Analyzing YOLO Models for Enhanced Small Object Detection in Indoor Environments

Teng-Hui Tan2, a), Shahbe Mat-Desa1, 2, b) and Wan-Noorshahida Mohd-Isa 1, 2, c)

1Centre for Image and Vision Computing, CoE for Artificial Intelligence, Multimedia University, Persiaran Multimedia, 63100 Cyberjaya, Malaysia

2Faculty of Computing and Informatics, Multimedia University, Persiaran Multimedia, 63100 Cyberjaya, Malaysia

b) Corresponding author: shahbe@mmu.edu.my

a) 1211102289@student.mmu.edu.my  
c) wan.noorshahida.isa@mmu.edu.my

**Abstract.** Indoor small object detection presents several difficulties, primarily due to low resolution and reduced detection accuracy. Despite significant advancements in the field, many approaches continue to face challenges. A key challenge in achieving real-time performance is minimizing computational costs while maintaining an acceptably high level of accuracy. In YOLO-based methods, activation functions play a crucial role in optimizing both computational efficiency and detection accuracy. This paper evaluates several activation functions: SiLU, ReLU, ELU, LeakyReLU, and Sigmoid, and assesses their impact on the performance of YOLOv8 and YOLOv11 models in indoor environments. A custom dataset, consisting of small objects like Lego pieces, dice, and cubes, has been created and annotated with bounding boxes. In the experiments, YOLOv8 with the ELU activation function achieves the highest mAP50-95 score of 0.801. The experimental results demonstrate that selecting a suitable activation function for YOLO can lead to a significant improvement in detection accuracy.

# Introduction

Object detection is a core task in computer vision, essential to a wide range of applications including robotics, autonomous driving, industrial monitoring, medical imaging, video surveillance, and UAV tracking. Traditional approaches rely on handcrafted features and classical classifiers, which are often computationally intensive and prone to lower performance. The emergence of deep learning, especially convolutional neural networks (CNNs), has transformed the field by enabling the development of faster and more accurate detection algorithms such as YOLO [1], Faster R-CNN [2], SSD [3], and RetinaNet [4].

Despite substantial progress in detecting medium and large objects, the detection of small and extremely small-scale objects remains a significant challenge, primarily due to low contrast and limited sensor resolution. These challenges are further increased by occlusion, cluttered backgrounds, and a lack of sufficient contextual information, which make detection even more difficult. To overcome the difficulties, researchers have investigated specialized techniques such as super-resolution, multi-scale processing, attention mechanisms, and the integration of depth information and multi-camera perspectives. YOLO [1], recognized as a one-stage object detection framework, has gained significant attention for its fast processing and practical utility. It treats detection as a regression problem by dividing the image into grids, allowing simultaneous prediction of bounding boxes and classifications. While YOLO excels in real-time applications, its performance in detecting small objects is occasionally outperformed by two-stage models like Faster R-CNN. Each approach offers distinct advantages: one-stage models prioritize speed and efficiency, while two-stage models often achieve higher accuracy with small objects. As the need for accurate small object detection grows, there is increasing emphasis on advancing algorithmic design.

This study attempts to bridge the gap between the general-purpose datasets and the custom dataset for small object detection. It also highlights how activation functions can improve detection performance.

# Related Work

A new backbone and an anchor-free head were proposed in YOLOv8 [5] for achieving better performance on the VisDrone dataset [6] with a mean average precision (mAP) of 45.9%, precision of 76.8%, and outperformed the existing models for detecting extremely small objects. Vasanthi et al. [5] propose an improved YOLOv8 backbone incorporating additional bottleneck layers for effective dimensionality reduction of feature maps while preserving their representational strength. The framework employs an anchor-free head to directly predict object center points and dimensions, eliminating the need for predefined anchor boxes. Training is further optimized using mix-up and mosaic data augmentation techniques on the VisDrone dataset, which contributes to improved detection accuracy.

Moreover, [7] presents DS-YOLO for the detection of dense and small objects in cluttered surveillance scenes. DS-YOLO overcomes the object detection challenges due to occlusion and scale change among the tightly crowded objects by introducing improvements over the YOLOv8 framework. To strengthen feature extraction, the model uses a redesigned backbone network to use shallow-layer object features without increasing the computational load. Key modules, including DO-C2f and C2fUIB, are integrated to further improve the efficiency of feature extraction with low parameter requirements. DS-YOLO also introduces the LFS-PAFPN module, which enables better aggregation of detailed high-resolution and positional low-resolution features. This improves the missed detection rate and general accuracy since it covers a larger range of object information. In [8], a novel Composite Dilated Convolution and Attention Module (CDAM) is proposed to capture rich contextual information through dilated convolutions at varying rates while reducing noise with a CBAM attention mechanism. To address imbalances caused by medium and large objects in shallow feature layers, the Feature Elimination Module (FEM) is implemented to suppress irrelevant features and emphasize small object features. Furthermore, a composite backbone network combines two separate backbone architectures to provide robust and diversified feature extraction using their complementary strengths.

Most current research predominantly relies on general-purpose datasets such as MS COCO [9] and PASCAL VOC, which lack optimizations for small object detection in crowded indoor environments. Despite extensive work on architectural advancements in object detection frameworks like YOLO, there has been limited investigation into the role of activation functions in improving the detection performance of small objects.

# Methodology

## Problem Analysis

To bridge research gaps, two important contributions have been proposed in this project:

* A specific dataset with small items drawn from an indoor environment has been developed in a high-fidelity and challenging environment with variable scales of objects, changing lights, and occlusions.
* An evaluation of several activation functions, including ReLU, LeakyReLU, SiLU, ELU, and Sigmoid, was conducted using YOLOv8, along with a comparative analysis against YOLOv11 [10].

## Data Preparation

A custom dataset comprising ten object classes including dice, Lego, cube, hexagon, cylinder, battery, screw, eraser, marble, and paperclip, was created using images sourced from Roboflow. All images were annotated with bounding boxes to ensure accurate representation of the target objects. A sample of annotated images is shown in Figure 1. To improve model robustness and detection performance under varying conditions, data preprocessing and augmentation techniques were applied, including random rotations, horizontal flips, and 90-degree rotations. The final dataset, consisting of 3,219 images, is divided into training, validation, and test sets using a 70:20:10 split ratio.



**FIGURE 1**. Sample annotated small object in images

## Transfer Learning with YOLO

Transfer learning was used to take advantage of the knowledge gained when a model is trained on another dataset. The pre-trained YOLOv8 model was fine-tuned on the MS COCO [9] combined with our custom dataset. First, the default training setting of the model was used, i.e., employing the default SiLU activation function. Further sets of experiments were conducted using YOLOv8 to observe the change in performance due to the variation of different activation functions: ReLU, ELU, Leaky ReLU, and Sigmoid. Another set of experiments is also conducted to measure the difference in performance between YOLOv8 and YOLOv11. In contrast to YOLOv8, YOLOv11 was taken in its default architecture and its default activation function, without any changes. The mathematical expressions of the activation functions are given in Equations (1)- (5) [11].

## Testing and Evaluation

These YOLOv8 and YOLOv11 models and their experimental variants were each tested on the same metrics, including mAP, precision, recall, and F1-score. The results were analyzed to compare the effect of applying different activation functions in YOLOv8, and the performance comparison of YOLOv8 and YOLOv11 on small object detection. The evaluation metrics are given in Equations (6) – (9) [5].

# Results and Discussion

This section presents a comparison of the YOLOv8 performance when trained with various activation functions (SiLU, ReLU, LeakyReLU, ELU, and Sigmoid), and evaluated at two different epoch stages: 20 epochs and 50 epochs. All models were trained under the same training settings, including batch size of 16, image size of 640×640, and AdamW as an optimizer. The results in Table 1 show that the performance obtained using 50 epochs is better compared to the 20 epochs. Amongst all functions tested using 50 epochs, ELU scores the highest overall mAP50-95 of 0.801, along with having the best recall (0.990). This is closely followed by ReLU with mAP50-95 of 0.799. SiLU, YOLOv8's default activation, also performs consistently well at 50 epochs, with precision, recall, mAP50 and mAP50-95 are 0.992, 0.988, 0.990 and 0.793, respectively. In contrast, Sigmoid underperforms with the lowest score for all measurements, i.e., it is not optimal for this use case.

Using similar training parameters as for YOLOv8, YOLOv11 was also tested with its default activation function, SiLU. The results in Table 2 show that at 50 epochs, there are small improvements in YOLOv11's performance as opposed to YOLOv8, with a precision of 0.990, a recall of 0.983, an mAP50 of 0.989, and an mAP50-95 of 0.799. Furthermore, performance for different classes is consistently high, with an mAP50-95 for objects such as hexagon and paperclip performing above 0.85. YOLOv11 shows similar or marginally improved performance compared to the mAP50-95. At 20 epochs, YOLOv8 achieved an mAP50-95 of 0.793, while YOLOv11 scored an mAP50-95 of 0.791, showing an almost negligible difference. At the 50 epochs, YOLOv11 exceeds YOLOv8 with an mAP50-95 rate of 0.799, compared to a rate of 0.793 for YOLOv8. While both models realized optimal or nearly optimal results for most object categories, YOLOv11 achieves improved consistent localization accuracy for tougher categories like hexagon and eraser, which achieved mAP50-95 rates of 0.901 and 0.868, respectively.

**TABLE** **1.** Validation results of all classes using YOLOv8 with different activation functions

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Activation function | Images | Instances | Measurement metrics at 20 epochs (above) and 50 epochs (below) | | | |
| Precision | Recall (R) | mAP50 | mAP50-95 |
| SiLU  (default) | 379 | 661 | 0.990 | 0.984 | 0.991 | 0.793 |
| **0.992** | 0.988 | 0.990 | 0.793 |
| ReLU | 379 | 661 | 0.983 | 0.981 | 0.987 | 0.781 |
| 0.991 | 0.984 | 0.988 | 0.799 |
| LeakyReLU | 379 | 661 | 0.979 | 0.980 | 0.983 | 0.778 |
| 0.986 | 0.986 | **0.991** | 0.795 |
| ELU | 379 | 661 | 0.986 | 0.988 | 0.988 | 0.784 |
| 0.990 | **0.990** | 0.989 | **0.801** |
| Sigmoid | 379 | 661 | 0.943 | 0.944 | 0.962 | 0.702 |
| 0.972 | 0.982 | 0.979 | 0.735 |

**TABLE 2.** Validation results of YOLOv11 model

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Class | Images | Instances | Measurement metrics at 20 epochs (above) and 50 epochs (below) | | | |
| Precision | Recall (R) | mAP50 | mAP50-95 |
| all | 379 | 661 | 0.988 | 0.982 | 0.987 | 0.791 |
| 0.990 | 0.983 | 0.989 | 0.799 |
| battery | 30 | 40 | 0.965 | 0.825 | 0.920 | 0.682 |
| 0.943 | 0.832 | 0.934 | 0.680 |
| cube | 40 | 40 | 0.970 | 1.000 | 0.995 | 0.824 |
| 0.999 | 1.000 | 0.995 | 0.830 |
| cylinder | 30 | 30 | 0.992 | 1.000 | 0.995 | 0.858 |
| 0.993 | 1.000 | 0.995 | 0.868 |
| dice | 40 | 312 | 0.994 | 0.998 | 0.991 | 0.636 |
| 0.994 | 0.999 | 0.992 | 0.637 |
| eraser | 40 | 40 | 0.985 | 1.000 | 0.995 | 0.875 |
| 0.988 | 1.000 | 0.995 | 0.868 |
| hexagon | 40 | 40 | 0.995 | 1.000 | 0.995 | 0.890 |
| 0.994 | 1.000 | 0.995 | 0.901 |
| lego | 40 | 40 | 0.994 | 1.000 | 0.995 | 0.718 |
| 0.994 | 1.000 | 0.995 | 0.739 |
| marble | 40 | 40 | 0.994 | 1.000 | 0.995 | 0.761 |
| 0.993 | 1.000 | 0.995 | 0.765 |
| paperclip | 40 | 40 | 0.995 | 1.000 | 0.995 | 0.855 |
| 0.996 | 1.000 | 0.995 | 0.878 |
| screw | 39 | 39 | 0.995  0.995 | 1.000  1.000 | 0.995  0.995 | 0.809  0.823 |

While the performance gap between YOLOv11 and YOLOv8 is small, YOLOv11 shows a slight edge in bounding box precision (mAP50-95) as training progresses, especially at 50 epochs. However, YOLOv8 with the ELU activation function achieved the highest mAP50-95 score of 0.801 at 50 epochs. This suggests that ELU offers a marginal advantage over the default SiLU under certain conditions, reinforcing the importance of activation function selection for accurate small object detection in complex indoor settings. Furthermore, YOLOv8 with the ELU activation function, trained for 50 epochs, emerged as the best model with a mAP50-95 score of 0.801 and consistently high precision and recall across all object classes. As shown in Figure 2, the F1-Confidence curve highlights the model’s performance across confidence thresholds. At a threshold of 0.648, it achieved an optimal F1 score of 0.99, reflecting a strong balance between precision and recall. This indicates the model’s effectiveness in reducing false positives while reliably detecting small indoor objects.

A precision of 1.00 was achieved at a confidence threshold of 0.931, indicating that the model produced no false positives above this threshold. At a threshold of 0.000, the model maintained a high recall of 0.99 across all classes, showing strong sensitivity in detecting nearly all true positives regardless of confidence. The precision-recall curve at an IoU threshold of 0.5 yielded an AP of 0.989, reflecting the model's effectiveness in minimizing false positives while detecting most objects. A strong precision-recall balance at a 0.5 confidence threshold further supports the model’s high detection accuracy across object classes.

A collage of graphs

Description automatically generated

**FIGURE 2**. Visualization of F1-confidence, precision-confidence, precision-recall, and recall-confidence curves for YOLOv8

A graph of loss and loss results

AI-generated content may be incorrect. A screenshot of a computer

AI-generated content may be incorrect.

**FIGURE** **3**. (a) Loss of boxes, objects, and classes in YOLOv8, (b) YOLOv8 confusion matrix

As depicted in Figure 3(a), training box loss started at a value just above 1.45 and continually dropped across epochs to eventually approach a value of just around 1.00 at the end of training. The downward trend indicates that the model achieved improved prediction of bounding boxes with increasing accuracy. On the other hand, validation box loss started at around 1.20 and gradually dropped below 0.90, reflecting consistent convergence as well as proper generalization during validation. Classification training loss shows a clear upward trend. It was first noted at around 2.6 and saw a dramatic dip to around 0.50, which indicates an increase in the model's confidence in class assignment predictions. The validation classification loss started at around 1.0, although with certain fluctuations, indicates a steady downward trend to finally arrive at a figure below 0.40. With regards to distribution focal loss (DFL), we witnessed a training DFL loss reduction from around 1.27 to below 1.10, indicating improved effectiveness for bounding box regression. The validation DFL loss, likewise, saw a decrease from around 1.21 to 1.03, mirroring a trend consistent with that of the training curve. All three types of loss show generally desirable downward trends with minimal difference between training and validation, evidencing successful training of the YOLOv8 with ELU activation, with no signs of overfitting.

The YOLOv8’s confusion matrix in Figure 3(b) shows that the model performing at optimality achieved a classification rate of 1.00 for most object classes. The model performed well with an accuracy rate of 0.90 for the classes of batteries, misclassifying only 7% of instances of batteries as background on certain instances. On the other hand, the confusion rate was found to be negligible within the hexagon class since only 2% of instances were misclassified to other categories. The background class was dealt with effectively most of the time, but confusion occurred at 29% relative to the hexagon class and at 71% with respect to the battery class. Overall, strong class discrimination was realized by the model, depicted through low misclassification rates, hence proving its effectiveness for detecting small indoor objects.

# Conclusion

This paper presents a preliminary investigation into the impact of activation functions on small object detection in indoor environments using YOLOv8 and YOLOv11. It aims at improving detection performance for low-resolution, densely packed objects. Preliminary results indicate that YOLOv8 with the ELU activation function achieved the highest mAP50-95 score of 0.801. YOLOv11 also showed consistent and reliable performance. These findings underscore the significance of activation function selection and provide a useful baseline for further advancements in small object detection.

# References

1. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” (Jun. 2015), arXiv:1506.02640.
2. R. Girshick, “Fast R-CNN,” in *2015 IEEE International Conference on Computer Vision (ICCV)*, (IEEE, Santiago, Chile, 2015), pp. 1440–1448.
3. W. Liu et al., “SSD: Single Shot MultiBox Detector,” in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. (Springer, Cham, 2016), pp. 21–37.
4. T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, “Focal Loss for Dense Object Detection,” IEEE Transactions on Pattern Analysis and Machine Intelligence **42**, 318–327 (2020).
5. P. Vasanthi and L. Mohan, “Efficient YOLOv8 algorithm for extreme small-scale object detection,” Digital Signal Processing: A Review Journal **154**, 104682 (2024).
6. P. Zhu et al., “Detection and Tracking Meet Drones Challenge,” IEEE Transactions on Pattern Analysis and Machine Intelligence **44**, 1–18 (2022).
7. H. Zhang et al., “DS-YOLO: A dense small object detection algorithm based on inverted bottleneck and multi-scale fusion network,” Biomimetic Intelligence and Robotics **4**, 100190 (2024).
8. X. Jing, X. Liu, and B. Liu, “Composite Backbone Small Object Detection Based on Context and Multi-Scale Information with Attention Mechanism,” Mathematics **12**, 622 (2024).
9. T.-Y. Lin et al., “Microsoft COCO: Common Objects in Context,” (May 2014), *arXiv preprint* arXiv:1405.0312.
10. R. Khanam and M. Hussain, “YOLOv11: An Overview of the Key Architectural Enhancements,” (2024), arXiv:2410.17725.
11. M. Erkuş and A. Çınar, “The Effect of the Improved YOLOv8 Model with Activation Functions on the Detection of Blood Cells and Urine Particles,” (Nov. 16, 2024), *Authorea Preprints*.