots), suggesting lower residual loss by the end of training. This tendency parallels the accuracy outcomes in Table 2, where baseline and blur configurations are associated with the weakest mAP scores.

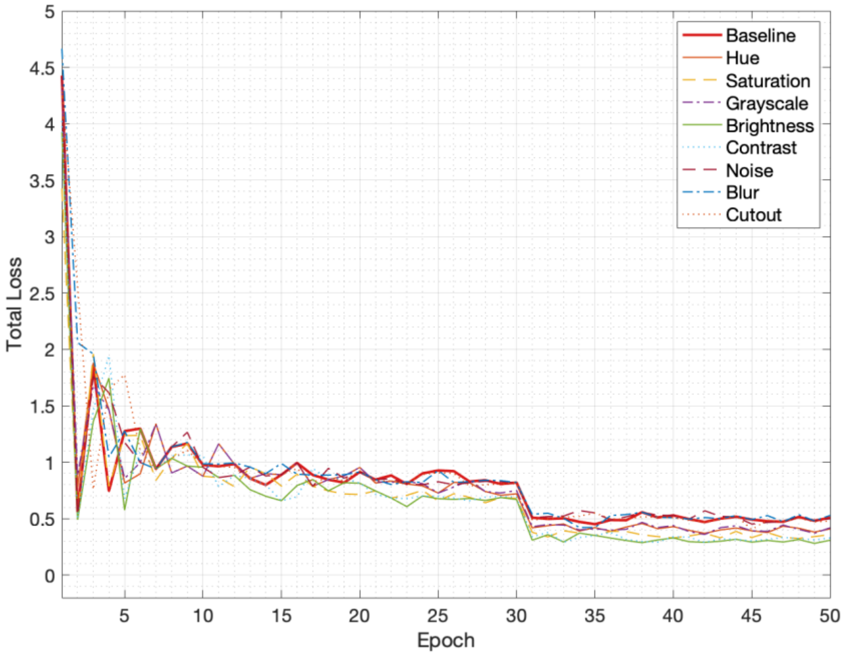


Figure 4. Faster R-CNN: Total Loss vs. Epochs (Baseline and Single-Augmentation Scenarios).

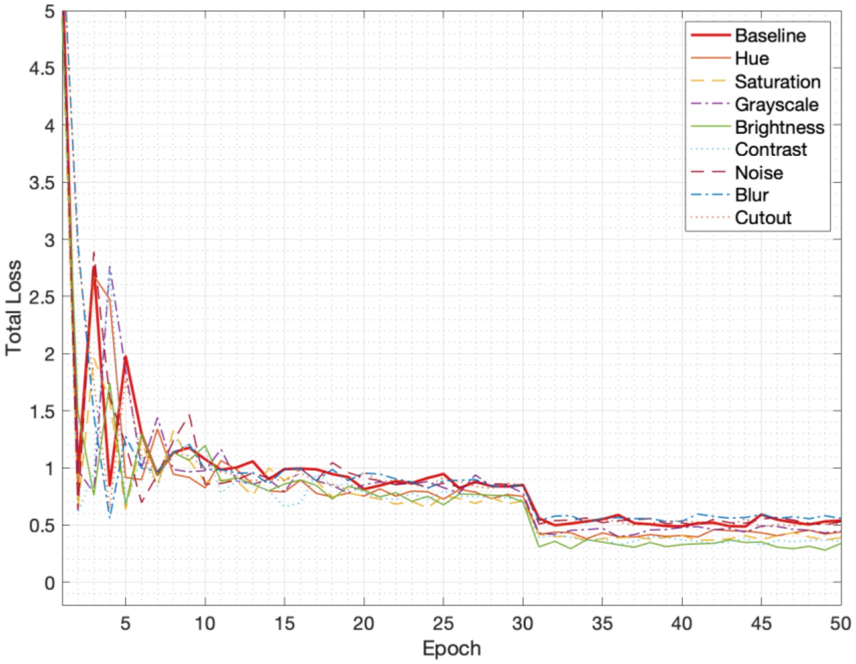


Figure 5. YOLOv7: Total Loss vs. Epochs (Baseline and Single-Augmentation Scenarios).

YOLOv7 (Figure 5) exhibits more pronounced oscillations during the first 10 epochs compared with Faster R-CNN (Figure 4). This behavior aligns with its one-stage architecture, where classification and localization are optimized simultaneously. Such coupling may increase early sensitivity to data variability introduced by augmentation. Faster R-CNN's two-stage formulation (proposal generation followed by classification) appears to dampen this sensitivity, yielding smoother early-epoch dynamics.

Across methods, augmentations such as Brightness and Contrast achieve noticeably lower final loss values, consistent with their comparatively higher mAP scores. This alignment between convergence behavior and detection accuracy suggests that certain appearance-level perturbations may support stronger generalization without disrupting the structural characteristics of the underlying medical images.

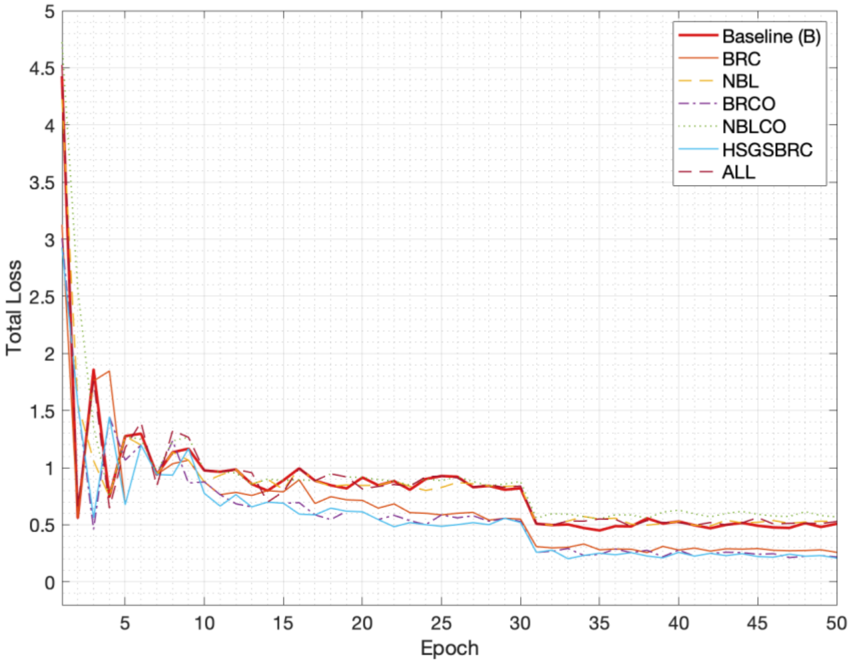


Figure 6. Faster R-CNN: Total Loss vs. Epochs (Baseline and Multi-Augmentation Scenarios)

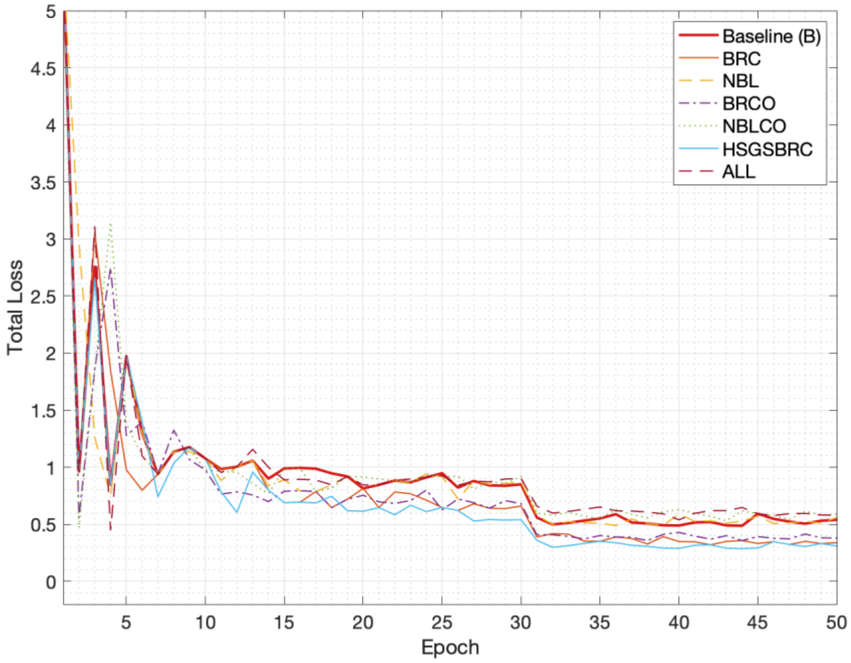


Figure 7. YOLOv7: Total Loss vs. Epochs (Baseline and Multi-Augmentation Scenarios)

In summary, the convergence curves indicate that augmentation strategies frequently produce measurable improvements compared with the baseline, though the magnitude varies across techniques. YOLOv7 shows higher early-phase volatility, yet both models ultimately benefit (at varying degrees) from appropriate augmentation, as reflected in their lower final losses and higher mAP values.

3.2.2. Under multiple-augmentation application

The multi-augmentation results provide additional insight into how combined transformations influence model optimization. In Figure 6, Faster R-CNN demonstrates relatively stable behavior for color-based combinations, especially the HSGSBRC configuration (Hue + Saturation + Grayscale + Brightness + Contrast). This scenario consistently follows the lowest loss trajectory, aligning with its strong mAP results. These trends suggest that well-structured color-based combinations may introduce beneficial diversity without compromising essential structural cues.

By contrast, distortion-dominant combinations such as NBL, BRCO, and NBLCO converge to higher loss values and correspondingly lower mAP in Table 2. The ALL configuration is particularly noteworthy: its loss curve remains above the baseline throughout training, and its accuracy metrics reflect similar limitations. This pattern implies that applying numerous heterogeneous augmentations indiscriminately may introduce excessive regularization and hinder model optimization.

YOLOv7 (Figure 7) mirrors many of these trends but with amplified early volatility in multi-augmentation settings, especially those involving multiple distortion methods (e.g., NBLCO and ALL). This behavior is consistent with the model’s architectural sensitivity and the elevated input variability generated by these transformations. While convergence eventually stabilizes, the final loss remains higher than that of color-based configurations, reinforcing the relative difficulty posed by aggressive structural augmentations.

Taken together, these results suggest that while moderate multi-augmentation (particularly color-based combinations) can support improved learning dynamics, extensive augmentation across diverse transformation families may yield diminishing returns or even negative effects. Future studies may examine these behaviors on larger-scale benchmarks such as MS COCO or Pascal VOC to further assess their generalizability.

4. Conclusions and Recommendations

This study provides a systematic evaluation of how single and combined augmentation strategies affect training convergence and detection performance. The primary finding is that color-based augmentations generally contribute positively to both convergence behavior and accuracy, whereas distortion-based methods require more caution, as they may introduce disruptive artifacts. Furthermore, extensive augmentation (particularly when combining incompatible transformations) can impose strong regularization pressures that limit convergence and degrade performance.

By analyzing augmentation strategies individually and in combination, the study highlights the importance of balanced design choices. Structured color transformations tend to yield the clearest improvements, whereas indiscriminate mixing, as in the ALL configuration, may counteract potential benefits. These insights help clarify the complex, nonlinear interactions between augmentation types and model optimization, particularly in medical imaging settings where preserving structural fidelity is essential.

Acknowledgements

The author would like to thank the High Performance Computing Center of Atatürk University for providing the computational resources used in the data analysis of this study.

Authors' Contributions

The author contributed solely to all stages of the study.

Statement of Conflicts of Interest

There is no conflict of interest.

Statement of Research and Publication Ethics

The author declares that this study complies with Research and Publication Ethics.

References

Alin, A. Y., Kusrini, & Yuana, K. A. (2023). Data Augmentation Method on Drone Object Detection with YOLOv5 Algorithm. *2023 8th International Conference on Informatics and Computing, ICIC 2023*. https://doi.org/10.1109/ICIC60109.2023.10382123

Alomar, K., Aysel, H. I., & Cai, X. (2023). Data Augmentation in Classification and Segmentation: A Survey and New Strategies. *Journal of Imaging 2023, Vol. 9, Page 46*, *9*(2), 46. https://doi.org/10.3390/JIMAGING9020046

Cerqueira, V., Santos, M., Roque, L., Baghoussi, Y., & Soares, C. (2024). Online Data Augmentation for Forecasting with Deep Learning. *Lecture Notes in Computer Science*, *16121 LNAI*, 217–229. https://doi.org/10.1007/978-3-032-05176-9_17

Chen, J., Zhu, S., & Luo, W. (2024). Instance segmentation of underwater images by using deep learning. Electronics, 13(2), 274. https://doi.org/10.3390/electronics13020274

Cheung, T. H., & Yeung, D. Y. (2024). A Survey of Automated Data Augmentation for Image Classification: Learning to Compose, Mix, and Generate. *IEEE Transactions on Neural Networks and Learning Systems*, *35*(10), 13185–13205. https://doi.org/10.1109/TNNLS.2023.3282258

Detectron2. (n.d.). *MODEL_ZOO*. Retrieved September 15, 2025, from https://github.com/facebookresearch/detectron2/blob/main/MODEL_ZOO.md

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*, *88*(2), 303–338. https://doi.org/10.1007/S11263-009-0275-4/METRICS

Feng, S. Y., Gangal, V., Wei, J., Chandar, S., Vosoughi, S., Mitamura, T., & Hovy, E. (2021). A survey of data augmentation approaches for NLP. Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, 968–988. https://doi.org/10.18653/v1/2021.findings-acl.84

Gao, X., Xiao, Z., & Deng, Z. (2024). High accuracy food image classification via vision transformer with data augmentation and feature augmentation. *Journal of Food Engineering*, *365*, 111833. https://doi.org/10.1016/J.JFOODENG.2023.111833

Goceri, E. (2023). Medical image data augmentation: techniques, comparisons and interpretations. *Artificial Intelligence Review*, *56*(11), 12561–12605. https://doi.org/10.1007/S10462-023-10453-Z/TABLES/10

Iglesias, G., Talavera, E., González-Prieto, Á., Mozo, A., & Gómez-Canaval, S. (2023). Data Augmentation techniques in time series domain: a survey and taxonomy. *Neural Computing and Applications*, *35*(14), 10123–10145. https://doi.org/10.1007/S00521-023-08459-3/FIGURES/9

Jha, D., Smedsrud, P. H., Riegler, M. A., Halvorsen, P., de Lange, T., Johansen, D., & Johansen, H. D. (2019). Kvasir-SEG: A Segmented Polyp Dataset. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *11962 LNCS*, 451–462. https://doi.org/10.1007/978-3-030-37734-2_37

Korzhebin, T. A., & Egorov, A. D. (2021). Comparison of Combinations of Data Augmentation Methods and Transfer Learning Strategies in Image Classification Used in Convolution Deep Neural Networks. *Proceedings of the 2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering, ElConRus 2021*, 479–482. https://doi.org/10.1109/ELCONRUS51938.2021.9396724

Kumar, T., Brennan, R., Mileo, A., & Bendechache, M. (2024a). Image Data Augmentation Approaches: A Comprehensive Survey and Future Directions. *IEEE Access*, *12*, 187536–187571. https://doi.org/10.1109/ACCESS.2024.3470122

Kumar, T., Brennan, R., Mileo, A., & Bendechache, M. (2024b). Image Data Augmentation Approaches: A Comprehensive Survey and Future Directions. *IEEE Access*, *12*, 187536–187571. https://doi.org/10.1109/ACCESS.2024.3470122

Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft COCO: Common Objects in Context. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *8693 LNCS*(PART 5), 740–755. https://doi.org/10.1007/978-3-319-10602-1_48

Luo, Y., & Zhu, L. (2020). Research on Data Augmentation for Object Detection Based on X-ray Security Inspection Picture. *Proceedings of 2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications, AEECA 2020*, 219–222. https://doi.org/10.1109/AEECA49918.2020.9213654

Mumuni, A., & Mumuni, F. (2022). Data augmentation: A comprehensive survey of modern approaches. *Array*, *16*, 100258. https://doi.org/10.1016/J.ARRAY.2022.100258

Mumuni, A., Mumuni, F., & Gerrar, N. K. (2024). A survey of synthetic data augmentation methods in computer vision. *Machine Intelligence Research*, *21*(5), 831–869. https://doi.org/10.1007/s11633-022-1411-7

Nanni, L., Paci, M., Brahnam, S., & Lumini, A. (2021). Comparison of Different Image Data Augmentation Approaches. *Journal of Imaging 2021, Vol. 7, Page 254*, *7*(12), 254. https://doi.org/10.3390/JIMAGING7120254

O'shea, R. P., Singh, G., Goodman, A. B., Keane, T. J., Brenner, M. A., Jaworowski, C. J., Patel, T. A., & Hing, J. T. (2025). Comparison of Augmentation Techniques with Stable Diffusion for Aircraft Identification. *Journal of Image and Graphics (United Kingdom)*, *13*(2), 151–157. https://doi.org/10.18178/JOIG.13.2.151-157

Pellicer, L. F. A. O., Ferreira, T. M., & Costa, A. H. R. (2023). Data augmentation techniques in natural language processing. *Applied Soft Computing*, *132*, 109803. https://doi.org/10.1016/J.ASOC.2022.109803

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(6), 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031

Ross, S. M. (2014). Introduction to Probability Models. *Introduction to Probability Models: Eleventh Edition*, 1–767. https://doi.org/10.1016/C2012-0-03564-8

Tan, M., Pang, R., & Le, Q. V. (2020). EfficientDet: Scalable and efficient object detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 10778–10787. https://doi.org/10.1109/CVPR42600.2020.01079

torchvision.transforms — Torchvision 0.8.1 documentation. (n.d.). Retrieved December 19, 2025, from https://docs.pytorch.org/vision/0.8/transforms.html

Turay, T., Korkmaz, O. E., & Ergün, E. (2025). Simultaneous EEG-fNIRS study of visual cognitive processing: ERP analysis and decision-related hemodynamic responses in healthy adults. PLOS ONE, 20(6), e0325017. https://doi.org/10.1371/journal.pone.0325017

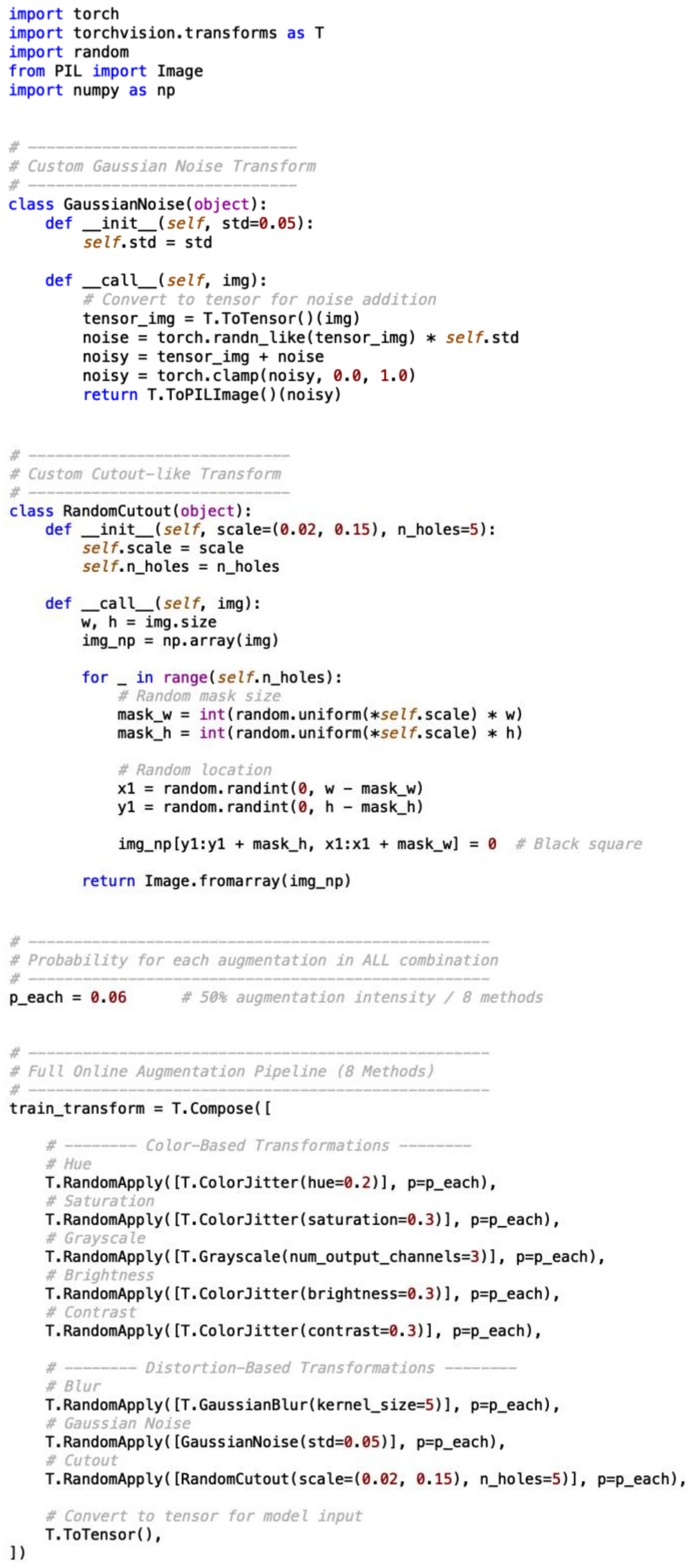
Wagner, F., Eltner, A., & Maas, H. G. (2023). River water segmentation in surveillance camera images: A comparative study of offline and online augmentation using 32 CNNs. *International Journal of Applied Earth Observation and Geoinformation*, *119*, 103305. https://doi.org/10.1016/J.JAG.2023.103305

Yim, S., Cho, M. A., & Lee, S. (2023). Object-Oriented Cutout Data Augmentation for Tiny Object Detection. *2023 International Technical Conference on Circuits/Systems, Computers, and Communications, ITC-CSCC 2023*. https://doi.org/10.1109/ITC-CSCC58803.2023.10212481

Yuan, Z., Li, S., Yang, P., & Li, Y. (2022). Lightweight Object Detection Model with Data Augmentation for Tiny Pest Detection. *IEEE International Conference on Industrial Informatics (INDIN)*, *2022-July*, 233–238. https://doi.org/10.1109/INDIN51773.2022.9976137

Zoph, B., Cubuk, E. D., Ghiasi, G., Lin, T. Y., Shlens, J., & Le, Q. V. (2019). Learning Data Augmentation Strategies for Object Detection. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *12372 LNCS*, 566–583. https://doi.org/10.1007/978-3-030-58583-9_34

APPENDIX



```
import torch
import torchvision.transforms as T
import random
from PIL import Image
import numpy as np


# ----------------------------
# Custom Gaussian Noise Transform
# ----------------------------
class GaussianNoise(object):
    def __init__(self, std=0.05):
        self.std = std

    def __call__(self, img):
        # Convert to tensor for noise addition
        tensor_img = T.ToTensor()(img)
        noise = torch.randn_like(tensor_img) * self.std
        noisy = tensor_img + noise
        noisy = torch.clamp(noisy, 0.0, 1.0)
        return T.ToPILImage()(noisy)


# ----------------------------
# Custom Cutout-like Transform
# ----------------------------
class RandomCutout(object):
    def __init__(self, scale=(0.02, 0.15), n_holes=5):
        self.scale = scale
        self.n_holes = n_holes

    def __call__(self, img):
        w, h = img.size
        img_np = np.array(img)

        for _ in range(self.n_holes):
            # Random mask size
            mask_w = int(random.uniform(*self.scale) * w)
            mask_h = int(random.uniform(*self.scale) * h)

            # Random location
            x1 = random.randint(0, w - mask_w)
            y1 = random.randint(0, h - mask_h)

            img_np[y1:y1 + mask_h, x1:x1 + mask_w] = 0  # Black square

        return Image.fromarray(img_np)


# ---------------------------------------------------
# Probability for each augmentation in ALL combination
# ---------------------------------------------------
p_each = 0.06      # 50% augmentation intensity / 8 methods


# ---------------------------------------------------
# Full Online Augmentation Pipeline (8 Methods)
# ---------------------------------------------------
train_transform = T.Compose([

    # -------- Color-Based Transformations --------
    # Hue
    T.RandomApply([T.ColorJitter(hue=0.2)], p=p_each),
    # Saturation
    T.RandomApply([T.ColorJitter(saturation=0.3)], p=p_each),
    # Grayscale
    T.RandomApply([T.Grayscale(num_output_channels=3)], p=p_each),
    # Brightness
    T.RandomApply([T.ColorJitter(brightness=0.3)], p=p_each),
    # Contrast
    T.RandomApply([T.ColorJitter(contrast=0.3)], p=p_each),

    # -------- Distortion-Based Transformations --------
    # Blur
    T.RandomApply([T.GaussianBlur(kernel_size=5)], p=p_each),
    # Gaussian Noise
    T.RandomApply([GaussianNoise(std=0.05)], p=p_each),
    # Cutout
    T.RandomApply([RandomCutout(scale=(0.02, 0.15), n_holes=5)], p=p_each),

    # Convert to tensor for model input
    T.ToTensor(),
])
```

Figure A1. A Reference PyTorch Implementation (Illustrative Code Snippet) of the Online Augmentation Pipeline. The Code is Provided for Reproducibility and Illustrative Purposes.