Air Leak Detection Using Sobel-Enhanced YOLO Algorithm from Infrared Images

Shuaiang Rong[1], Emadeldeen Hamdan[1], Hamed Khaleghi[2], Aslihan Karatas[2], Ahmet Enis Cetin[1]
[1]Department of Electrical and Computer Engineering, [2]Department of Civil, Materials, and Environmental Engineering
University of Illinois Chicago, IL, USA
Emails: srong4@uic.edu, ehamda3@uic.edu, hkhale3@uic.edu, akaratas@uic.edu, aecyy@uic.edu

2025 IEEE International Symposium on Circuits and Systems (ISCAS) | 979-8-3503-5683-0/25/$31.00 ©2025 IEEE | DOI: 10.1109/ISCAS56072.2025.11043806

***Abstract*—Air leakage significantly contributes to energy loss in commercial and residential building envelopes. For the next generation of sustainable building construction, enhancing energy efficiency through effective air leak detection is paramount. Traditional air leak tests rely on intrusive methods and manual detection, which lack precision and speed in locating leaks. The rapid advancements of artificial intelligence offer tremendous opportunities to revolutionize these detection methods. This paper introduces an automatic real-time detection method based on the integration of an edge detection Sobel block with the standard real-time object detector YOLO (You Only Look Once) framework. It transforms single-channel infrared (IR) images into three channels that provide additional edge information while retaining the original data. It has been implemented in various YOLO versions to demonstrate its effectiveness with training on a custom dataset primarily collected from a test chamber experiment. The results show that the proposed enhanced models achieve fast and accurate detection with compact model size, making it easily implementable in handheld IR cameras or drone-mounted systems for both indoor and outdoor applications. This allows a significant advancement in the non-intrusive and precise detection of air leaks.**

***Index Terms*—YOLO, air leak detection, Sobel filters, non-intrusive, real-time, deep learning**

I. Introduction

Homes and commercial buildings make up 40% of U.S. energy use, with air leaks contributing to significant inefficiencies [1]. Air leak, also known as infiltration, is the accidental entry of external air through cracks and doorways, driven by wind, the stack effect, or building mechanical systems [2]. Humid air infiltrating buildings during summer and warm indoor air seeping into colder cavities in winter can cause condensation within the structure, leading to mold, rot, and corrosion [3-4]. Air leaks are responsible for 25% to 40% of the energy used for heating and cooling in typical homes, significantly affecting comfort levels and contributing to about 360 million tons of greenhouse gas emissions released annually [5-6].

The current method for evaluating air leaks typically relies on the blower door test, which uses a powerful fan mounted on an exterior door to alter indoor pressure and induce air leaks [7-8]. The test results are represented by the airflow rate, which measures the total volume of air moving through the building at a given pressure difference between the interior and exterior [9]. While effective for assessing the overall airtightness of a building, this test does not provide information about the specific locations of leaks [10-11]. Recently, infrared thermography (IRT) has emerged as a promising technique for identifying air leak locations, as it detects temperature differences caused by air leaks at and around the points where leaks occur [12].

Several studies have utilized IRT to assess air leaks [13-15], but these studies primarily rely on traditional methods focused on overall airtightness, analyzing pressure and temperature differences at specific leak points. In these approaches, IRT functions mainly as a thermal imaging tool, with leak areas manually selected as regions or lines of interest. Pixel values within these regions are then analyzed from static or time-sequenced images for temperature measurements. While effective in controlled experiments with pre-designed or highly visible leak points due to significant pressure differences, the detection process relies heavily on manual intervention and is labor-intensive for large-scale building applications or when a large number of detections are required. Currently, there are no automated systems capable of accurately detecting air leaks in real-time scenarios.

This paper proposes an automatic deep learning method that utilizes an enhanced YOLO algorithm to detect air leak points by identifying patterns, including texture, shape, and edges of the air leak and its surrounding area in IR images. The approach integrates a novel Sobel block, which adds two additional feature channels to the original single-channel IR image, seamlessly integrating with the YOLO algorithm. The neural network is trained using image data collected from an environmental chamber experiment, covering scenarios with varying pressure levels to help the model recognize both weak and strong air leak cases. The proposed Sobel block enhances YOLO's ability to achieve high detection accuracy with limited image data by effectively extracting edge as prior information. Its simple and robust implementation in lightweight YOLO algorithms makes it suitable for easy deployment on mobile systems, enabling real-time air leak detection.

II. Sobel-Enhanced YOLO Algorithm

A. YOLO Algorithm

YOLO represents a major breakthrough in object detection for its speed and accuracy [16]. Unlike traditional methods like Faster RCNN [17], which use a Region Proposal Network (RPN) to identify potential areas of interest and then analyze

This work is funded by NSF CIVIC-PG Track Project no: 2431169

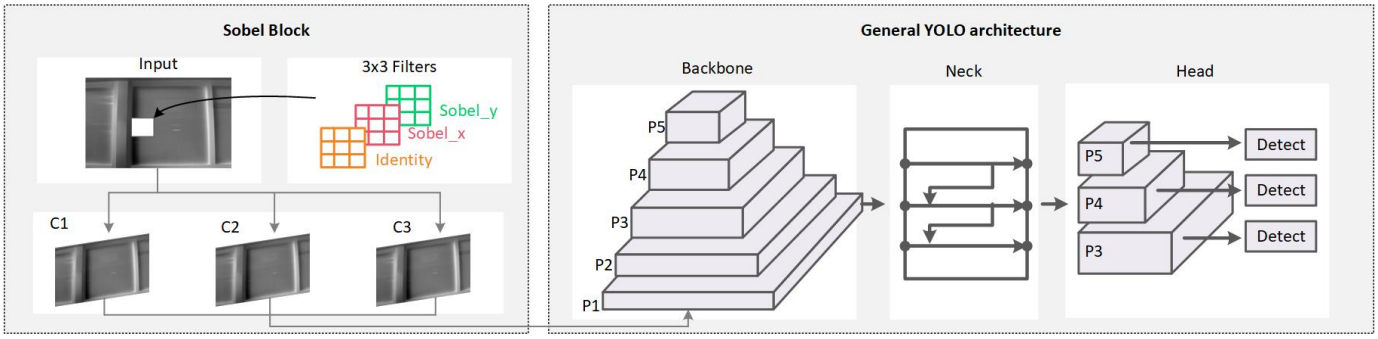


Fig. 1: Sobel filtering enhanced YOLO network structure

these regions separately, YOLO takes a fundamentally different approach. It utilizes an end-to-end neural network that simultaneously predicts bounding boxes and class probabilities, reducing computational demands and making real-time detection possible.

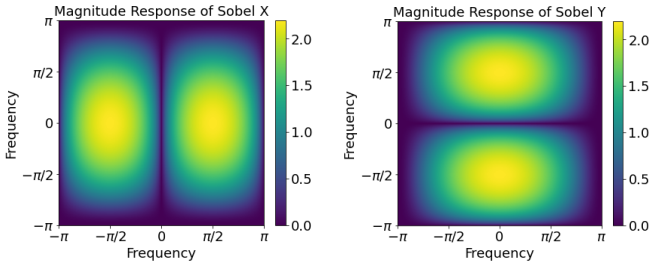
Over the years, YOLO has evolved with over a dozen variations within the series [18]. The foundational architecture comprises three main components: the backbone, neck, and head, as shown in Fig. 1. This architecture has seen several refinements in subsequent versions, such as transitioning from Darknet-53 in YOLOv3 [19] to EfficientNet [20] in YOLOv5 [21]. The neck has also developed from the Feature Pyramid Network (FPN) in YOLOv3 to the BiFPN in YOLOv5, and later to the Path Aggregation Network (PAN). The configuration of the head varies, being either coupled or decoupled in different versions. Moreover, enhancements like data augmentation and NMS-free training have been integrated into the YOLO series.

The core process remains consistent: the backbone downsamples the input to generate feature maps (P1–P5) at decreasing scales. P1 and P2 produce larger, shallower feature maps that retain detailed spatial information essential for detecting small objects, while P4 and P5 generate smaller, deeper feature maps that capture abstract features crucial for recognizing large objects. The neck merges these multiscale features through bottom-up and/or top-down approaches, effectively enhancing the model's capability to detect objects of varying sizes. The head integrates layers P3, P4, and P5, effectively leveraging multiscale strengths to detect objects across a wide range of sizes.

B. Sobel Filter Bank

To enhance the processing of IR images, we designed a Sobel filtering block that can be integrated into the initial layers of the YOLO backbone. This block consists of three convolutional layers, each with a single 3-by-3 filter. The first layer applies an identity filter (the kernel has a 1 in the center and 0 elsewhere) to produce an identical copy of the original IR image. The second and third layers apply the Sobel operator to detect horizontal and vertical edges, respectively.

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad S_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$



(a) (b)

Fig. 2: 2D Frequency responses of (a) Sobel X and (b) Sobel Y filters, respectively.

The Sobel operator is a 3×3 approximation of the Derivative of Gaussians (DoG) for edge extraction using two 3x3 convolution kernels S_x and S_y, where one is a 90° rotation of the other. Their frequency responses are depicted in Fig. 2. Compared to complex methods like Canny, Sobel is faster, easier to implement, and less sensitive to noise, requiring fewer resources and tuning [22]. Commonly used on grayscale images, Sobel estimates gradient magnitudes through 2D spatial measurements, highlighting high-frequency edge regions [23-25]. It enhances edge directionality by capturing horizontal and vertical gradients, making it ideal for detecting air leaks around windows, which typically align with these directions.

After the convolution, the channels are processed through an absolute layer followed by a LeakyReLU layer, which helps manage negative values. This designed Sobel block enhances the extraction of edge features, thereby allowing the YOLO network to capture more detailed information from the IR images. Due to the limited size of our dataset, the network may struggle to learn crucial edge-like features, particularly if these appear sparsely or exhibit subtle gradients. Integrating Sobel filters into the YOLO network provides prior knowledge that air leaks typically appear as boundary-like structures in IR images, effectively guiding attention to high-frequency components associated with air leaks.

III. IR Image Dataset

A. Image Data Collection

As depicted in Fig. 3, two extruded polystyrene (XPS) wall panels were bonded together with sealant and Zip System tape to form a single panel with a vertical joint, where the central area was selected to introduce air leaks and capture IR images. The XPS wall panel was positioned between two sides of the BEI Lab Walk-in Environmental Chamber: (1) the metering chamber, representing the outdoor environment, and (2) the ambient chamber, representing the indoor environment. The chamber applied air pressure differentials across the panel in various stages, allowing evaluation of the air leakage in accordance with ASTM E2357 [26] and ASTM E283 [27] standards. An aluminum foil was attached beside the air leakage source on the joint to calibrate the IR camera for temperature measurement. To capture the full extent of air leakage through the crack in the XPS wall panel while minimizing reflections, the VarioCAM HDx camera was positioned on a tripod 20 inches away from the panel at a 75º angle to the normal. This placement ensured optimal visibility of thermal patterns associated with air leakage while reducing glare and distortions that could affect image accuracy.

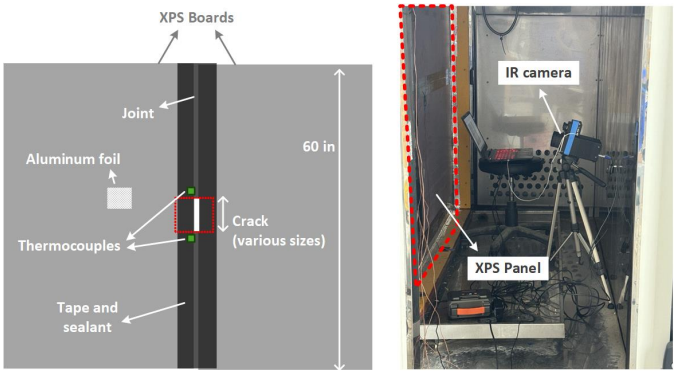


Fig. 3: Environmental chamber setup

B. Customized Image Dataset

The collected IR images capture air leaks through cracks under various pressure differences, ranging from mild to strong scenarios. The images were taken inside the chamber, and since the leaked air is relatively cold, it appears as darker areas. To simulate hot air leaks and expand the dataset, we created additional images by inverting the collected ones, as shown in (1).

$$I_{\text{inverted}}(x, y) = 255 - I_{\text{original}}(x, y) \tag{1}$$

where $I_{\text{original}}(x, y)$ is the intensity of the original image at pixel location (x, y), and $I_{\text{inverted}}(x, y)$ is the intensity of the inverted image at the same pixel location. Inverted images can effectively represent hot air leaks, as IR thermography prioritizes relative temperature differences over absolute values. Since leak edges maintain their structure with only a polarity reversal, geometric features remain intact. This augmentation enhances the network's ability to detect leaks across varying thermal conditions.

Furthermore, we gathered IR images of window air leaks from buildings in UIC (University of Illinois Chicago) to include more realistic scenarios, which were also processed to generate inverted versions. All images were then manually labeled with bounding boxes to annotate the air leak areas. As a result, the customized IR image dataset includes 92 images and 313 air leak instances. Fig. 4 shows a mosaic of some of the labeled images in the dataset.

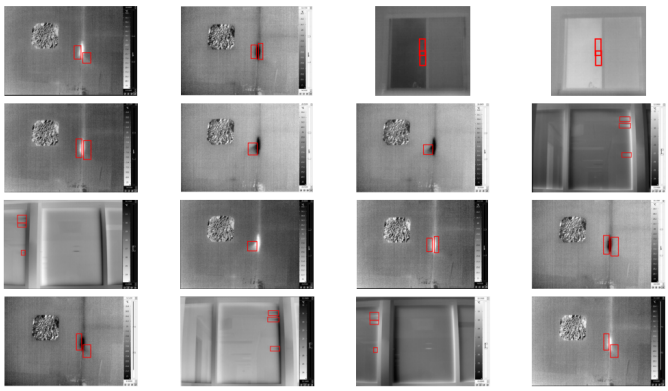


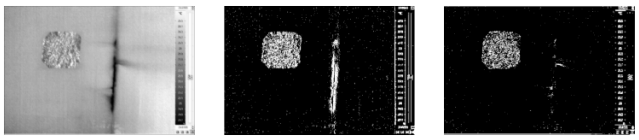
Fig. 4: Labeled image examples

IV. Experimental Results

The proposed Sobel block is integrated into various versions of the YOLO algorithms to evaluate its effectiveness. The Sobel-enhanced YOLO models were trained and tested using the customized IR image dataset, which was split into training and validation sets with an 8:2 ratio. The results and improvements in performance are presented below.

A. Effectiveness of Sobel Block

Fig. 5 illustrates the output of the Sobel block using a sample image from the dataset. In Fig. 5(a), the first channel processed by the identity filter passes the original image to the YOLO network. Fig. 5(b) and 5(c) display the outputs of the second and third channels, which are processed by Sobel X and Sobel Y filters, respectively, to extract edge information in the vertical and horizontal directions. It is noted that the images are presented in binary for clearer visualization.



(a) (b) (c)

Fig. 5: Sobel block output: (a) Channel 1 (original), (b) Channel 2 (Sobel X), and (c) Channel 3 (Sobel Y).

As a result, the Sobel block effectively enhances edge information while preserving the original image, providing

valuable data to the YOLO network during training. This approach improves the network's ability to detect features by combining both the original and edge-processed information, as demonstrated in the following section.

B. Performance of Sobel-Enhanced YOLO Algorithms

In this study, YOLOv5, YOLOv8, and the well-known real-time detector Single Shot MultiBox Detector (SSD) [28] are selected for evaluation. YOLOv5 and its variants are primarily used to test the Sobel block for their ease of use and flexibility [21]. With support for ONNX, CoreML, and TFLite, YOLOv5 is ideal for deployment. It offers five sizes—nano, small, medium, large, and extra-large. Due to the limited dataset, we evaluate the nano and small versions and their variants, as introduced below.

- YOLOv5n: Nano version with depth and width ratios of 0.33 and 0.25;
- YOLOv5n-p2: Extends YOLOv5n by adding a P2 detection layer (downsample ratio of 4) to improve small object detection;
- YOLOv5s: Small version with depth and width ratios of 0.33 and 0.5;
- YOLOv5s-ghost: Modifies YOLOv5s with GhostConv layers [29] that use fewer convolutional operations to reduce computational costs.

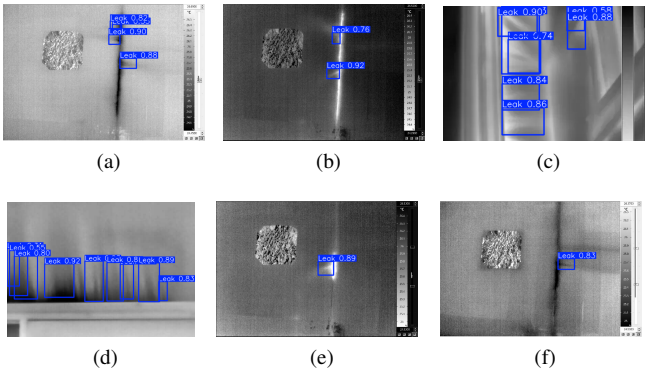
The proposed Sobel block is integrated into the above networks and evaluated, as shown in TABLE I. The results indicate that incorporating the Sobel block into YOLO algorithms leads to notable improvements over the original versions. Specifically, Sobel-YOLOv5n achieves a AP^{50} of 98. 4%, reflecting a 23. 7% increase over the original YOLOv5n. Similarly, Sobel-YOLOv5n-P2 reaches AP^{50} of 93. 5%, representing improvements of 31%, compared to the original version. Although larger models such as YOLOv5s and YOLOv5s ghost show lower AP^{50} scores due to the limited dataset, the Sobel block still improves their performance.

Additionally, the Sobel-YOLOv5n model is tested on a computer equipped with an NVIDIA GeForce RTX 4060 laptop GPU, which has 8188 MiB of memory. It achieves a high frame rate of 97.1 FPS (frames per second) and maintains a compact size of 1.8M parameters. In addition, we evaluated the Sobel block on the newer YOLOv8n model [30], which achieved an average precision (AP50) of 99.4%. The addition of the Sobel block to YOLOv8n resulted in a mild improvement of 0.1% in AP^{50}. However, it is important to note that YOLOv8n has 3M parameters, much more than the proposed Sobel-YOLOv5n, due to its C2f backbone, BiFPN neck and decoupled head, replacing the lighter architecture of YOLOv5. Although these upgrades improve detection, they also increase the size of the model. In contrast, the proposed Sobel-YOLOv5n enhances detection performance to a level comparable to YOLOv8 with just a single additional layer, while remaining lightweight and efficient for real-time embedded systems. To enhance the comparison, the Sobel-enhanced YOLO models are evaluated against SSD300 (VGG16) [31], which exhibits a low AP^{50} of 46. 1% and a AP^{50-90} of 31. 2%.

TABLE I: Performance Results

Model	AP^{50}	AP^{50-90}	Params(M)	FPS
YOLOv5n	0.747	0.31	1.7	100
Sobel-YOLOv5n	**0.984**	**0.652**	**1.8**	**97.1**
YOLOv5n-p2	0.628	0.269	1.8	181
Sobel-YOLOv5n-p2	**0.935**	**0.575**	**1.8**	**78.7**
YOLOv5s	0.395	0.136	7.0	99
Sobel-YOLOv5s	**0.444**	**0.178**	**7.1**	**95.2**
YOLOv5s-ghost	0.497	0.174	3.7	83
Sobel-YOLOv5s-ghost	**0.591**	**0.255**	**3.8**	**71.9**
YOLOv8n	0.994	0.894	3.0	94.3
Sobel-YOLOv8n	**0.995**	**0.859**	**3.0**	**62**
SSD300 (VGG16)	0.461	0.312	35.6	19

Fig. 6 presents examples of detection results using the trained Sobel-YOLOv5n model. Air leaks are effectively identified with bounding boxes in the test images, including strong leaks as shown in Fig. 6(b) and 6(e), and weak leaks as shown in Fig. 6(a) and 6(f). Realistic air leakage scenarios are also accurately detected, as demonstrated in Fig. 6(c) and 6(d). This shows that the trained Sobel-Enhanced YOLO network can effectively detect air leaks in various scenarios.



(a) (b) (c)

(d) (e) (f)

Fig. 6: Detection results using Sobel-YOLOv5n

V. Conclusion

This paper proposes a smart air leak detection method using IR images based on a Sobel filtering-enhanced YOLO algorithm using a designed Sobel block to provide prior information to guide the YOLO algorithm for effective feature extraction. The results demonstrate that the proposed Sobel block greatly improves the accuracy of various YOLOv5 models, achieving gains of 5% to 31% in AP^{50} and 5% to 35% in AP^{50-90}. Notably, Sobel-YOLOv5n, with only 1.8M parameters, maintains a high detection speed of 97.1 FPS and 98.4% accuracy, offering performance comparable to YOLOv8 but with a much smaller model size. This makes it highly suitable for real-time applications on mobile platforms, including handheld IR cameras and drone-mounted systems. In the future, field-based data collection will be a key focus of our ongoing work.

REFERENCES

[1] U.S. Department of Energy, "Why Energy Efficiency Matters," Energy Saver, [Online]. Available: https://www.energy.gov/energysaver/why-energy-efficiency-matters.

[2] V. E. M. Cardoso, P. F. Pereira, N. M. M. Ramos, and R. M. S. F. Almeida, "The Impacts of Air Leakage Paths and Airtightness Levels on Air Change Rates," *Buildings*, vol. 10, no. 3, Art. 3, Mar. 2020, doi: 10.3390/buildings10030055.

[3] S. You, W. Li, T. Ye, F. Hu, and W. Zheng, "Study on Moisture Condensation on the Interior Surface of Buildings in High Humidity Climate," *Build. Environ.*, vol. 125, pp. 39–48, Nov. 2017, doi: 10.1016/j.buildenv.2017.08.041.

[4] T. Kalamees, "Air Tightness and Air Leakages of New Lightweight Single-Family Detached Houses in Estonia," *Build. Environ.*, vol. 42, no. 6, pp. 2369–2377, Jun. 2007, doi: 10.1016/j.buildenv.2006.06.001.

[5] U.S. Environmental Protection Agency, "Air Sealing: Sealing Air Leaks and Adding Attic Insulation," ENERGY STAR®, [Online]. Available: https://www.energystar.gov/ia/home_improvement/home_sealing/AirSealingFS_2005.pdf.

[6] J. Crandell, "What's the Big Deal With Air Leakage," Continuous Insulation, Aug. 13, 2019, [Online]. Available: https://www.continuousinsulation.org/content/whats-big-deal-air-leakage.

[7] Y.-S. Hsu, X. Zheng, E. Cooper, M. Gillott, and C. J. Wood, "Evaluation of the Indoor Pressure Distribution During Building Airtightness Tests Using the Pulse and Blower Door Methods," *Build. Environ.*, vol. 195, p. 107742, May 2021, doi: 10.1016/j.buildenv.2021.107742.

[8] A. Frattolillo, L. Stabile, and M. Dell'Isola, "Natural Ventilation Measurements in a Multi-Room Dwelling: Critical Aspects and Comparability of Pressurization and Tracer Gas Decay Tests," *J. Build. Eng.*, vol. 42, p. 102478, Oct. 2021, doi: 10.1016/j.jobe.2021.102478.

[9] X. Zheng, E. W. Cooper, J. Mazzon, I. Wallis, and C. J. Wood, "Experimental Insights Into the Airtightness Measurement of a House-Sized Chamber in a Sheltered Environment Using Blower Door and Pulse Methods," *Build. Environ.*, vol. 162, 2019.

[10] Y. Ji and L. Duanmu, "Air-Tightness Test and Air Infiltration Estimation of an Ultra-Low Energy Building," *Sci. Technol. Built Environ.*, vol. 23, no. 3, pp. 441–448, Apr. 2017, doi: 10.1080/23744731.2017.1262707.

[11] E. L. Hult, M. H. Sherman, and I. Walker, "Blower-Door Techniques for Measuring Interzonal Leakage," 2013, Accessed: Oct. 13, 2024. [Online]. Available: https://escholarship.org/uc/item/4bw629bg.

[12] A. Kirimtat and O. Krejcar, "A Review of Infrared Thermography for the Investigation of Building Envelopes: Advances and Prospects," *Energy Build.*, vol. 176, pp. 390–406, Oct. 2018, doi: 10.1016/j.enbuild.2018.07.052.

[13] T. Feng, Z. Shen, S. S. Shrestha, and D. E. Hun, "A Novel Transient Infrared Imaging Method for Non-Intrusive, Low-Cost, Fast, and Accurate Air Leakage Detection in Building Envelopes," *J. Build. Eng.*, vol. 91, 2024.

[14] M. Mahmoodzadeh, V. Gretka, S. Wong, T. Froese, and P. Mukhopadhyaya, "Evaluating Patterns of Building Envelope Air Leakage With Infrared Thermography," *Energies*, vol. 13, no. 13, Art. 3545, 2020.

[15] E. Barreira, R. M. S. F. Almeida, and M. Moreira, "An Infrared Thermography Passive Approach to Assess the Effect of Leakage Points in Buildings," *Energy Build.*, vol. 140, pp. 158–171, 2017.

[16] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 779–788, 2016.

[17] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection With Region Proposal Networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[18] M. Hussain, "YOLOv1 to v8: Unveiling Each Variant–A Comprehensive Review of YOLO," *IEEE Access*, vol. 12, pp. 42816–42833, 2024.

[19] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[20] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," *arXiv preprint arXiv:1905.11946*, 2019.

[21] G. Jocher, "YOLOv5 by Ultralytics (Version 7.0) [Computer Software]," 2020. Available: https://doi.org/10.5281/zenodo.3908559.

[22] R. C. Gonzalez and P. Wintz, *Digital Image Processing*, 2nd ed., Addison-Wesley, 1987.

[23] O. N. Gerek and A. E. Cetin, "A 2-D Orientation-Adaptive Prediction Filter in Lifting Structures for Image Coding," *IEEE Trans. Image Process.*, vol. 15, no. 1, pp. 106–111, Jan. 2006.

[24] A. Bozkurt, A. Suhre, and A. E. Cetin, "Multi-Scale Directional-Filtering-Based Method for Follicular Lymphoma Grading," *Signal Image Video Process.*, vol. 8, pp. 63–70, 2014.

[25] S. F. Atici, R. Ansari, V. Allareddy, O. Suhaym, A. E. Cetin, and M. H. Elnagar, "Fully Automated Determination of the Cervical Vertebrae Maturation Stages Using Deep Learning With Directional Filters," *PLoS One*, vol. 17, no. 7, 2022.

[26] ASTM E2357-18, "Standard Test Method for Determining Air Leakage Rate of Air Barrier Assemblies," vol. 04.12, p. 11.

[27] ASTM E283/E283M-19, "Standard Test Method for Determining Rate of Air Leakage Through Exterior Windows, Skylights, Curtain Walls, and Doors Under Specified Pressure Differences Across the Specimen," vol. 04.11, p. 7.

[28] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single Shot MultiBox Detector," in *Computer Vision – ECCV 2016*, pp. 21–37, 2016, doi: 10.1007/978-3-319-46448-0_2.

[29] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, "GhostNet: More Features From Cheap Operations," *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 1577–1586, 2020.

[30] G. Jocher, J. Qiu, and A. Chaurasia, "Ultralytics YOLO (Version 8.0.0) [Computer Software]," 2023. Available: https://github.com/ultralytics/ultralytics.

[31] PyTorch, "SSD300 VGG16 (TorchVision Models)," [Online]. Available: https://pytorch.org/vision/main/models/generated/torchvision.models.detection.ssd300_vgg16.html.