Object Detection in Indoor Environments Using YOLOv11n: Feature Extraction with ResNet18 and Grad-CAM++, Feature Engineering with ROI, HOG, LBP, and PCA for Feature Selection

Nafiz Fahad1, a), Md. Jakir Hossen2, b) and Md. Shohel Sayeed3, c)

*1Faculty of Information Science and Technology, Multimedia University, Jalan Ayer Keroh Lama, 75450, Bukit Beruang, Melaka, Malaysia.*

*2Center for Advanced Analytics (CAA), COE for Artificial Intelligence, Faculty of Engineering & Technology (FET), Multimedia University, 75450 Melaka, Malaysia.*

*3Centre for Intelligent Cloud Computing, COE of Advanced Cloud, Faculty of Information Science and Technology, Multimedia University, Jalan Ayer Keroh Lama, 75450 Bukit Beruang, Melaka, Malaysia*

*b) Corresponding author:* [*jakir.hossen@mmu.edu.my*](mailto:jakir.hossen@mmu.edu.my)  *a) fahadnafiz1@gmail.com*

*c) shohel.sayeed@mmu.edu.my*

# Abstract. Object detection is important in computer vision because it finds and locates objects in images. This study uses the YOLOv11n model to detect objects in indoor settings, working with a dataset of 2,213 images across seven object classes. The dataset contains different backgrounds, lighting conditions, and occlusion, which make detection challenging. To boost model performance, the images were resized and augmented. ResNet18 extracted features, and Grad-CAM++ showed the areas that guided the model’s decisions. Feature-engineering methods such as Region of Interest (ROI), Histogram of Oriented Gradients (HOG), and Local Binary Patterns (LBP) were also used to improve detection. Principal Component Analysis (PCA) helped choose the most useful features. After training, YOLOv11n achieved strong results: precision was 0.920 for training, 0.900 for validation, and 0.885 for testing. Recall was 0.901 for training, 0.877 for validation, and 0.802 for testing. The mean Average Precision at 50% Intersection over Union (mAP50) reached 0.948 for training, 0.921 for validation, and 0.896 for testing, showing good detection accuracy under varied conditions.

# Introduction

Object detection is a key task in computer vision. Its goal is to find and mark objects from certain categories in images [1,2,3,4]. Deep Convolutional Neural Networks (DCNNs) have driven major progress here, achieving strong results in image and video recognition, face recognition, edge detection, motion detection, optical character recognition, and machine vision [5,6,7]. These networks learn useful image features on their own through convolution, pooling, and fully connected layers. By applying DCNNs to object detection, machines now understand and interpret visual data far better in fields such as security, transportation, defense, and healthcare [8,9].

Moreover, A recent study compares YOLO-ELWNet with several lightweight models using the KITTI dataset. YOLO-ELWNet achieves a mAP of 85.08%, FPS of 48.45, with 23.87 million parameters and 20.01 GFLOPs. In comparison, MobileNetV3 has a mAP of 77.44%, FPS of 72.98, and 23.18 million parameters. MobileNeXt achieves a mAP of 78.24%, FPS of 62.09, and 22.73 million parameters. GhostNet shows a mAP of 78.84%, FPS of 56.22, and 22.25 million parameters. EfficientNetV2 has a mAP of 78.58%, FPS of 57.72, and 31.85 million parameters. They’re used YOLO-ELWNet achieves 85.08% accuracy whereas their study has still limitations which is why they did not achieve benchmark results [10]. A different recent study introduce LSOD-YOLO, a lightweight small object detection model based on YOLOv8, which significantly improves detection performance while reducing computational cost. The model is compared with other versions like YOLOv8s and YOLOv8s-P2 on the VisDrone2019 dataset. LSOD-YOLO achieves an mAP0.5 of 37.0%, Precision of 48.4%, and recall of 38.2%, with a parameter count of just 3.8M and a model size of 7.6MB. In comparison, YOLOv8s and YOLOv8s-P2 have mAP0.5 scores of 34.5% and 36.9%, respectively, with larger model sizes and higher computational demands. Additionally, LSOD-YOLO outperforms models like SSD and YOLOv5 in accuracy, while maintaining high real-time processing speed (93 FPS) and efficiency, making it suitable for resource-constrained applications. Their study’s result is not benchmarked, and they did not apply any feature engineering techniques [11]. Moreover, In the year of 2025 one solution proposes the YOLO-SSP model for object detection in remote sensing images. The model is tested against various other models on the DIOR, TGRS-HRRSD, and SIMD datasets. Results show that YOLO-SSP achieves a mAP of 64.7%, surpassing other models like YOLOv8m (62.4%) and YOLOv7 (62.5%). YOLO-SSP has 28.3M parameters and processes at 72.3 FPS. The model incorporates improvements such as a lightweight SPD-Conv module, a small object detection layer, and a pyramid spatial attention mechanism (PYSAM), enhancing detection accuracy, especially for small objects in complex backgrounds. However, their result is not benchmark and they did not use any feature selection technique [12]. One different study, In the year of 2025 proposes the YOLO-LDFE model for underwater object detection, which outperforms several other models. Compared to YOLOv8, which achieves an mAP@0.5 of 88.9% with 3.0M parameters and 8.1 GFLOPs, YOLO-LDFE achieves a higher mAP@0.5 of 93% with 4.7M parameters and 18.2 GFLOPs. YOLO-LDFE enhances detection accuracy by 4.1% while maintaining efficient performance, making it well-suited for real-time applications in resource-constrained environments [13]. Their study also did not achieve benchmark results.

Nevertheless, object detection has advanced greatly through deep learning models such as YOLO. However, even with these advances, current models still struggle to balance accuracy, speed, and real-time processing, especially on devices with limited power. For example, A solution which is compared YOLO-ELWNet with other lightweight models on the KITTI dataset and reached a good mAP of 85.08 percent, but their results still fell short of benchmark goals because of limits in their method [10]. Likewise, another solution which introduced LSOD-YOLO for small-object detection. Although the model was accurate and fast, their study did not reach benchmark targets and did not include feature-engineering steps [11]. One different solution developed YOLO-SSP, which achieved a mean average precision of 64.7 percent on remote-sensing images, but the study omitted benchmarking and feature-selection methods [12]. Later, another different solution which released YOLO-LDFE for underwater object detection, raising mean average precision to 93 percent, yet it still fell short of benchmark requirements [13]. These works highlight the ongoing challenge of improving object detection, especially in real-world scenes with varied object sizes, cluttered backgrounds, and limited computing power.

Therefore, this study aims to strengthen object detection for indoor scenes with YOLOv11n. We expand dataset diversity through rotation, flipping, and brightness adjustment. To understand model decisions, we apply Grad-CAM++ for visualization and use feature-extraction techniques such as Histogram of Oriented Gradients and Local Binary Patterns. We also employ Principal Component Analysis to reduce bounding-box features and improve accuracy. The goal is to create a more robust and efficient detection system for real-time use in indoor spaces with limited resources.

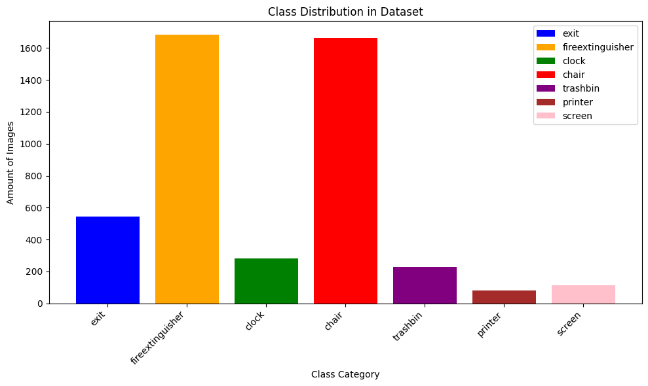
# METHOD

## Problem Definition

This study aims to improve object detection in indoor settings using YOLOv11n. The training dataset has 2,213 images covering seven object classes, with challenges such as different backgrounds, lighting conditions, and occlusions. To make the model stronger, it applies data-augmentation methods like rotation, flipping, and brightness changes. The study also examines ways to see and explain the model’s decisions through Grad-CAM++ and feature-extraction methods such as HOG and LBP. By lowering the model’s complexity and raising its accuracy, the goal is to build a system that can detect objects in real time and work well even on devices with limited resources.

## Dataset Collection

This study used a fully labeled object detection dataset collected from indoor scenes. This indoor dataset consists of 2213 image frames containing seven classes. In contrast to existing indoor datasets, our dataset includes a variety of background, lighting conditions, occlusion and high inter-class differences. Moreover, figure 1 shows the class distribution in dataset and figure 2 shows some images of the dataset [14]

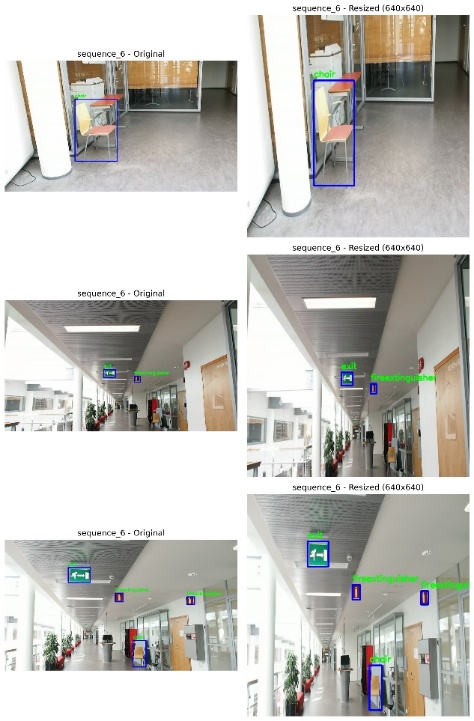


**FIGURE 1.** Class distribution in dataset



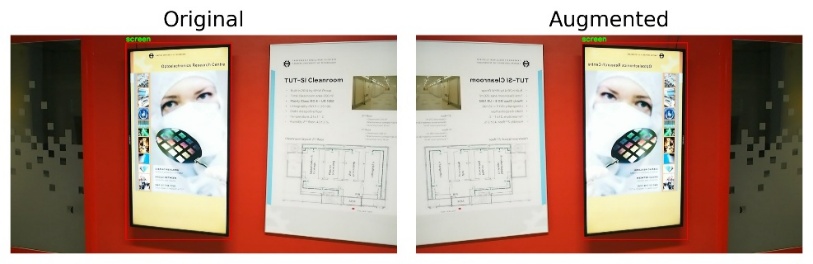
**FIGURE 2.** Some images from the used dataset (Tampere University of Technology (TUT) indoor dataset) for object detection with annotated bounding boxes and labels

Moreover, firstly parsing XML annotation files to extract bounding boxes and associated labels for each image. The images are resized to a target resolution of 640x640, with corresponding adjustments made to the bounding box coordinates to preserve object positions. After that, draw bounding boxes function is used to overlay bounding boxes and their respective labels on the images, while the viusalize image function displays the original and resized images in a 3x2 grid. Additionally, this process resized the six image sequence files, selected three images per sequence, and visualized the bounding box annotations. However, figure 3 shows original images and resized images of a sequence file.



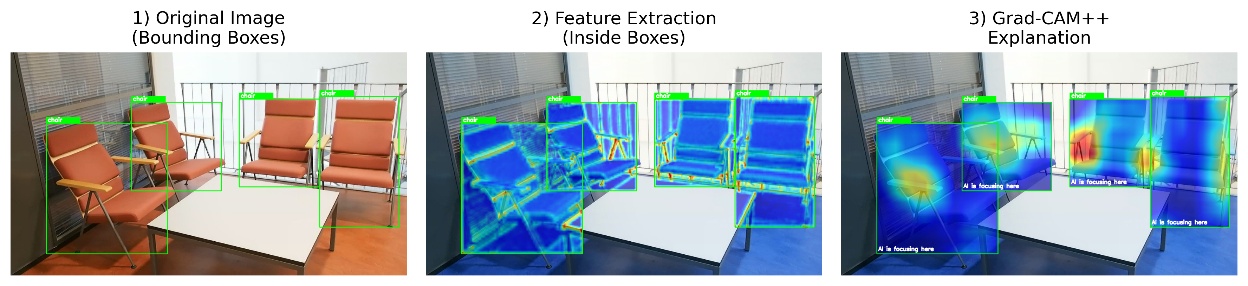
**FIGURE 3.** Original image and resized images

Moreover, the augmentation process involves applying several transformations such as rotation, flipping, brightness adjustment, and Gaussian blur. These augmentations are designed to increase the variability of the dataset, improving the model's ability to detect objects under different orientations and conditions. By augmenting the dataset, the model can generalize better, becoming more robust to real-world variations. However, the rotate image function is used to rotate the image and adjusts the bounding boxes, while the flip image function horizontally flips the image. The adjust brightness and apply gaussian blur functions enhance image diversity. New annotations for augmented images are saved as XML files, and the augmented images are visualized side by side for comparison. For example, the original and augmented images are shown side by side in Figure 4, where the augmented image has been altered to demonstrate how these transformations improve image diversity.



**FIGURE 4.** Original image and augmented image

Figure 5 visualizes how a pre-trained ResNet-18 model processes images by focusing on bounding box regions. It initializes the model and applies Grad-CAM++ to highlight important regions, captures intermediate feature maps to understand learned representations, and visualizes which areas within the bounding boxes contribute most to the model's predictions. The code also analyzes cropped regions from bounding boxes to show the model's focus, annotates images with bounding boxes and labels, and optionally highlights the areas the model focuses on. It combines the original image, feature extraction heatmaps, and Grad-CAM++ explanations in a side-by-side visualization, helping to interpret the model's decision-making process. XML annotations are read to obtain bounding-box coordinates and labels for processing, and the main processing loop produces visual explanations for the dataset images. The shown image illustrates the feature extraction and Grad-CAM++ visualizations, marking key regions and helping to explain how the model handles the image.



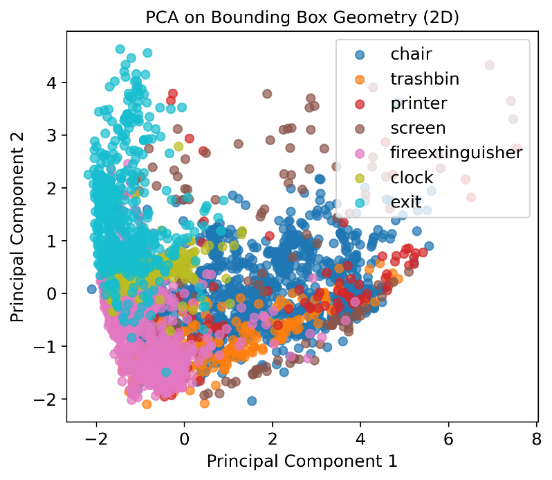
**FIGURE 5.** Original image with feature extraction and Grad-CAM++ Visualization

This study uses classical feature-engineering methods, including Histogram of Oriented Gradients (HOG) and Local Binary Patterns (LBP). It extracts bounding boxes and labels from annotations in XML files and applies the HOG and LBP methods to each region of interest (ROI) in the images. The first part of the process draws bounding boxes around detected objects such as “exit” signs. Next, HOG and LBP features are extracted from the cropped regions inside these bounding boxes, and the results are visualized. The HOG visualization shows the gradient features, while the LBP image shows the texture patterns. This side-by-side visualization helps in understanding how different features of the images are represented and processed by the model, giving clarity into how the model interprets these regions. Figure 6 shows the original image with bounding boxes around the exit signs and the visual output of the HOG and LBP methods applied to one of the detected regions. HOG visualization highlights the gradient structure of the region, while the LBP image shows the texture details. These visualizations help to better understand the decision-making process of the object-detection system and its focus on specific features within the bounding boxes.

This study uses Principal Component Analysis (PCA) to reduce the dimensionality of the bounding box features from an object detection dataset. The features selected are the width, height, area, and aspect ratio of the bounding boxes. These features are extracted from XML annotations and processed for each image in the dataset. After scaling, PCA reduces these features to two components. The explained variance ratio for the two components is [0.72261, 0.25659], meaning the first component explains most of the variance. The results are displayed in a 2D scatter plot, with each point representing a bounding box and colored by its class label. Figure 7 shows this PCA visualization of bounding box geometry. The plot reveals how bounding boxes from different classes (such as exit, chair, and fire extinguisher) are spread across the 2D space formed by the first two principal components. This helps to understand how the bounding box shapes differ between object classes, with clear separation in the plot. Moreover this current study use yolov11n model and visualize the results.

|  |
| --- |
|  |
|  |

**FIGURE 6.** Original image, ROI, HOG visualization and LBP image



**FIGURE 7.** Shows PCA visualization of bounding box geometry

# Result

For this study we trained in the yolov11n model and Figures 8 present key results from the study on object detection performance. Figure 8(a) shows the final precision values for the train, validation, and test sets, with values of 0.920, 0.900, and 0.885, respectively. Figure 8(b) illustrates recall metrics for the same sets, with values of 0.901 for training, 0.877 for validation, and 0.802 for testing. Figure 8(c) displays the mean Average Precision at IoU threshold 50 (mAP50), with values of 0.948 for training, 0.921 for validation, and 0.896 for testing. Figure 8(d) shows the mAP50-95 values, with training, validation, and test values of 0.701, 0.696, and 0.664, respectively. Figure 8(e) presents the final fitness values, with training, validation, and test scores of 0.726, 0.718, and 0.687. Finally, Figure 8(f) displays random test images with YOLOv11n detections, highlighting the model’s effectiveness in detecting objects such as exit signs and fire extinguishers. These figures provide a clear overview of the model's performance across different metrics

|  |  |  |
| --- | --- | --- |
| (a) | (b) | (c) |
| (d) | (e) | (f) |

**FIGURE 8.** Th key results of yolov11n model.

# CONCLUSION

This study clearly shows the use of the YOLOv11n model for object detection in indoor settings, using a dataset that contains challenging variations in background, lighting, and occlusion. Applying data-augmentation methods and resizing the images increased the model’s robustness. Feature extraction with ResNet18, together with Grad-CAM++ for visualizing decision areas, provided useful insight into how the model identifies key features. Including feature-engineering methods such as ROI, HOG, and LBP further improved the model’s detection capability. PCA was used effectively for feature selection, reducing dimensionality and enhancing performance. The YOLOv11n model achieved strong results, with precision values of 0.920 for training, 0.900 for validation, and 0.885 for testing, and recall values of 0.901, 0.877, and 0.802, respectively. In addition, the mAP50 scores showed solid detection accuracy at 0.948 for training, 0.921 for validation, and 0.896 for testing. These findings confirm that YOLOv11n is a powerful and efficient tool for object detection in indoor environments, even with limited computational resources.

# Acknowledgments

The authors want to thank Multimedia University, Malaysia.

# References

1. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen, “Deep learning for generic object detection: A survey,” International *Journal of Computer Vision* **128**, 261 (2019).
2. Y. Amit, P. Felzenszwalb, and R. Girshick, “Object detection,” In *Computer vision: A reference guide* (pp. 875-883). Cham: Springer International Publishing.
3. A. K. Gupta, A. Seal, M. Prasad, and P. Khanna, "Salient object detection techniques in computer vision—A survey," *Entropy*, *22*(10), 1174.
4. X. Zhang, Y. H. Yang, Z. Han, H. Wang, and C. Gao, "Object class detection: A survey,” *ACM Computing Surveys (CSUR)*, *46*(1), 1-53.
5. A. A. Tulbure, A. A. Tulbure, and E. H. Dulf, "A review on modern defect detection models using DCNNs–Deep convolutional neural networks," *Journal of Advanced Research*, *35*, 33-48.
6. M. Z. Alom, M. Hasan, C. Yakopcic, T. M. Taha, and V. K. Asari, "Inception recurrent convolutional neural network for object recognition," *Machine Vision and Applications*, *32*, 1-14.
7. R. Kaur and S. Singh, "A comprehensive review of object detection with deep learning*," Digital Signal Processing*, *132*, 103812.
8. Y. Sun, Z. Sun, and W. Chen, "The evolution of object detection methods,” *Engineering Applications of Artificial Intelligence*, *133*, 108458.
9. B. Song, J. Chen, W. Liu, J. Fang, Y. Xue, and X. Liu, "YOLO-ELWNet: A lightweight object detection network," *Neurocomputing*, 129904.
10. H. Wang, J. Liu, J. Zhao, J. Zhang, and D. Zhao, "Precision and speed: LSOD-YOLO for lightweight small object detection," *Expert Systems with Applications*, *269*, 126440.
11. Y. Liu, D. Yang, T. Song, Y. Ye, and X. Zhang, "YOLO-SSP: an object detection model based on pyramid spatial attention and improved downsampling strategy for remote sensing images," *The Visual Computer*, *41*(3), 1467-1484.
12. J. Liu, R. Zhou, Y. Li, and P. Ren, "Enhanced underwater object detection with YOLO-LDFE: a model for improved accuracy with balanced efficiency," *Journal of Real-Time Image Processing*, *22*(2), 58.
13. B. Adhikari, J. Peltomaki, J. Puura, and H. Huttunen, "Faster bounding box annotation for object detection in indoor scenes," In *2018 7th European Workshop on Visual Information Processing (EUVIP)* (pp. 1-6). IEEE.