**Iğdır Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 14(1), 8-17, 2024**
***Journal of the Institute of Science and Technology, 14(1), 8-17, 2024***

**ISSN: 2146-0574, eISSN: 2536-4618**

**Computer Engineering** **DOI: 10.21597/jist.1328255**

**Research Article**

**Received: 16.07.2023** **Accepted: 08.12.2023**

**To Cite:** Tekin, A. & Bozkır, A. S. (2024). Enhance or Leave It: An Investigation of the Image Enhancement in Small Object Detection in Aerial Images. *Journal of the Institute of Science and Technology*, 14(1), 8-17.

# Enhance or Leave It: An Investigation of the Image Enhancement in Small Object Detection in Aerial Images

Alpay TEKİN[1*], Ahmet Selman BOZKIR[1]

**Highlights:**

- YOLOV8
- YOLOV7
- YOLOV6
- Deep Learning
- MPRNet

**Keywords:**

- Object Detection
- Image Restoration
- MPRNet
- Single Shot Object Detection

**ABSTRACT:**

Recent years of object detection (OD), a fundamental task in computer vision, have witnessed the rise of numerous practical applications of this sub-field such as face detection, self-driving, security, and more. Although existing deep learning models show significant achievement in object detection, they are usually tested on datasets having mostly clean images. Thus, their performance levels were not measured on degraded images. In addition, images and videos in real-world scenarios often involve several natural artifacts such as noise, haze, rain, dust, and motion blur due to several factors such as insufficient light, atmospheric scattering, and faults in image sensors. This image acquisition-related problem becomes more severe when it comes to detecting small objects in aerial images. In this study, we investigate the small object identification performance of several state-of-the-art object detection models (Yolo 6/7/8) under three conditions (noisy, motion blurred, and rainy). Through this inspection, we evaluate the contribution of an image enhancement scheme so-called MPRNet. For this aim, we trained three OD algorithms with the original clean images of the VisDrone dataset. Followingly, we measured the detection performance of saved YOLO models against (1) clean, (2) degraded, and (3) enhanced counterparts. According to the results, MPRNet-based image enhancement promisingly contributes to the detection performance and YOLO8 outperforms its predecessors. We believe that this work presents useful findings for researchers studying aerial image-based vision tasks, especially under extreme weather and image acquisition conditions

[1]Alpay TEKİN (**Orcid ID:** 0009-0001-2858-1228), Ahmet Selman BOZKIR (**Orcid ID:** 0000-0003-4305-7800), Hacettepe University, Department of Computer Engineering, Ankara, Türkiye

*** Corresponding Author:** Alpay TEKİN, e-mail: alpaytekin@hacettepe.edu.tr

## INTRODUCTION

Object detection is a fundamental task in computer vision that involves identifying and localizing objects within images. It plays a crucial role in various applications, including autonomous driving, surveillance systems, robotics, and augmented reality. Over the years, significant advancements have been made in object detection techniques, particularly with the emergence of deep-learning models especially convolutional neural networks (CNNs). From a general perspective, object detection algorithms can be examined under two primary pipelines, two-stage and one-stage approaches.

The two-stage approach performs object detection in two stages: (1) extracting regions of interest (RoIs) and (2) classifying and regressing the RoIs (i.e. Region of Interest). The R-CNN (Girshick et al., 2014) uses a selective search algorithm (Uijlings et al.,2013) to extract region proposals. On the other hand, CNN networks process the proposed regions and extract feature maps. Following the extraction of feature maps, the SVM model classifies each RoI independently. However, this scheme might be problematic in terms of real-time execution since the inference phase takes too much time due to the extraction of region proposals. Fast R-CNN (Girshick, 2015) and SPP-Net (He et al., 2015) improve the R-CNN and reduce the inference time via extracting RoIs from feature maps. They, nevertheless, still use fixed algorithms and cannot learn how to extract RoIs. The Faster R-CNN (Ren et al., 2015) enables be trained end-to-end by replacing the selective search with a region proposal network (RPN) which learns how to extract RoIs. RPN allows the model to learn how to generate candidate regions proposals and reduces the inference time dramatically. Mask R-CNN (He et al., 2017) adds a mask prediction branch on Faster R-CNN to detect objects and it predicts their masks simultaneously. The R-FCNN (Dai et al., 2016) introduces position-sensitive score maps to enhance object detection quality.

One-stage approaches, in contrast, remove the RoI extraction. Instead, they regress and classify candidate anchor boxes. The SSD (i.e. Single-Shot Detection) (Liu et al., 2016) extracts feature maps from anchor boxes by employing several small convolutional filters classifying the bounding boxes and assigning confidence scores. DSSD (Fu et al., 2017) adds a deconvolution path to SSD, yielding an improvement in the detection of small objects. The Corner-Net (Law & Deng, 2018) is a key-point-based approach that can detect objects using corner points. Center-net (Duan et al., 2019), another key-point-based approach, utilizes a center point in addition to a corner point to capture visual patterns and eliminate the mispredicted bounding boxes. The well-known algorithm YOLO (Redmon & Farhadi, 2017) divides the image into $S \times S$ grids and predicts the bounding boxes. Each bounding box generates the corresponding vector containing coordinate points, width, height, and confidence score. It leverages the intersection of union ND non-maximum-suppression between ground truth and predicted bounding box to eliminate redundant bounding boxes. There are several vanilla versions and extensions of YOLO in the literature. The recent one, YOLO8 introduces self-attention and features pyramids to improve detection quality.

Although these models show promising performance in object detection, they are evaluated on datasets having only clean images. However, images and videos in real-world scenarios may contain several natural artifacts such as noise, motion blur, and rain due to various factors such as atmospheric scattering (Li et al., 2017) or faults in image sensors. Degraded images can significantly reduce the accuracy of object detection models, especially for small objects. As an illustration, noise can significantly impact the performance of object detection algorithms, leading to false positives or missed detections.

In this study, we contribute by specifically investigating and evaluating the robustness of small OD tasks against those artifacts in the absence and presence of image restoration. To achieve this, we utilized several state-of-the-art YOLO variants to perform the detection of small objects on clean and degraded images. We first trained the three single-shot OD algorithms (YOLO 6/7/8) with clear images of the original VisDrone dataset (Cao et al., 2021). Second, we derived three degraded test sets from the VisDrone test set adding synthetic noise, motion blur, and rain. Followingly, we employed Multi Stage Progressive Image Restoration (MPRNet) (Rajaei et al., 2023) on degraded test sets to obtain their restored counterparts. Finally, we evaluated the OD models on (a) clean, (b) degraded, and (c) enhanced versions. The results show that OD models trained with clean images cannot generalize well and their performance diminishes on degraded images whereas leveraging MPRNet-based image enhancement significantly improves the model detection quality on degraded images.

The rest of this paper is organized as follows. We first introduce the employed deep architectures in the section of Material and Methods. Later, we provide details of the data in the section of Datasets. Followingly, the metrics we utilized are given in the section of Evaluation Metrics. The section of Results and Discussion present the results of experiments and provides some discussions on the findings. The last section concludes the paper.

## MATERIALS AND METHODS

### You Only Look Once (YOLO)

YOLO is an end-to-end fashioned real-time object detection algorithm that can perform object detection with a single pass of the network. It divides the image into $S \times S$ grids with equal shapes. Each cell is responsible for detecting objects that appear within them and predicts bounding boxes represented by a feature vector containing center points coordinates, width, height, and confidence score that indicates how model confidence on whether that box contains an object and how accurate the predicted box is. Then, the model computes the intersection over union (IoU) between predicted and ground-truth bounding boxes to eliminate those which cannot pass the threshold value. Since the algorithm may predict multiple bounding boxes that pass the threshold for the same object, non-maximum suppression (NMS) is applied to identify redundant boxes and output one box for each object in the image. YOLO OD series are often used in real-time use cases covering on the edge inference.

### YoloV6

YOLOv6 (Li et al., 2022) is a cutting-edge object detector that provides a balance between speed and accuracy. It introduces notable enhancements to its architecture including a reparameterized backbone, Path Aggregation Network (PAN) (Liu et al., 2018), and efficient decoupled head for prediction.

The reparameterized network enhances the detection quality and speeds up the inference phase. The network architecture is switched during training and inference to balance speed and accuracy. It uses a simple network during inference for efficiency whereas a complex one is preferable in training to provide higher accuracy. Path aggregation network concatenates features from different reparameterized blocks hence it is called as RepPan. Compared to the previous Yolo version, Yolov6 uses efficient decoupled heads to split classification and detection paths. This approach reduces the computational complexity and provides higher accuracy.

**YoloV7**

YOLOv7 (Wang et al., 2023) proposes many novelties in its architecture to increase efficiency such as extended Efficient Layer Aggregation Network (E-ELAN) (Wang et al., 2022) and Compound Model Scaling. The E-ELAN is employed as the computational block for the YOLOv7 backbone architecture. It uses expand, shuffle, and merge cardinality to continuously improve the network's capacity for learning while preserving the original gradient path. Scaling helps the model to comply with the needs of objective tasks in terms of speed and accuracy. The authors of YOLOv7 optimized the network architecture search technique (NAS) and proposed a compound model scaling approach that scales the width and height in coherence for concatenation-based models.

Furthermore, YOLOv7 contains two trainable bags of freebies named planned re-parameterized convolution (RepConvN) and Coarse for auxiliary and fine for lead loss (CAFL). RepConv combines 3x3 convolution, 1x1 convolution, and identity connection in one convolution layer. The RepConvN is RepConv without an identity connection. This technique increases the training time yet provides higher accuracy in prediction. By utilizing the CAFL approach, Yolov7 overcomes the limitation of a single head. It contains a lead head responsible for predicting output, and an auxiliary head that assists training in the middle layers.

**YoloV8**

YOLOv8, as the recent version, improves over previous ones in terms of speed and accuracy. It enriches the detection quality by utilizing self-attention, feature pyramids, and mosaic augmentation.

The enhanced CSPDarkNet53 builds the backbone of the YOLOv8 containing 53 convolutional layers and leverages cross-state partial connections to provide communication between different layers. The head consists of convolutional layers followed by a series of fully-connected layers. It predicts bounding boxes, confidence scores, and class probabilities for the detected objects. The self-attention forces the model to focus on different features based on their relevance to the task. It is noteworthy that the feature pyramid network allows the model to perform multi-scale object detection. It contains multiple layers that can detect objects at different scales. Another key enhancement in YOLOv8 is the mosaic augmentation which helps the object detection models to learn how to detect objects in cluttered or complex scenes yielding better generalization in the *wild*. By utilizing mosaic augmentation, the model is exposed to a wider variety of visual contexts and can learn to recognize objects in a more robust and generalizable way. It should be noted that YOLOv8 is an anchor free approach reducing the number of bounding box predictions and boosts the NMS.

**Multi stage progressive image restoration (MPRNet)**

MPRNet (Zamir et al., 2021; Rajaei et al., 2023) is a multi-stage model for image restoration. Since it is multi-stage, it breaks down the restoration process of the degraded image into sub-tasks and progressively learns the restoration function and restores the degraded image followingly. The model first learns the contextualized information using an encoder-decoder network and then combines them with a high-resolution subnetwork that retains the local information. At each stage, the supervised attention module re-weights the local features and exchanges information between different stages. In order to avoid loss of information, cross-state feature fusion is leveraged at each state to establish the connection between feature processing blocks.

At any given state $S$, the model predicts the residual image $R_S$ and adds the degraded input image $I$ to obtain the predicted restored image: $X_S = I + R_S$. The model is optimized end-to-end with following loss function:

$$\mathcal{L} = \sum_{S=1}^{3}[\mathcal{L}_{char}(X_S, Y) + \lambda\mathcal{L}_{Edge}(X_s, Y)] \quad (1)$$

where $Y$ represents the ground-truth image, and $\mathcal{L}_{Char}$ is the Charbonnier loss:

$$\mathcal{L}_{Char} = \sqrt{\|X_S - Y\|_2 + \epsilon_2} \quad (2)$$

with constant $\epsilon$ empirically set to $10^{-3}$. In addition, $\mathcal{L}_{Edge}$ is the edge loss defined as:

$$\mathcal{L}_{Edge} = \sqrt{\|\Delta(X_S) - \Delta(Y)\|_2 + \epsilon_2} \quad (3)$$

where $\Delta$ denotes the Laplacian operator. The parameter $\lambda$ controls the importance of the two loss term.

The key parts of the MPRNet are encoder-decoder sub-network, original resolution sub-network, cross-state feature fusion, and supervised attention module. The encoder-decoder architecture allows the model to focus on more relevant features at each stage. The cross-state feature fusion module is employed to make the model less vulnerable the information loss due to the encoder-decoder subnetwork. The supervised attention module generates the feature map that filters less informative features and only allows useful ones to propagate to the next stage. Finally, the original resolution network is employed at the last stage to generate spatially-encriched, high-resolution images.

## Datasets

We conducted the experiments on VisDrone dataset (Cao et al., 2021). This dataset covering 10 object classes, contains 6471 images for training and 1580 images for testing. We built separate "noisy", "blurry" and "rainy" test sets derived from the original VisDrone test set by adding synthetic noise, motion blur, and rain effects. In the following, we applied the pre-trained MPRNet on degraded test sets to obtain their restored versions called "noisy-clear", "blur-clear", and "rain-clear". Fig. 1 represents our pipeline for generating the degraded and their restored test sets. In the stage of image restoration, though our experimentation setup involves one 16 GB VRAM equipped 3080TI GPU, we have experienced insufficient memory issues which forces us to decrease the resolution of the original images in test sets in a way to have 640x640 pixels. The summaries of the datasets used in this experiment are listed in Table 1.

**Table 1.** Summary of the Original And Derived Datasets

| Dataset | Size | Description |
|---|---|---|
| Original Train set | 6471 images | Novel VisDrone train set |
| Original Test set | 1580 images | Novel VisDrone test set |
| Noisy test set | 1580 images | Derived from VisDrone test set by adding noise |
| Blurry test set | 1580 images | Derived from VisDrone test set by adding motion blur |
| Rainy test set | 1580 images | Derived from VisDrone test set by adding synthetic rain |
| Noise-clear test set | 1580 images | Derived from Noisy test set by performing denoising (MPRNet) |
| Blur-clear test set | 1580 images | Derived from Blurry test set by performing deblurring (MPRNet) |
| Rain-clear test set | 1580 images | Derived from Rainy test set by performing deraining (MPRNet) |

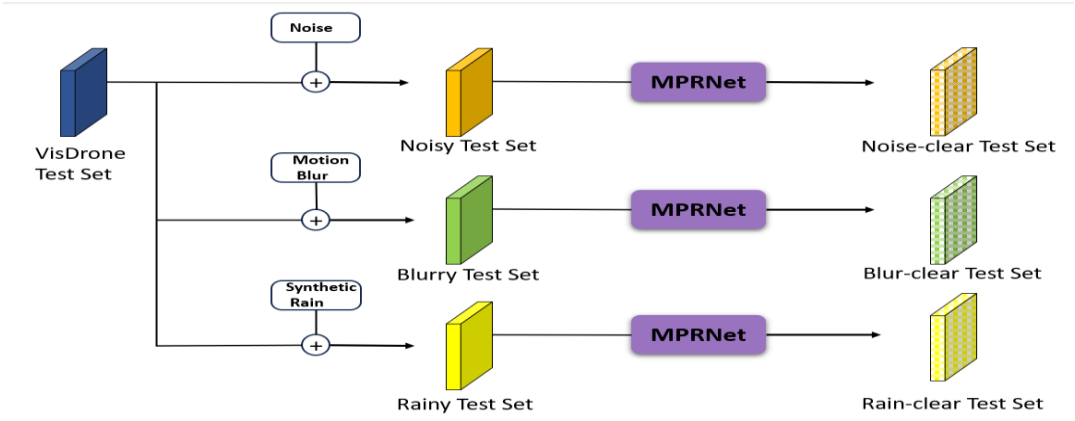


**Figure 1.** Test Set Generation Pipeline

### Evaluation metrics

In this experiment, the metrics of precision, recall, f1-score and mean average precision (mAP) were used to evaluate the accuracy of YOLO models we utilized. The following equations shows how the performance metrics are computed.

$$Precision = \frac{True\ Positive}{True\ Positive+False\ Positive} \quad (4)$$

$$Recall = \frac{True\ Positive}{True\ Positive+False\ Negative} \quad (5)$$

$$F1 = \frac{2\ x\ Precision\ x\ Recall}{Precision+Recall} \quad (6)$$

$$Recall = \frac{True\ Positive}{True\ Positive+False\ Negative} \quad (7)$$

$$mAP = \frac{1}{n}\sum_{1}^{n} AP_k \quad (8)$$

where $AP = \sum_{0}^{n}[Recall(i) - Recall(i+1)]\ x\ Precision(i)$ and $n$, and $i$ denote the number of classes and treshold respectively.

## RESULTS AND DISCUSSION

We fine-tuned *tiny*, *small*, and *medium* variations of each YOLO version for 50 epochs using an NVIDIA RTX 3060 GPU equipped with 6 GB VRAM. We set the batch size and learning rate to 2 and 0.01 respectively. We evaluated each model by using test sets listed in Table 1 and reported precision, recall, f1, and mAP scores. We used degraded and restored test sets to investigate how image restoration improves the accuracy of the detection model in case the input image is degraded. The Table 2, 3, and 4 report the experiment results for YOLO v6, v7, and v8 respectively.

**Table 2.** Yolov6 Test Results

| Dataset | Model | Precision | Recall | F1 Score | mAP |
|---|---|---|---|---|---|
| **Original test set** | Nano | 0.281 | 0.23 | 0.226 | 0.18 |
| | Small | 0.382 | 0.304 | 0.311 | 0.151 |
| | Medium | 0.449 | 0.36 | 0.378 | 0.322 |
| **Noisy test set** | Nano | 0.113 | 0.088 | 0.091 | 0.038 |
| | Small | 0.146 | 0.119 | 0.118 | 0.059 |
| | Medium | 0.178 | 0.127 | 0.132 | 0.068 |
| **Blurry test set** | Nano | 0.127 | 0.097 | 0.1 | 0.046 |
| | Small | 0.127 | 0.109 | 0.112 | 0.052 |
| | Medium | 0.144 | 0.093 | 0.104 | 0.050 |
| **Rainy test set** | Nano | 0.169 | 0.147 | 0.141 | 0.084 |
| | Small | 0.246 | 0.202 | 0.204 | 0.138 |
| | Medium | 0.295 | 0.221 | 0.232 | 0.165 |
| **Noise-clear test set** | Nano | 0.14 | 0.101 | 0.111 | 0.056 |
| | Small | 0.157 | 0.118 | 0.127 | 0.066 |
| | Medium | 0.178 | 0.137 | 0.146 | 0.081 |
| **Blur-clear test set** | Nano | 0.238 | 0.178 | 0.18 | 0.123 |
| | Small | 0.256 | 0.224 | 0.22 | 0.158 |
| | Medium | 0.289 | 0.216 | 0.233 | 0.165 |
| **Rain-clear test set** | Nano | 0.167 | 0.146 | 0.141 | 0.082 |
| | Small | 0.245 | 0.194 | 0.2 | 0.133 |
| | Medium | 0.364 | 0.253 | 0.277 | 0.213 |

The experiments show that the detection performance of YOLO models is significantly reduced with degraded images. Performing the MPRNet-based image enhancement on degraded images yields promising improvement in object detection performance in line with our hypothesis. For the YOLOv6-tiny model, the denoising operation results in a %47 improvement in the mAP score. It also increases the mAP scores for YOLOv7-small and YOLOv8-medium models by 116%, and 60% respectively.

**Table 3.** YOLOv7 Test Results

| Dataset | Model | Precision | Recall | F1 Score | mAP |
|---|---|---|---|---|---|
| **Original test set** | Nano | 0.358 | 0.292 | 0.322 | 0.248 |
| | Small | 0.526 | 0.416 | 0.465 | 0.397 |
| | Medium | 0.511 | 0.441 | 0.473 | 0.406 |
| **Noisy test set** | Nano | 0.133 | 0.071 | 0.092 | 0.025 |
| | Small | 0.139 | 0.075 | 0.097 | 0.030 |
| | Medium | 0.154 | 0.069 | 0.095 | 0.032 |
| **Blurry test set** | Nano | 0.102 | 0.093 | 0.097 | 0.044 |
| | Small | 0.127 | 0.085 | 0.102 | 0.048 |
| | Medium | 0.124 | 0.093 | 0.106 | 0.045 |
| **Rainy test set** | Nano | 0.181 | 0.188 | 0.184 | 0.107 |
| | Small | 0.322 | 0.248 | 0.28 | 0.187 |
| | Medium | 0.315 | 0.249 | 0.278 | 0.185 |
| **Noise-clear test set** | Nano | 0.137 | 0.097 | 0.114 | 0.05 |
| | Small | 0.182 | 0.108 | 0.136 | 0.065 |
| | Medium | 0.165 | 0.112 | 0.133 | 0.062 |
| **Blur-clear test set** | Nano | 0.215 | 0.214 | 0.214 | 0.138 |
| | Small | 0.317 | 0.249 | 0.279 | 0.187 |
| | Medium | 0.325 | 0.242 | 0.277 | 0.188 |
| **Rain-clear test set** | Nano | 0.191 | 0.176 | 0.183 | 0.106 |
| | Small | 0.302 | 0.247 | 0.272 | 0.18 |
| | Medium | 0.296 | 0251 | 0.272 | 0.178 |

Similarly, leveraging the deblurring process has shown a better performance boost than we expected on the blurred test set. It improves the mAP scores %230, 289%, and 263% for YOLOv6-medium, YOLOv7-medium, and YOLOv8-tiny respectively. Although MPRNet-based image enhancement improves the detection results for noisy and blurry images, it cannot maintain the same performance on rainy images due to the algorithm we used to add synthetic rain into images. The algorithm unfortunately generated poor-quality, non-realistic synthetic rain hence MPRNet cannot recognize and remove it well from the image.

**Table 4.** YOLOv8 Test Results

| Dataset | Model | Precision | Recall | F1 Score | mAP |
|---|---|---|---|---|---|
| **Original test set** | Tiny | 0.388 | 0.291 | 0.332 | 0.266 |
| | Small | 0.453 | 0.342 | 0.389 | 0.326 |
| | Medium | 0489 | 0.37 | 0.421 | 0.359 |
| **Noisy test set** | Tiny | 0.122 | 0.055 | 0.075 | 0.042 |
| | Small | 0.156 | 0.058 | 0.084 | 0.047 |
| | Medium | 0.196 | 0.066 | 0.099 | 0.061 |
| **Blurry test set** | Tiny | 0.122 | 0.054 | 0.075 | 0.041 |
| | Small | 0.139 | 0.058 | 0.081 | 0.049 |
| | Medium | 0.184 | 0.061 | 0.091 | 0.056 |
| **Rainy test set** | Tiny | 0.215 | 0.16 | 0.183 | 0.113 |
| | Small | 0.248 | 0.203 | 0.223 | 0.15 |
| | Medium | 0.278 | 0.219 | 0.245 | 0.171 |
| **Noise-clear test set** | Tiny | 0.172 | 0.073 | 0.102 | 0.060 |
| | Small | 0.206 | 0.075 | 0.11 | 0.075 |
| | Medium | 0.231 | 0.087 | 0.126 | 0.084 |
| **Blur-clear test set** | Tiny | 0.263 | 0.183 | 0.216 | 0.149 |
| | Small | 0.298 | 0.203 | 0.241 | 0.176 |
| | Medium | 0.323 | 0.218 | 0.26 | 0.192 |
| **Rain-clear test set** | Tiny | 0.217 | 0.16 | 0.184 | 0.116 |
| | Small | 0.251 | 0.198 | 0.221 | 0.15 |
| | Medium | 0.277 | 0.21 | 0.239 | 0..166 |

This is likely related to the irrelevant distribution between the real-world raindrops and our synthetic rain effect. Furthermore, one might question why the restoration process could not catch the

mAP performance that we gained from the novel test set. The unwanted image size reduction caused significant information loss, and reduces the model performance, especially for small objects. Our comprehensive visual inspection clearly revealed that extremely small objects (e.g., <10x10 pixels) in the original test set became impossible to detect in degraded and restored image sets upon the image resize phase.

Further, as expected, we also observed that the newer YOLO models performed better compared to their previous versions regardless of the applied image degradation. It should be noted that except for the YOLOv8 models, the model sizes (i.e., tiny/small) affected the OD performance when it comes to de-rained images. Heavier the model we applied, the better the mAP values we gained. The performance gain obtained with the use of YOLO8 models sources from several advancements such as (1) a decoupled head performing objectness, classification and regression tasks individually, (2) the introduction of anchor-free OD paradigm, and (3) improved mosaic based data augmentation techniques which incorporate MixUp and CutMix (Terven, Córdova-Esparza and Romero-González., 2023). Moreover, as Wang et al. (2023) pointed out, to reveal multi-scale feature maps, input images are processed through several convolution and C2f modules in YOLO8 saving the lightweight characteristics along with capturing more abundant gradient flow. The C2f module is mainly used for residual learning and is reported to be an enhanced version of ELAN structure presented by YOLO7 (Wang et al., 2023). Another key contribution of YOLO8 is the merging of Feature Pyramid Network (FPN) and Pyramid Attention Network (PAN) paradigms which enables blending high and low level features through semantic and localization related features making the model better utilizing varying scaled features resulting in improved detection performance when small and large objects come into prominence. From the perspective of listed improvements and our problem domain, it is not a surprise to obtain surpassing scores with the use of YOLO8 in our experiments since most of the objects in our dataset are significantly small.

Overall, our results prove that image enhancement significantly improves the detection quality of the YOLO models on degraded images. It also improves small object detection on aerial images. Due to space constraints, we could not share any run-time analysis of MPRNet.

## CONCLUSION

In this work, we hypothesized that small object detection tasks, especially in aerial images, may be improved by employing image restoration networks when it comes to degraded images. To achieve this, we applied several YOLO (6/7/8) models on the (a) original, (b) synthetically degraded, and (c) restored versions of the test portion of the VisDrone dataset. Our evaluation using both clean and degraded sets demonstrate how degraded images reduce the detection quality. To eliminate these artifacts from the image, we performed MPRNet-based image enhancement on the degraded test set and evaluate the YOLO models on these restored test sets. The results show that image enhancement significantly improves the detection quality, regardless of the underlying YOLO model, especially for small objects on degraded images. In future work, we aim to couple more image enhancement approaches with other OD models and analyze the run-time performance of these models in large and small image format regimes.

### Conflict of Interest

The article authors declare that there is no conflict of interest between them**.**

### Author's Contributions

The authors declare that they have contributed equally to the article**.**

## REFERENCES

Cao, Y., He, Z., Wang, L., Wang, W., Yuan, Y., Zhang, D., & Liu, M. (2021). VisDrone-DET2021: The vision meets drone object detection challenge results. In *Proceedings of the IEEE/CVF International conference on computer vision* (pp. 2847-2854).

Dai, J., Li, Y., He, K., & Sun, J. (2016). R-fcn: Object detection via region-based fully convolutional networks. *Advances in neural information processing systems, 29*.

Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., & Tian, Q. (2019). Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 6569-6578).

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 580-587).

Girshick, R. (2015). Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision* (pp. 1440-1448).

Fu, C. Y., Liu, W., Ranga, A., Tyagi, A., & Berg, A. C. (2017). Dssd: Deconvolutional single shot detector. *arXiv preprint arXiv:1701.06659*.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence, 37*(9), 1904-1916.

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision* (pp. 2961-2969).

Law, H., & Deng, J. (2018). Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 734-750).

Li, B., Peng, X., Wang, Z., Xu, J., & Feng, D. (2017). Aod-net: All-in-one dehazing network. In *Proceedings of the IEEE international conference on computer vision* (pp. 4770-4778).

Li, C., Li, L., Jiang, H., Weng, K., Geng, Y., Li, L., & Wei, X. (2022). YOLOv6: A single-stage object detection framework for industrial applications. *arXiv preprint arXiv:2209.02976*.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14* (pp. 21-37). Springer International Publishing.

Liu, S., Qi, L., Qin, H., Shi, J., & Jia, J. (2018). Path aggregation network for instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 8759-8768).

Rajaei, B., Rajaei, S., & Damavandi, H. (2023). An Analysis of Multi-stage Progressive Image Restoration Network (MPRNet). *Image Processing On Line, 13*, 140-152.

Redmon, J., & Farhadi, A. (2017). YOLO9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7263-7271).

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems, 28*.

Terven, J., Córdova-Esparza, D. M., & Romero-González, J. A. (2023). A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS. *Machine Learning and Knowledge Extraction, 5*(4) (pp.1680-1716).

Uijlings, J. R., Van De Sande, K. E., Gevers, T., & Smeulders, A. W. (2013). Selective search for object recognition. *International journal of computer vision*, *104*, 154-171.

Wang, C. Y., Bochkovskiy, A., & Liao, H. Y. M. (2023). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 7464-7475).

Wang, X., Gao, H., Jia, Z., & Li, Z. (2023). BL-YOLOv8: An Improved Road Defect Detection Model Based on YOLOv8. *Sensors*, *23*(20), 8361.

Wang, C. Y., Liao, H. Y. M., & Yeh, I. H. (2022). Designing Network Design Strategies Through Gradient Path Analysis. *arXiv preprint arXiv:2211.04800*.

Zamir, S. W., Arora, A., Khan, S., Hayat, M., Khan, F. S., Yang, M. H., & Shao, L. (2021). Multi-stage progressive image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 14821-14831).