A Study on Image Resolution and Object Scale Adjustment for Efficient Object Detection in Mobile Network Environments

*Pill-Won Park**

Software-Centered University Project Group,
Hanyang University ERICA,
Republic of Korea
pillwon79@hanyang.ac.kr

Abstract

Object detection is one of the most fundamental and core research areas in the field of computer vision, and the YOLO Series, a representative model series, is widely utilized across various artificial intelligence systems. Mobile networks serve as a crucial connectivity element that links nearly all industrial sectors, connecting various IoT devices through these networks. A typical example is network cameras (CCTV). Some deep learning AI models often exhibit degraded object detection performance compared to their reported benchmark results. While multiple factors may contribute to this, one well-known reason is the difference in characteristics between publicly available training datasets and images collected in CCTV environments. Due to the inherent bandwidth limitations of wireless networks, data transmission is often constrained. Particularly in mobile network environments, various approaches such as applying edge computing have been researched to reduce network load for object detection models deployed on CCTV systems. In this study, we systematically and linearly adjusted image resolution and object scale in video/image data transmitted over the network to analyze their impact on detection performance. Through this, the goal is to explore practical methods for achieving efficient real-time object detection that consider the constraints of network environments.

Keywords: Image resolution, Object proportion within images, Deep learning, Mobile network, Edge computing

1 Introduction

Early forms of artificial intelligence, such as expert systems, relied on explicitly programmed knowledge and rules. These systems generated responses to user queries solely based on pre-entered data and rules, making them incapable of reasoning beyond their programmed knowledge. To address this limitation, researchers developed techniques enabling computers to autonomously generate rules from data, a concept now known as machine learning [1-2].

Machine learning is utilized in almost every domain where artificial intelligence is applied, with computer vision being one of its most active application areas. Early computer vision methods primarily focused on identifying features based on pixel relationships and differences within images or objects. Representative algorithms include SIFT (Scale-Invariant Feature Transform), which extracts keypoints across multiple scales using Difference of Gaussian (DoG) operations, and Haar feature-based methods, which leverage contrast in image regions. However, these traditional approaches often failed to deliver satisfactory performance, especially when confronted with complex image features or significant variations such as changes in illumination, distortion, and noise. Furthermore, low-level approaches like SIFT and Haar focused primarily on edges, corners, and intensity, making it difficult to capture high-level semantics such as object identity or contextual information [3-4].

The introduction of artificial neural networks, inspired by human cognition, addressed many of these challenges and ultimately gave rise to modern deep learning. Artificial neural networks are composed of input layers, output layers, and multiple hidden layers, each containing numerous nodes. As input data propagates through the network, each node generates intermediate responses by applying weighted signals and bias, ultimately producing an output prediction based on the collective inference of all layers.

Prominent deep learning models in computer vision include AlexNet, YOLO, SSD, and Faster R-CNN. AlexNet, a convolutional neural network (CNN) designed for image classification, demonstrated the effectiveness of deep learning approaches by winning the ILSVRC 2012 competition. YOLO (You Only Look Once) represents a real-time single-stage object detection model that simultaneously predicts object positions and categories in a single pass, making it particularly well suited for applications requiring rapid inference. SSD (Single Shot MultiBox Detector) utilizes multiple feature maps to detect objects of varying sizes, while Faster R-CNN introduces a region proposal network (RPN) to efficiently generate candidate object regions for two-stage detection. Among these, YOLO has become one of the most widely adopted CNN-based object detection models, with ongoing improvements through to its twelfth version. In

*Corresponding Author: Pill-Won Park; Email: pillwon79@hanyang.ac.kr
DOI: https://doi.org/10.70003/160792642025112606011

practice, YOLO is also available under licenses that permit commercial use [5-8].

Deep learning models typically consist of numerous layers and nodes, each with a large number of parameters. Although the mathematical framework is well defined, it remains challenging to fully interpret how each specific input influences final predictions. As a result, model parameters are not usually disclosed in detail, but pre-trained models—trained on curated public datasets—are commonly shared via SOTA leaderboards or repositories such as GitHub.

Pre-trained models are generally unable to distinguish between unseen object classes. For example, a YOLO model trained on the COCO dataset can distinguish between "person" and "airplane," but cannot differentiate between "person" and "soldier," or between "airplane" and "drone," since such fine-grained classes are not part of its training data. Thus, for real-world applications targeting specific objects, additional training—namely fine-tuning—on relevant data, as illustrated in Figure 1, is required [9].

Figure 1. Results of analyzing the same image data with different label policies

When fine-tuned models are evaluated using public datasets, their reported performance can be competitive with SOTA. However, such results often do not generalize well to images collected in real-world environments. As shown in Table 1, this issue is widespread in the field of computer vision, affecting not only object detection but also tasks such as denoising, deblurring, and image enhancement under adverse conditions [10-11].

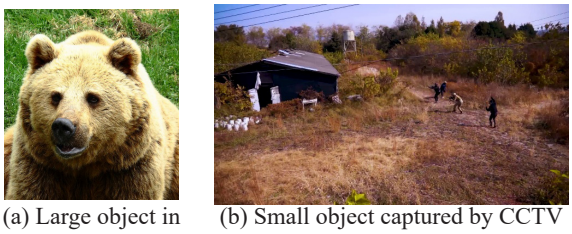
Table 1. PSNR values from SOTA and from real-world data

	PSNR	SSIM
Public Datasets & gUnet (SOTA)	41.34~33.52	0.996~0.971
Real-world collec -ted data & gUnet	38.86~34.48	0.99~0.978

Performance gaps between benchmarks and practice are largely attributable to differences in characteristics between public training datasets and real-world data. While it is practically impossible to account for every possible scenario in computer vision, this issue can be narrowed down to two main factors in the context of object detection.

First, there is the difference in image resolution. For example, the widely used COCO dataset primarily consists of images with a resolution of 640×480 pixels, whereas modern CCTV systems typically generate high-resolution images of 1920×1080 pixels or higher. Previous studies have shown that object detection performance improves with higher-resolution images. Therefore, whenever possible, it is necessary to perform object detection using high-resolution images [12-13].

Second, there is a difference in object size within images. While COCO 2017 images typically contain a single large object, CCTV images are designed to cover wide areas, resulting in much smaller objects relative to the overall scene. Figure 2(a) illustrates an example of a large object in an image from the COCO dataset, whereas Figure 2(b) shows an example of a small object captured by CCTV. Previous studies have reported that object size is inversely proportional to detection accuracy. This is because larger objects contain more information, which increases the likelihood that a deep learning model will make accurate inferences about them [14-18].



(a) Large object in an COCO image (b) Small object captured by CCTV

Figure 2. Example of the Relative object scale in public datasets.

In the past, dedicated cables connected DVR (Digital Video Recorders) and object detection devices were commonly used. Currently, many systems are configured so that cameras, NVRs (Network Video Recorders), and control centers are connected via networks. Using high-resolution videos or images directly over the network inevitably places a heavy burden on the network. Therefore, it is necessary to identify a resolution that minimizes the impact on object detection performance while enabling downscaling. This study empirically investigates the effects of image resolution and object size on detection accuracy and aims to propose practical guidelines for the effective application of AI models in real-world environments [19-20].

2 Related Works

2.1 Network-Based CCTV Architecture

In the past, it was common to connect DVRs directly to CCTV cameras and link dedicated computers for object detection. However, as network-based systems became widely adopted, CCTV systems evolved into configurations where captured data is managed through the network.

The architecture of early systems using the network model was as follows: data captured by CCTV is stored in an NVR, and downscaled data is transmitted to a central control center where object detection is performed.

Although simple in structure, this approach places a heavy load on the entire network from the CCTV to the central control center and requires high computing power at the central detection system.

To address these drawbacks, edge computing techniques have been proposed. While data captured by CCTV is still stored in an NVR, an edge computer is deployed near the CCTV to perform object detection locally. In the event of an unusual situation, alerts and video/image data are sent to the central control center. Although installing edge devices increases costs, their physical proximity to the CCTV compared to the central processing center enables faster response times. Additionally, selective processing at the edge reduces network load and usage since not all data is sent to the central server, and edge computing also affords resilience to network failures or central server issues [21-25].

Nonetheless, in both scenarios, video or image data travels over the network, raising the critical need for effective traffic management.

2.2 Public Datasets

In deep learning, the availability of large amounts of training data is generally considered beneficial for model performance. Constructing image datasets for deep learning involves substantial manpower for image collection and annotation, which has historically posed challenges. However, the release of various public datasets has significantly facilitated the development of deep learning models.

Representative public datasets include the following:

- MNIST: One of the earliest publicly available datasets, designed for handwritten digit recognition (digits 0 to 9). It contains 70,000 grayscale images of size 28×28 pixels, with 60,000 for training and 10,000 for testing, and has been widely used for evaluating convolutional neural network performance [26].
- ImageNet: A large-scale dataset containing approximately 14 million images across more than 1,000 object classes. It gained prominence following the success of AlexNet in 2012 and has since become a standard benchmark for evaluating computer vision models [27].
- COCO (Common Objects in Context): The most widely used benchmark dataset for object detection research. The 2017 version includes over 330,000 images annotated for 80 everyday object categories. Most images have a resolution of 640×480 pixels.
- Open Images: Developed by Google, this extensive dataset consists of over 9 million images with more than 16 million bounding box annotations.
- Places2: Created by MIT, this dataset provides over 10 million images spanning more than 400 scene categories [28].

Beyond these, specialized datasets have been established for domains such as satellite imagery and medical imaging, including MRI and X-ray datasets.

2.3 CCTV Data Generation and Characteristics of Generated Images

According to market research reports, the demand for 4K and higher resolution CCTV cameras and the adoption of IP-based CCTV systems are rapidly increasing. Modern CCTV systems typically generate video data with resolutions of 1920×1080 pixels or higher. Additionally, studies analyzing real urban CCTV footage have shown that most images possess resolutions of at least 1280×720 pixels.

In contrast, the widely used public dataset COCO predominantly consists of images with a much lower resolution of 640×480 pixels. Furthermore, most public datasets contain objects that occupy a significantly larger proportion of the image compared to real-world CCTV footage. For example, in the COCO 2017 dataset, objects occupy approximately 24% to 41% of the total image area, whereas in actual CCTV footage, this proportion is often below 1%.

Urban CCTV analysis research indicates that small objects—defined as those with a longest side less than 100 pixels—account for approximately 55.3% of all detected objects. Specifically, 66.62% of pedestrians and 41.92% of vehicles fall into this small object category. These findings highlight the predominance of small objects in real CCTV footage, reflecting a notable disparity compared to the relatively large object scale proportions observed in public datasets.

2.4 YOLO Series

YOLO (You Only Look Once), first introduced in 2015, is a prominent family of object detection and image segmentation models. The principal characteristic of YOLO is its single neural network architecture that enables the simultaneous and real-time prediction of the locations and classes of multiple objects within an image. Compared to other CNN-based object detection models such as Faster R-CNN and SSD, YOLO offers substantially faster inference, making it particularly suitable for applications requiring real-time processing, including video surveillance and autonomous driving.

While each version of YOLO has introduced specific architectural innovations, the fundamental pipeline remains largely consistent:

- The input image is divided into an S×S grid.
- Each grid cell predicts B bounding boxes, each with an associated confidence score.
- Bounding box information includes the center coordinates (x, y), width (w), height (h), and confidence (pc).
- Conditional class probabilities are estimated for each bounding box.
- A convolutional backbone extracts image features, which are then processed by the prediction head to simultaneously infer bounding boxes and class labels for each grid cell.
- Bounding boxes with low confidence are discarded, and redundant boxes are suppressed using Non-Maximum Suppression (NMS) to yield the final object locations and classes.

The architecture of the YOLO model is shown in Figure 3.

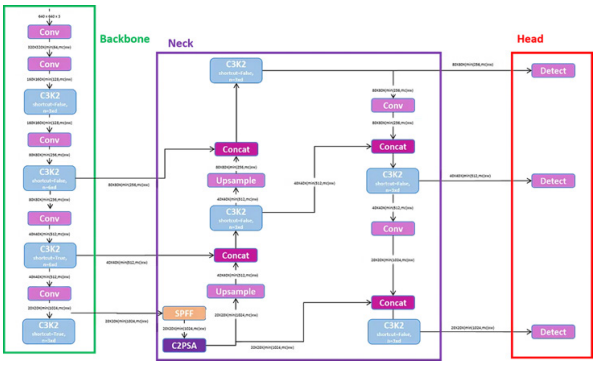


Figure 3. YOLO model architecture

As of now, YOLO has reached version 11, with version 12 under active development. An enterprise license is also available, facilitating widespread adoption in industrial applications.

Distinct features of each YOLO version include:

- YOLOv1: Real-time object detection over the entire image via a single network pass.
- YOLOv2: Introduction of anchor boxes, batch normalization, and multi-scale training, improving accuracy and detection of small objects.
- YOLOv3: Integration of residual blocks and Feature Pyramid Networks, enhancing the detection of objects at various scales.
- YOLOv4: Incorporation of CSPNet, SPP, and other cutting-edge techniques, further improving efficiency and small object detection.
- YOLOv5: Python-based implementation with various model sizes and mosaic data augmentation, optimized for practical deployment.
- YOLOv6: Anchor-free design and optimization for high-performance, lightweight, real-time industrial use.
- YOLOv7: Experimental architectures and efficient model compression improving both accuracy and speed.
- YOLOv8: Expanded functionality including classification, segmentation, pose estimation, and user-friendly APIs.
- YOLOv9: Enhanced gradient-based methods and model compression, optimizing for multi-task support.
- YOLOv10: End-to-end efficient design, removal of NMS, and innovations in both performance and speed.
- YOLOv11: Architectural improvements including C3k2, SPPF, and C2PSA, with support for multi-vision tasks.
- YOLOv12: Adoption of attention-based structures, setting new standards in mAP performance and inference speed.

Typically, YOLO models are trained with input image resolutions of 640×640 pixels but can be scaled to 1280×1280 as needed. However, modern CCTV systems frequently generate images at resolutions of 1920×1080 or higher, necessitating downsampling for model compatibility. This process often results in information loss, potentially impacting detection performance.

2.5 Understanding the Impact of Image Quality and Object Distance on Object Detection Performance

This study highlights that when processing images collected from CCTV systems with various resolutions, downsampling frequently occurs for transmission to remote servers or for faster computation. However, this downsampling process often leads to the loss of pixel information for small objects, resulting in degraded detection performance.

Furthermore, the distance between the objects and the camera also significantly affects detection accuracy. In particular, depth information—representing the distance from the camera to objects within the image—has been shown to improve object detection rates. However, this approach requires that both the training and detection datasets include depth data, necessitating the use of RGB-D datasets (datasets containing both color and depth information). Additionally, input images used for testing or deployment must also contain corresponding depth channels.

Nonetheless, widely used generic image datasets such as ImageNet and COCO do not contain depth information. Although rule-based or AI-based techniques exist to estimate depth from 2D images, these methods typically suffer from lower accuracy, indicating the need for dedicated research to address this limitation. Figure 4 shows an example of an image containing depth information.

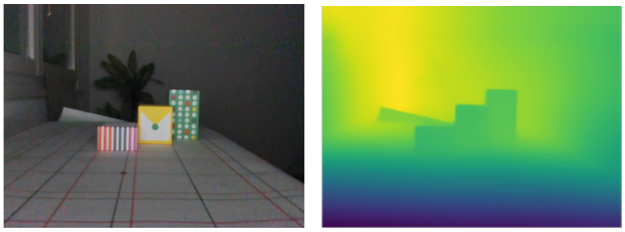


Figure 4. Standard image and RGB-D image

3 Methodology

In this study, we utilized a publicly available human object detection dataset from Roboflow. This dataset was selected over the COCO dataset due to its relatively higher image resolution, making it more suitable for evaluating detection accuracy as image resolution is systematically reduced.

The training set from Roboflow was used to train a YOLO 11-n model, thereby generating a corresponding deep learning model. The YOLO 11-n variant contains approximately 2,600,000 parameters, representing the number of trainable weights within the model. Its computation and resource requirements are quantified as 6.5 billion FLOPs. All other parameters were kept

at their default settings. Training images were resized to 640×640 pixels. This study focused on the effects of image resolution, the accepted pixel size by the model, and object-to-image scale ratio, with minimal modifications to experimental options to facilitate fair comparison.

The technical specifications of the system used for training are as follows:

Operating System: Windows 11

CPU: AMD Ryzen 7 6800H

GPU: NVIDIA GeForce RTX 3070 Ti Laptop GPU

Object Detection Model: YOLO 11-n

Dataset: Roboflow, construction-safety-gsnvb-0h1pm.

class information: helmet: 2,543, no-helmet: 129, no-vest: 892, person: 2,817, vest: 1,343

Dataset Split: Train Set: 83%, Valid Set: 10%, Test Set: 7%

3.1 Resolution Change

For the analysis of image resolution, the original image was designated as 100%, and nine comparative datasets were generated by incrementally decreasing the resolution in 10% intervals. Figure 5 is an image created through original and resolution adjustment.

Figure 5. Sample images: original and resolution-adjusted

In total, ten datasets were constructed by resizing the images from 100% down to 10% of their original resolution. The mean resolution of the original images was 1,056.49 pixels in width and 766.55 pixels in height.

3.2 The Ratio of the Size of the Image to the Size of the Target Object

For the object size ratio experiment, comparative datasets were created as follows:

The original image was designated as (a).

To maintain consistent resolution, a ground truth (gt) image was generated by downscaling the original image to half its width and height, denoted as (c).

As a comparison group, a region corresponding to 50% of the original image's size was extracted, centered on the object, and identified as (b).

This region was cropped from the original image to produce the comparison dataset, referred to as (d).

Using this method, the object size within the image was effectively increased by approximately four times relative to the original.

Figure 6 shows an example of how the proportion of object size within an image changes through resolution adjustment and image cropping, compared to the original image.

Using these groups, we compared the effect of varying object-to-image area ratios on detection performance.

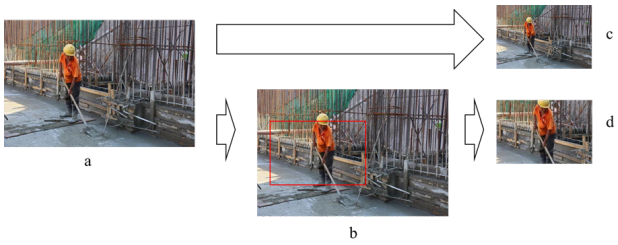


Figure 6. Process for adjusting the object-to-image area ratio

4 Experimental Results

4.1 Performance Variation by Resolution

Figure 7 presents a graph showing the variation in detection performance according to changes in image size. The dataset scaled to 10% of original resolution, which suffered the greatest data loss, exhibited the lowest object detection performance. As data loss decreased—that is, as the resolution of test images increased—object detection performance improved. As shown in Table 2, the detection accuracy remained similar to that of the original image at approximately 60% of the full size (about 596.80 pixels in width and 428.02 pixels in height). This trend is consistent with previous studies indicating that detection performance improves as image resolution increases up to a certain level.

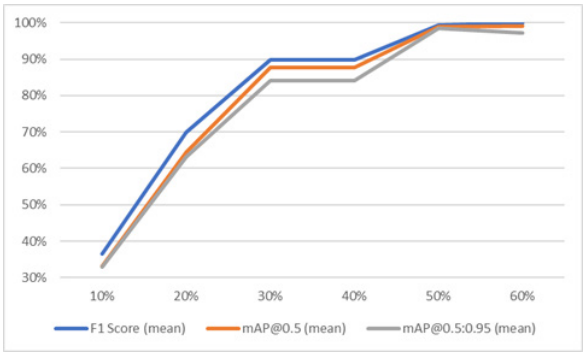


Figure 7. Changes in detection performance according to variations in image size

Table 2. Detailed performance indicators according to ratio changes compared to the reference image

Ratio of original	10%	20%	30%	40%	50%	60%
F1	36.5%	69.9%	89.7%	89.7%	99.3%	99.9%
mAP@0.5	33.1%	64.4%	87.8%	87.8%	98.9%	99.2%
mAP@0.5:0.95	32.9%	63.2%	84.1%	84.1%	98.4%	97.2%
Precision	71.4%	84.2%	100%	100%	99.9%	96.4%
Recall	25.3%	59.9%	82.4%	82.4%	98.2%	100%

However, the lack of further improvement beyond the 60% threshold likely has a different explanation. The resized image dimensions (approximately 596.80×428.02 pixels) closely match the default input size for YOLO model training (640×640 pixels). Therefore, without further adjustment, this resolution emerges as optimal for

the model. Previous studies that addressed resolution issues often included steps to modify the training resolution of the models, supporting the relevance of this approach in similar contexts.

Synthesizing these findings, both the resolution of images used in training and testing, and the input resolution accepted by the detection model, are principal factors impacting model performance.

4.2 Performance Variation with Changes in Object Size Ratio

Assessing the effects of object-to-image size ratio adjustment on detection performance presents several challenges.

First, cropping and enlarging only the object from the image essentially tests the model on a fragment of the original image, resulting in a loss of contextual information. Second, cropping inherently reduces the overall image resolution, which, as highlighted in Section 4.1, can lead to degraded detection accuracy. Third, when objects within the original images are already sufficiently large for effective recognition by the deep learning model, further enlargement may not offer meaningful improvements.

Following the procedure described in the process diagram (formerly Figure 6), the object-to-image ratio within each sample was increased fourfold. Figure 8 illustrates the comparative performance in object detection as the object size ratio is varied.

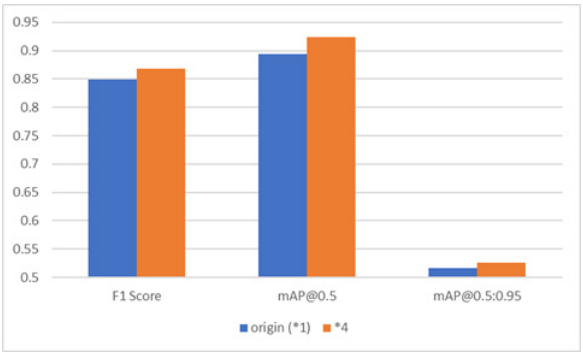


Figure 8. Performance comparison by object size ratio

Table 3. Detailed performance metrics of object detection for original and enlarged object images

	F1	mAP@0.5	mAP@0.5:0.95
Original	0.8486	0.8939	0.5164
Enlarged	0.8686	0.9239	0.5264

Table 3 compares the results between the dataset with a fourfold increase in object size ratio and the original dataset. It was observed that increasing the object ratio slightly improved detection performance, although the effect was relatively minor. This is likely because the objects in the original dataset were already large enough to be accurately detected, making further enlargement unnecessary.

Additionally, previous studies have demonstrated that depth information can significantly influence detection accuracy. Therefore, simply modifying the object size ratio may be insufficient for substantial improvements. Incorporating depth or other auxiliary data may be necessary to achieve meaningful gains in detection performance.

5 Discussion

When conducting experiments on object detection rates according to image resolution, consideration must be given to the fact that AI models commonly downscale input images during training. Therefore, research into the image sizes actually used by object detection models for both training and inference should precede experiments concerning detection rates across differing image resolutions, as this would lead to more effective and reliable results.

There are also several challenges in experiments involving the adjustment of object-to-image size ratios. If objects within an image are already sufficiently large for accurate identification by the model, increasing their size further will have limited benefit. Furthermore, for small objects in pixel-based images, simple enlargement does not necessarily enhance the quality of feature information obtained from the original image. As most CCTV systems produce pixel-based images, alternative research approaches are necessary. Thus, rather than relying solely on image scaling or geometric transformations, generating datasets using cameras with varying focal lengths or zoom capabilities would enable more precise investigation of how object-to-image size ratio affects detection performance.

6 Conclusion

This study restructured the Roboflow dataset to analyze the effects of image resolution and object-to-image scale ratio on object detection performance. The results demonstrated that both the resolution of the images and the resolution of data used by deep learning models during training and testing have a significant impact on detection accuracy, while the proportion of objects within an image plays a comparatively minor role.

When performing object detection using AI models, it can be inferred that the resolution at which a model is trained and tested represents its optimal operating resolution. Therefore, analyzing high-resolution images effectively requires adjusting the input resolution settings during both training and inference phases.

Regarding object size ratio, simple transformations such as image scaling, as used in this study, are insufficient. Instead, it is necessary to construct dedicated datasets containing the same objects captured at multiple distances to properly investigate the influence of object scale on detection performance.

Consequently, when employing models such as YOLO, it is recommended to predefine the input image resolution used for training and testing, and to utilize images matching this resolution for optimal results.

References

[1] V. A. Kshirsagar, S.-C. Lo, G. Lee, Exploring Techniques for Abnormal Event Detection in Video Surveillance, *Journal of Internet Technology*, Vol. 25, No. 5, pp. 781–787, September, 2024. https://doi.org/10.70003/160792642024092505013

[2] S. Sun, W. Ren, T. Wang, X. Cao, Rethinking Image Restoration for Object Detection, *Advances in Neural Information Processing Systems (NeurIPS)*, New Orleans, LA, USA, 2022, pp. 1-14. https://proceedings.neurips.cc/paper_files/paper/2022/file/1cac8326ce3fbe79171db9754211530c-Paper-Conference.pdf

[3] D. G. Lowe, Object Recognition from Local Scale-Invariant Features, *Proceedings of the International Conference on Computer Vision (ICCV)*, Corfu, Greece, pp. 1150–1157, 1999. https://doi.org/10.1109/ICCV.1999.790410

[4] P. Viola, M. Jones, Rapid Object Detection Using a Boosted Cascade of Simple Features, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Kauai, HI, USA, 2001, pp. I-511–I-518. https://doi.org/10.1109/CVPR.2001.990517

[5] A. Krizhevsky, I. Sutskever, G. E. Hinton, ImageNet Classification with Deep Convolutional Neural Networks, *Advances in Neural Information Processing Systems (NeurIPS)*, Lake Tahoe, NV, USA, 2012, pp. 1097–1105. https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

[6] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You Only Look Once: Unified, Real-Time Object Detection, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, pp. 779–788. https://doi.org/10.1109/CVPR.2016.91

[7] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A. C. Berg, SSD: Single Shot Multibox Detector, *Proceedings of the European Conference on Computer Vision (ECCV)*, Amsterdam, Netherlands, 2016, pp. 21–37. https://doi.org/10.1007/978-3-319-46448-0_2

[8] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, *Advances in Neural Information Processing Systems (NeurIPS)*, Montreal, Canada, 2015, pp. 91–99. https://proceedings.neurips.cc/paper_files/paper/2015/file/14bfa6bb14875e45bba028a21ed38046-Paper.pdf

[9] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, Microsoft COCO: Common Objects in Context, *Proceedings of the European Conference on Computer Vision (ECCV)*, Zurich, Switzerland, 2014, pp. 740–755. https://doi.org/10.1007/978-3-319-10602-1_48

[10] A. Hore, D. Ziou, Image Quality Metrics: PSNR vs. SSIM, *Proceedings of the International Conference on Pattern Recognition (ICPR)*, Istanbul, Turkey, 2010, pp. 2366–2369. https://doi.org/10.1109/ICPR.2010.579

[11] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, Image Quality Assessment: From Error Visibility to Structural Similarity, *IEEE Transactions on Image Processing*, Vol. 13, No. 4, pp. 600–612, April, 2004. https://doi.org/10.1109/TIP.2003.819861

[12] Expert Market Research, *CCTV Camera Market Size 2025–2034*, Available: https://www.expert marketresearch.com/, accessed October 2025.

[13] R. Vandaele, G. A. Nervo, O. Gevaert, Topological Image Modification for Object Detection and topological image processing of skin lesions, *Scientific Reports*, Vol. 10, Article No. 21061, December, 2020. https://doi.org/10.1038/s41598-020-77933-y

[14] S. Singh, A. Yadav, J. Jain, H. Shi, J. Johnson, K. Desai, Benchmarking Object Detectors with COCO: A New Path Forward, *Proceedings of the European Conference on Computer Vision (ECCV)*, Milan, Italy, 2024, pp. 279–295. https://doi.org/10.1007/978-3-031-72784-9_16

[15] Z. Wang, J. Guo, D. Bu, C. Shi, Investigating Failure Patterns in Machine Learning-Based Object Detection Tasks in Software Development Courses, *Journal of Internet Technology*, Vol. 24, No. 4, pp. 1001–1008, July, 2023. https://doi.org/10.53106/160792642023072404017

[16] Y. Hao, H. Pei, Y. Lyu, Z. Yuan, J.-R. Rizzo, Y. Wang, Understanding the Impact of Image Quality and Distance of Objects to Object Detection Performance, *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Detroit, MI, USA, 2023, pp. 11436–11442. https://doi.org/10.1109/IROS55552.2023.10342139

[17] G. Yin, B. Liu, H. Zhu, T. Gong, N. Yu, A Large Scale Urban Surveillance Video Dataset for Multiple-Object Tracking and Behavior Analysis, *arXiv preprint*, arXiv: 1904.11784, July, 2020. https://arxiv.org/abs/1904.11784

[18] M. Ofori-Oduro, M. Amer, Defending Object Detection Models Against Image Distortions, *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, Waikoloa, HI, USA, 2024, pp. 3842–3851. https://doi.org/10.1109/WACV57701.2024.00381

[19] N. Panwar, S. Sharma, A. Singh, A Survey on 5G: The Next Generation of Mobile Communication, *Physical Communication*, Vol. 18, pp. 64–84, March, 2016. https://doi.org/10.1016/j.phycom.2015.10.006

[20] S. A. H. Mohsan, Y. Li, A Contemporary Survey on 6G Wireless Networks: Potentials, Recent Advances, Technical Challenges and Future Trends, *arXiv preprint*, arXiv:2306.08265, June, 2023. https://arxiv.org/abs/2306.08265

[21] J. Akhtman, L. Hanzo, Power versus Bandwidth-Efficiency in Wireless Communications: The Economic Perspective, *Proceedings of the IEEE Vehicular Technology Conference (VTC)*, Anchorage, AK, USA, 2009, pp. 1–5. https://doi.org/10.1109/VETECF.2009.5379027

[22] T. Makhalanyane, L. Mamushiane, H. Kobo, A. Lysko, Towards a 5G Equipped Video Surveillance UAV for Public Safety, *IST-Africa Conference Proceedings*, Dublin, Ireland, 2024, pp. 1–11. https://doi.org/10.23919/IST-Africa63983.2024.10569324

[23] J. Zhang, L. Yu, S. Liu, Y. Cai, Y. Zhang, H. Xing, T. Jiang, Wireless Environmental Information Theory: A New Paradigm Toward 6G Online and Proactive Environment Intelligence Communication, *Engineering*, pp. 1–20, August, 2025. https://doi.org/10.1016/j.eng.2025.07.028

[24] M. Adaramola, M. Adelabu, Implementation of Closed-circuit Television (CCTV) Using Wireless Internet Protocol (IP) Camera, *Innovative Systems Design and Engineering*, Vol. 8, No. 5, pp. 10-19, July, 2017.

https://iiste.org/Journals/index.php/ISDE/article/view/37737

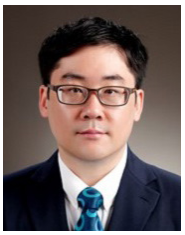
[25] B.-K. Kim, S.-H. Wang, J. Lee, Efficient Object Detection Using Edge Computing for Collected Video, *Journal of Advanced Navigation Technology*, Vol. 29, No. 2, pp. 242–247, April, 2025. http://dx.doi.org/10.12673/jant.2025.29.2.242

[26] Y. LeCun, C. Cortes, C. J. C. Burges, *The MNIST Database of Handwritten Digits*, Available: http://yann.lecun.com/exdb/mnist/, accessed October 2025.

[27] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, F.-F. Li, ImageNet: A Large-Scale Hierarchical Image Database, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Miami, FL, USA, pp. 248–255, 2009. https://doi.org/10.1109/CVPR.2009.5206848

[28] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, A. Torralba, Places: A 10 Million Image Database for Scene Recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 40, No. 6, pp. 1452–1464, June, 2018. https://doi.org/10.1109/TPAMI.2017.2723009

Biography



Pill-Won Park, Ph.D. in Computer and Radio Communications Engineering, Computer Science and Engineering, Korea University, August 2017. March 2025 – Present: Full-time Lecturer in Software Education, Hanyang University. Research interests: Mobile communication, sensors, machine learning, deep learning.