International Journal of Combinatorial Optimization Problems and Informatics, 16(3), May-Aug 2025, 300-309. ISSN: 2007-1558.
https://doi.org/10.61467/2007.1558.2025.v16i3.713

www.editada.org

Optimizing Damage Detection in Pipelines with Drone-Based Deep Learning Models

Ali Bin Khalifa[1], Hamad Al-Nuaimi[1], Latifa Bint Ahmed[2], Salem Jassim[1], Affan Alkim[2],
[1] Computer Engineering Department, Qatar University, Qatar
[2] Department of Information Technology, Lusail University, Qatar
E-mails: alkim@lu.edu.qa

Abstract. This study investigates the application of deep learning algorithms, specifically SSD and YOLOv7, for pipeline damage detection using remote sensing technology. The research evaluates the performance of both algorithms in terms of precision, recall, mean average precision (mAP), and F1 score, using a dataset collected through UAV-based imagery. SSD demonstrated superior precision but slightly lower recall, while YOLOv7 excelled in recall and overall detection ability, making it more suitable for comprehensive inspections where missing defects is critical. The findings emphasize the importance of context-specific algorithm selection, with SSD being ideal for real-time monitoring systems and YOLOv7 for applications requiring high recall. Furthermore, the study explores potential improvements to SSD, including transfer learning, data augmentation, and advanced feature extraction techniques, to enhance recall and overall performance. The results offer valuable insights for optimizing pipeline damage detection in varying operational environments. These findings have significant applications in oil and gas transportation, water distribution networks, and industrial pipeline monitoring, where early damage detection is crucial for preventing leaks and structural failures.

Keywords: deep learning, optimization, drone, damage detection, pipeline control

Article Info
Received January 28, 2025
Accepted March 17, 2025

1 Introductıon

Pipelines, widely used in industrial applications, are essential components of modern urban infrastructure. They ensure the safe and efficient transportation of oil, gas, chemicals, and other fluid resources [1]. However, pipelines face significant challenges to their structural integrity, including mechanical stress, corrosion, temperature fluctuations, and various environmental factors [2]. One of the most common and critical issues is the formation of cracks on their internal and external surfaces [3]. If not detected promptly, these cracks can lead to leaks, causing environmental pollution, operational disruptions, and substantial economic losses [4]. Leaks not only result in product loss and increased maintenance costs but also reduce system efficiency. Therefore, early and accurate detection of pipeline damage is crucial for maintaining safety and optimizing maintenance strategies in large-scale pipeline networks [5].

Traditional pipeline monitoring methods, such as visual inspection, ultrasonic testing, magnetic particle testing, and radiographic inspection, are widely used [6]. However, these approaches are often costly, time-intensive, and potentially hazardous to human safety [7]. In contrast, remote sensing has emerged as a powerful alternative for monitoring large-scale infrastructures. By leveraging advanced sensors, it enables data collection from a distance, offering a safer and more efficient solution [8]. The integration of remote sensing technologies with unmanned aerial vehicles (UAVs) has revolutionized pipeline inspection [9]. UAVs equipped with high-resolution cameras, infrared sensors, and LIDAR systems provide fast, cost-effective, and flexible surveillance options [10]. These aerial platforms can access hard-to-reach areas, minimize human involvement in hazardous environments, and deliver accurate data for damage detection and analysis [11]. However, challenges remain, such as harsh weather conditions, variable lighting, and the complex geometries of pipelines, which can affect data quality.

Recent advancements in computer vision have driven the demand for automated detection and tracking solutions in industrial systems [12]. In this context, machine learning-based approaches, particularly object detection algorithms, have become powerful alternatives to traditional methods [13]. Convolutional Neural Networks (CNN)-based models, trained on large datasets, have significantly improved object detection performance, offering robust alternatives to conventional techniques [14]. Among these, the Single Shot Multibox Detector (SSD) is a widely recognized algorithm for real-time object detection [15]. SSD achieves an impressive balance between accuracy and speed by performing object localization and classification within a single network structure, detecting objects at varying scales, and processing high-quality images quickly [16]. Similarly, the You Only Look Once (YOLO) algorithm series approaches object detection as a regression problem, delivering high speed and efficiency [17]. Like SSD, YOLO has become an indispensable solution for real-time applications.

Recent advancements in UAV technology and deep learning-based object detection have significantly enhanced automated inspection capabilities in industrial applications [18, 19]. UAVs have evolved from simple remote-controlled aerial devices to highly autonomous platforms equipped with advanced sensors, real-time processing capabilities, and AI-driven analytics [20, 21]. The use of UAVs in structural monitoring has expanded across various domains, including power line inspections, structural monitoring, and city management, due to their ability to quickly gather high-resolution imagery with minimal operational costs [22-25]. In the field of object detection, CNN-based models such as SSD and YOLO have revolutionized real-time damage identification. SSD's multi-resolution feature maps enable it to detect objects of varying sizes with high accuracy, making it particularly effective for applications requiring a balance between speed and precision [26]. On the other hand, YOLO's single-pass detection architecture allows it to process images significantly faster, making it ideal for real-time applications where rapid decision-making is critical [27, 28]. While both algorithms have been widely adopted for tasks such as autonomous driving and surveillance, their implementation in UAV-assisted pipeline monitoring remains an area of ongoing research. This study builds upon these technological advancements to evaluate and optimize their performance in real-world damage detection scenarios, addressing the challenges posed by diverse environmental conditions and pipeline geometries.

This study focuses exclusively on external pipeline damage detection using UAV-based deep learning models, as internal inspection requires alternative techniques such as ultrasonic or magnetic testing. The study aims to compare the performance of SSD and YOLOv7 algorithms in terms of accuracy, speed, and ease of implementation, proposing a solution suitable for maintenance processes in industrial environments. Furthermore, the behaviors of both algorithms are analyzed under varying conditions, such as different pipeline types, environmental factors, and damage sizes. The findings of this study provide insights into the advantages of each algorithm, not only for pipeline monitoring but also for broader industrial applications.

2 Experimental Procedures

Single-stage object detection algorithms, primarily categorized as You Only Look Once (YOLO) and Single Shot Multibox Detector (SSD), provide fast and efficient solutions for object detection tasks [29]. SSD, while faster than two-stage approaches such as Faster R-CNN, generally offers higher detection accuracy compared to YOLO [30]. A key advantage of SSD lies in its ability to detect both large and small objects using feature maps at different resolutions. Lower-resolution feature maps from the network's deeper layers are optimized for detecting large objects, while higher-resolution feature maps from earlier layers target smaller objects [31]. However, SSD may exhibit lower performance in detecting small objects compared to methods like Faster R-CNN. The popularity of SSD in real-time applications stems from its fast computation and low latency. By leveraging anchor boxes and a multi-layer detection mechanism, SSD effectively adapts to objects of various sizes, making it versatile for applications ranging from autonomous vehicles to security systems and industrial automation [32]. While SSD prioritizes a balance between speed and accuracy, YOLO focuses on achieving higher speed at the cost of slightly reduced accuracy. SSD's ability to strike a balance between these factors makes it one of the most suitable algorithms for object detection in diverse scenarios.

2.1 SSD Network Architecture

The SSD model introduces an innovative approach to object detection, built upon three main components: the backbone network, original bounding box generation, and convolution-based prediction layers. This architecture positions SSD as a leading algorithm in terms of both speed and accuracy. The backbone network consists of a base network and additional feature extraction layers. The process begins by feeding the input image into a deep neural network, which extracts fundamental features and prepares them for object detection. Next, feature maps at multiple scales are generated to design default bounding boxes. These default boxes play a critical role in object classification and location estimation [33]. For each position on the feature maps, the algorithm predicts the coordinates of the bounding box and the associated object class. The final stage employs the Non-Maximum Suppression (NMS) algorithm. When multiple bounding boxes are predicted for the same object, NMS filters out boxes with low confidence scores, retaining only the most reliable prediction. This process eliminates redundancies and enhances detection accuracy, making SSD a robust choice for object detection in computer vision applications.

As illustrated in Figure 1, the SSD network architecture comprises three main components: the backbone network, original bounding box generation, and convolution-based prediction layers. The base network is typically a pre-trained convolutional neural network that facilitates high-level feature extraction from images. A widely used backbone in SSD is VGG-16 (Visual Geometry Group), a deep learning model known for its effective convolutional and max-pooling layers. Each layer of VGG-16 extracts different features from the input image. However, the fully connected layers of VGG-16 are unsuitable for SSD's requirements, so they are replaced with additional convolutional layers. These added layers generate feature maps at various resolutions, enabling the detection of objects of different sizes. For example, high-resolution feature maps are optimized for detecting small objects, while low-resolution feature maps target larger ones. SSD calculates object classifications and corresponding bounding boxes at each position by adding dedicated prediction layers to each feature map. This flexible design enables SSD to detect objects of various sizes with both high speed and accuracy, making it a powerful and versatile model in the field of object detection.

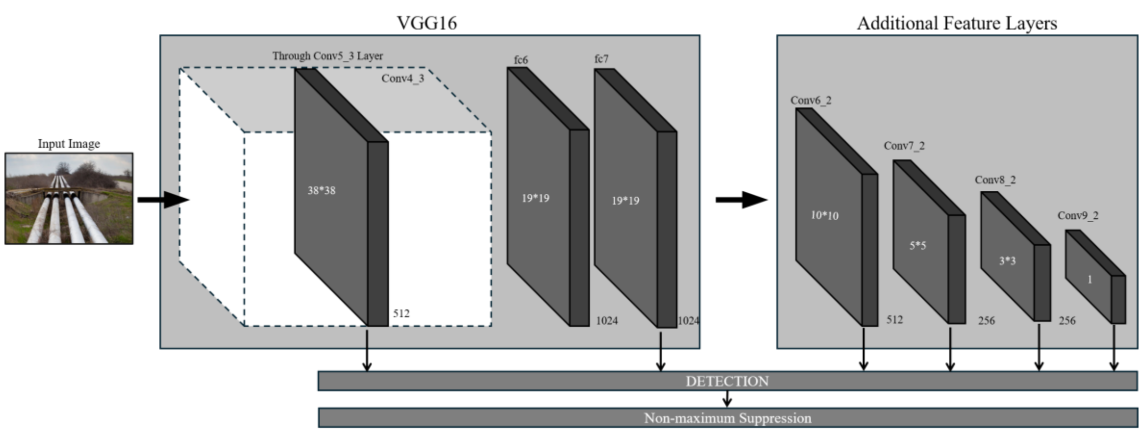

Figure 1. SSD network architecture.

2.2 VGG-16

VGG-16 is a Convolutional Neural Network (CNN) with 16 learnable weight layers. Figure 2 illustrates the VGG-16 network structure, which consists of 13 convolutional layers and 3 fully connected layers. The model progressively increases filter depth, allowing for hierarchical feature extraction, which enhances detection accuracy. [34–36]. Since pooling layers do not require learnable parameters, they are excluded from the weight count. The model accepts input tensors of size 224x224x3 and performs small-scale but highly efficient feature extraction using 3x3 convolutional filters across all convolutional layers. Max-pooling operations are carried out using 2x2 filters with a fixed stride. The architecture progressively increases the number of filters in the convolutional layers, starting with 64 and scaling up to 128, 256, and 512, respectively. Following the convolutional layers are three fully connected layers consisting of 4096, 4096, and 1000 units, respectively [37, 38]. The output layer utilizes a softmax activation function to calculate the probability of the input image belonging to various classes, enabling effective classification. One of the main reasons VGG-16 is favored in this study is its simple yet robust architecture, which allows for easy adaptation to different datasets using methods such as transfer learning. The consistent design of its 3x3 filters enables detailed feature extraction for each class, making it particularly effective during training. This compact yet powerful structure has made VGG-16 a popular choice for a wide range of computer vision tasks.

2.3 Loss Function

The SSD model is a highly effective deep learning approach that optimizes both classification accuracy and location estimation precision for object detection tasks. Its loss function combines two components, classification loss and location loss, to enhance object detection performance and minimize errors [39]. The combined loss function is defined as in Equation 1:

$$L(x,c,l,g) = \frac{1}{N}\left(L_{cls}(x,c) + \alpha L_{loc}(x,l,g)\right) \tag{1}$$

where N represents the number of positive predefined boxes, and α is a weighting coefficient that adjusts the relative importance of location loss.

The classification loss measures how accurately the model assigns predefined boxes to their correct classes. These predefined boxes are categorized as either containing objects (Pos) or representing the background (Neg). The classification loss is calculated as shown in Equation 2:

$$L_{cls}(x,c) = -\sum_{i\in Pos} x_{ij}\log(c_i) - \sum_{i\in Neg}\log(c_0) \tag{2}$$

where x_{ij} is an indicator variable that specifies whether a default box belongs to the j^{th} class. c_i represents the predicted class probabilities for each box, and c_0 is the probability of a box being classified as background.

The location loss evaluates how well the predicted bounding boxes overlap with the ground truth boxes. The difference between the predicted box (l_i) and the ground truth box (g_i) is typically calculated using the $smooth_{L1}$ function as shown in Equation 3:

$$L_{loc}(x,l,g) = \sum_{i\in Pos} x_{ij}\, smooth_{L1}(l_i - g_i) \tag{3}$$

where l_i is the predicted location of the predefined box, and g_i is the ground truth location of the object. The $smooth_{L1}$ function is designed to balance the impact of small and large errors, reducing the influence of outliers while ensuring precise learning of small errors. It is defined as indicated in Equation 4:

$$smoooth_{L1}(z) = \begin{cases} 0.5z^2, & \text{if } |z| < 1 \\ |z| - 0.5, & \text{otherwise} \end{cases} \tag{4}$$

This structure helps limit the effect of large errors on the model while ensuring that smaller errors are handled with precision.

In the SSD model, the number of negative examples often exceeds the number of positive examples, which can lead to imbalanced learning during classification. To address this, the Hard Negative Mining technique is applied [40]. This technique selects negative examples with the highest classification loss and balances their number relative to positive examples by maintaining a fixed ratio. By focusing on the most challenging negative examples, Hard Negative Mining improves classification accuracy and mitigates imbalance.

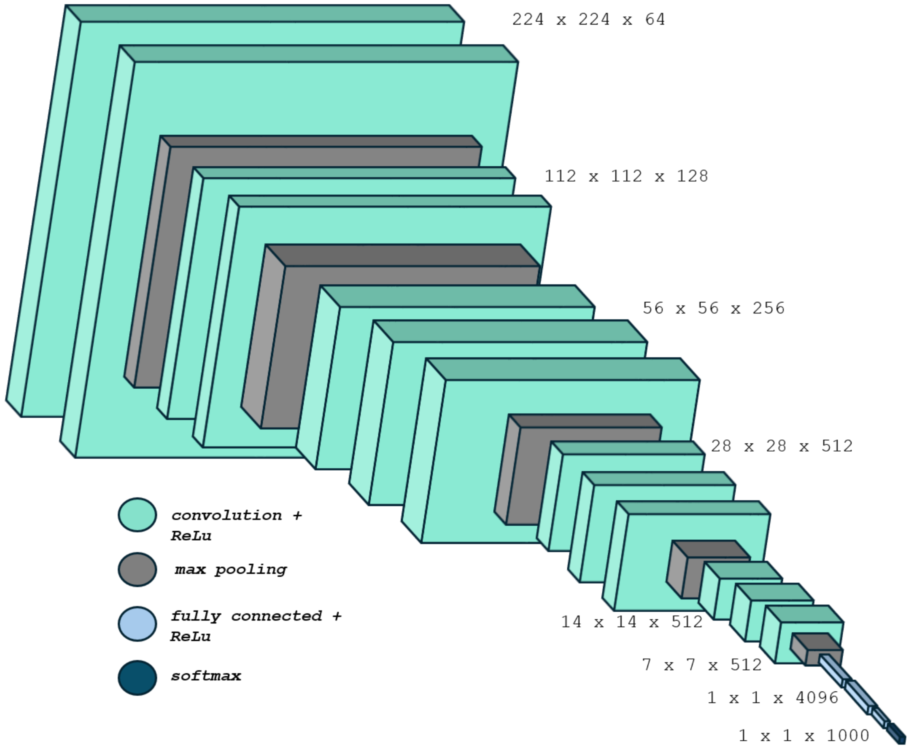

Figure 2. VGG-16's network structure.

2.4 Hardware Components and Operational Stages

The UAV used in this study is a quadcopter equipped with various MEMS sensors and artificial intelligence integration. It is capable of autonomous flight, executing its designated mission, and returning to its initial position, with a total flight time of 40 minutes. The drone captures images using an integrated camera sensor, which are processed by an onboard Raspberry Pi 5 computer before being transmitted to the ground station as video data. Additionally, a ground-based pilot remains in control to manually operate the UAV in case of unexpected scenarios, such as adverse weather conditions. Figure 3 illustrates a simulation of the workflow diagram. Diversity in the dataset plays a critical role in enhancing detection accuracy during algorithm training. A total of 2170 images were used for training and evaluation. These images were sourced from UAV-based aerial footage and publicly available datasets to ensure diversity in pipeline types, damage scenarios, and environmental conditions. Moreover, 70% of the dataset was collected from web-based sources, while 30% was derived from video footage. The dataset encompasses images of pipelines from various geographical regions and includes different pipe types. Efforts were made to capture variability in dimensions (e.g., thickness or thinness), color, angles, and altitudes. Furthermore, the inclusion of images captured under different climatic conditions enriched the dataset and improved the algorithm's robustness.

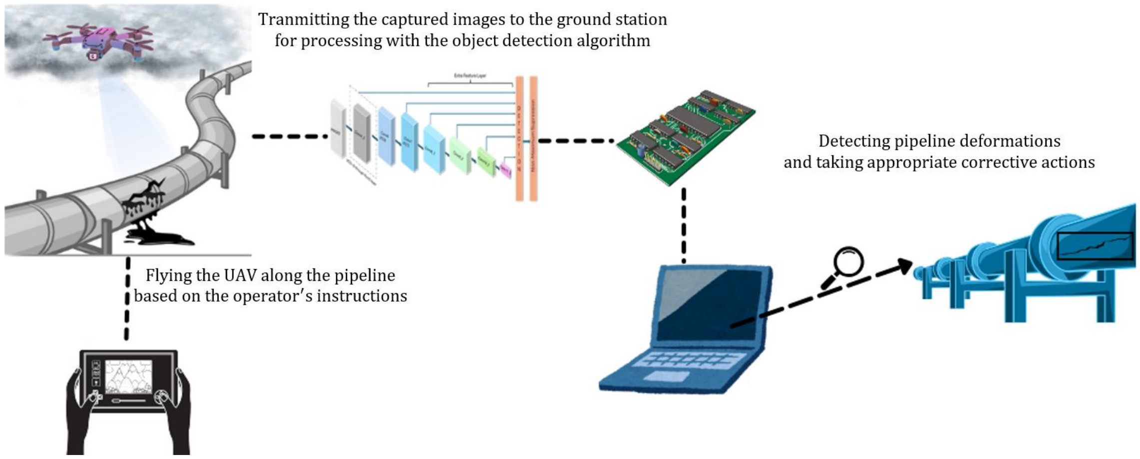

Figure 3. Workflow diagram.

The drone's flight path was preprogrammed using GPS and mapping software. Before takeoff, essential pre-flight checks were conducted, including battery level verification and GPS/IMU sensor calibration. Once airborne, the drone followed its predefined path along the pipeline, capturing images via its camera sensor. These images were processed onboard by the Raspberry Pi, where SSD and YOLOv7 algorithms identified potential pipeline damage. The detected damage areas were recorded in labeled data format. The information, along with the associated images, was compressed and transmitted to the ground station via the drone's WiFi module. The ground station software analyzed this data in real time, marking damaged pipeline areas. A detailed report, including specific information about each detected issue, was generated and shared with the user. To enhance algorithm performance, the model was periodically retrained with new field data. Before each flight, sensor calibration was repeated to maintain accuracy. The ground station, powered by an RTX 4070 GPU, significantly accelerated retraining and optimization processes, enabling efficient data analysis and faster model refinement.

3 Results

Examples of pipeline damage detection using SSD and YOLOv7 algorithms are shown in Figure 4. The performance of these algorithms, integrated with remote sensing technology, was evaluated using precision, recall, mean average precision (mAP), and F1 score. To ensure consistency and reliability, all experiments were conducted five times, and the reported results represent the average values obtained across these iterations. This approach minimizes the impact of any single anomalous result and provides a more robust evaluation of the models' performance. Figure 5 presents a comparative analysis of SSD and YOLOv7 performance metrics, demonstrating their strengths and trade-offs in pipeline damage detection. The SSD algorithm achieved a precision value of 0.719, indicating a relatively low false positive rate and reliable identification of damaged pipeline areas. However, its recall value of 0.656 highlights a slightly diminished ability to detect all instances of damage within the dataset. The mAP was calculated at 0.678, reflecting the overall effectiveness of SSD in accurately detecting and classifying pipeline damage. The F1 score, which balances precision and recall, was 0.686, suggesting a good trade-off between identifying actual damages and minimizing false alarms. On the other hand, the YOLOv7 algorithm achieved a precision value of 0.703, which, while slightly lower than SSD, still demonstrated strong accuracy in damage identification. Its recall value of 0.677 was slightly higher than SSD, indicating a better ability to detect more instances of pipeline damage. YOLOv7's mAP reached 0.685, representing a modest improvement in overall detection performance. The algorithm also achieved a higher F1 score of 0.690, indicating better overall consistency in balancing precision and recall.

Figure 4. Some visual examples of detections made with SSD.

A detailed examination of the metrics reveals that while SSD offers a slight advantage in precision, YOLOv7 outperforms SSD in recall and F1 score, making it better suited for applications where detecting all instances of damage is critical. Additionally, YOLOv7's higher mAP value demonstrates its superior ability to accurately locate damage while maintaining robust classification performance. The SSD algorithm stands out for its speed and simplicity, driven by its single-shot architecture, which enables real-time damage detection. This makes it a valuable option for scenarios requiring quick decision-making. However, the trade-off between speed and accuracy is evident in its recall value, which may limit its use in environments where even small, undetected damages could lead to serious consequences, such as pipeline failures or environmental hazards. The YOLOv7 algorithm, while slightly slower in practice, benefits from a more sophisticated network design that enhances recall and ensures more consistent performance. This makes YOLOv7 a more reliable choice for applications requiring high recall, such as detecting subtle or rare pipeline damages. However, SSD's slightly higher precision value highlights its effectiveness in reducing false alarms, which is crucial for minimizing unnecessary operational downtime caused by false positives.

Moreover, to quantify the speed difference between SSD and YOLOv7, inference time measurements were performed. SSD achieved an average inference time of 18.5 ms per image, while YOLOv7 required 22.3 ms per image. These results reaffirm SSD's suitability for real-time monitoring applications where speed is a critical factor. These results suggest that the choice between SSD and YOLOv7 depends on the specific requirements of the application. For scenarios prioritizing speed and minimizing false positives, SSD offers a suitable solution. Conversely, for applications where maximizing detection and recall is essential, YOLOv7 provides a more reliable alternative. Both algorithms demonstrate strong potential for real-world pipeline damage detection, with their strengths catering to different operational needs. Ultimately, YOLOv7 outperforms SSD in recall and overall detection reliability, making it the preferred choice for applications where minimizing missed detections is critical. The model's superior F1-score and mAP demonstrate its effectiveness in detecting a broad range of pipeline damages under varying conditions.

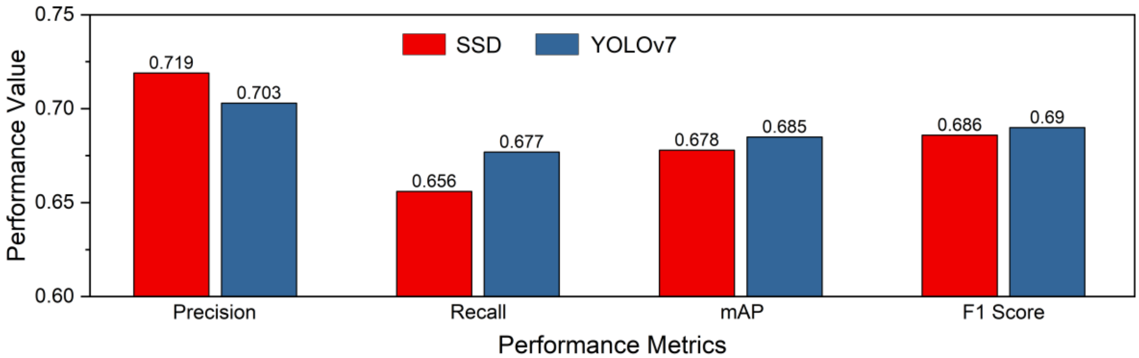

Figure 5. Comparison of detection performances of SSD and YOLOv7 algorithms.

4 Discussion and Conclusion

The findings of this research emphasize the critical importance of selecting algorithms based on the specific operational context. The SSD algorithm excels in precision and computational efficiency, making it ideal for real-time monitoring systems where speed and low latency are paramount. This makes SSD particularly valuable in scenarios requiring rapid decision-making, such as monitoring pipelines in high-risk environments or during emergencies. However, its lower recall suggests potential limitations in detecting subtle or rare defects, which could lead to missed damages in critical applications. On the other hand, YOLOv7 demonstrates superior recall and balanced overall performance metrics, making it a more suitable choice for comprehensive inspections where the risk of missing a defect is unacceptable. Its higher F1 score and mAP underline its capability to maintain consistency in detecting and classifying a broad range of damages, even in complex or diverse environmental conditions. While YOLOv7's slightly lower speed may not match SSD for real-time applications, its advanced network architecture offers a more reliable solution for detailed pipeline assessments.

Future research could explore hybrid approaches that combine the strengths of SSD and YOLOv7 to optimize both precision and recall. A hybrid framework might leverage SSD's rapid detection capabilities for initial screening while utilizing YOLOv7 for detailed analysis of areas flagged as potentially damaged. Such an approach could provide a balance between speed and thoroughness, ensuring both efficiency and reliability in pipeline monitoring systems. Further improvements to the SSD framework could focus on addressing its recall limitations through several techniques, including transfer learning, which leverages pre-trained models on similar datasets to enhance detection capabilities in diverse environments; data augmentation, which expands the variety of training datasets by simulating different pipeline materials, damage types, lighting conditions, and weather scenarios to improve robustness; and advanced feature extraction, which incorporates modern neural network designs, such as attention mechanisms or transformer-based modules, to enhance SSD's ability to detect small and subtle defects. Expanding the dataset used in this research to include more diverse pipeline conditions, such as different materials, damage types, geographical regions, and environmental factors, could provide a more comprehensive evaluation of both algorithms. Additionally, conducting field trials in real-world conditions would help validate the algorithms' performance beyond controlled experimental settings.

The integration of SSD and YOLOv7 into UAV-based systems underscores their significance for real-time pipeline monitoring. Beyond this application, these technologies hold great potential for broader use in industrial infrastructure inspections, bridge safety evaluations, and environmental monitoring—domains where balancing detection accuracy and processing speed is critical. As automated damage detection becomes increasingly vital in infrastructure management, continuous advancements in deep learning algorithms are essential. Future developments in object detection, alongside tailored domain-specific improvements, could substantially lower maintenance costs, mitigate the risk of catastrophic failures, and enhance the overall safety and reliability of critical infrastructure. By aligning algorithm selection with specific operational requirements and exploring hybrid solutions, this research paves the way for more efficient and robust automated inspection systems. Additionally, the findings emphasize the importance of choosing the right detection model based on application needs. Future studies could investigate hybrid approaches that leverage SSD's computational efficiency with YOLOv7's superior recall, further refining pipeline monitoring solutions for practical deployment.

Data Availability

The dataset used in this study is available upon request from the corresponding author. Due to licensing agreements, direct public access is restricted.

References

1. Vishnuvardhan, S., Murthy, A.R., Choudhary, A. (2023). A review on pipeline failures, defects in pipelines and their assessment and fatigue life prediction methods. *International Journal of Pressure Vessels and Piping*, 201(104853). https://doi.org/10.1016/j.ijpvp.2022.104853
2. Zharinova, N.V., Vakulenko, M.V. (2020). Ensuring industrial safety of pipelines with developed hard sections. *IOP Conference Series: Materials Science and Engineering*, 862(3), 032014. https://doi.org/10.1088/1757-899X/862/3/032014
3. Li, Z., Jiang, X., Hopman, H. (2020). Surface crack growth in offshore metallic pipes under cyclic loads: A literature review. *Journal of Marine Science and Engineering*, 8(5), 339. https://doi.org/10.3390/jmse8050339
4. Wong, B., McCann, J.A. (2021). Failure detection methods for pipeline networks: From acoustic sensing to cyber-physical systems. *Sensors*, 21(15). https://doi.org/10.3390/s21154959
5. Razvarz, S., Jafari, R., Gegov, A., Razvarz, S., Jafari, R., Gegov, A. (2021). A review on different pipeline defect detection techniques. *Flow Modelling and Control in Pipeline Systems: A Formal Systematic Approach*, 25-57. https://doi.org/10.1007/978-3-030-59246-2_2
6. Bakirci, M. (2024). Smart city air quality management through leveraging drones for precision monitoring, *Sustainable Cities and Society,* 106, 105390. https://doi.org/10.1016/j.scs.2024.105390
7. Latif, J., Shakir, M.Z., Edwards, N., Jaszczykowski, M., Ramzan, N., Edwards, V. (2022). Review on condition monitoring techniques for water pipelines. *Measurement*, 193(110895).
8. Bakirci, M. (2024). Utilizing YOLOv8 for enhanced traffic monitoring in intelligent transportation systems (ITS) applications, *Digital Signal Processing,* 152, 104594. https://doi.org/10.1016/j.dsp.2024.104594
9. Ho, M., El-Borgi, S., Patil, D., Song, G. (2020). Inspection and monitoring systems subsea pipelines: A review paper. *Structural Health Monitoring*, 19(2), 606-645.
10. Aba, E.N., Olugboji, O.A., Nasir, A., Olutoye, M.A:, Adedipe, O. (2021). Petroleum pipeline monitoring using an internet of things (IoT) platform. *SN Applied Sciences*, 3, 1-12.
11. Bakirci, M. (2024). Enhancing air pollution mapping with autonomous UAV networks for extended coverage and consistency, *Atmospheric Research,* 306, 107480. https://doi.org/10.1016/j.atmosres.2024.107480
12. Xu, Z., Liu, K., Gu, B., Yan, L., Pang, X., Gao, K. (2023). Image recognition model of pipeline magnetic flux leakage detection based on deep learning. *Corrosion Reviews*, 41(6), 689-701.
13. Anupreetham, A., Ibrahim, M., Hall, M., Boutros, A., Kuzhively, A., Mohanty, A., Seo, J.S. (2021). End-to-end FPGA-based object detection using pipelined CNN and non-maximum suppression. In 2021 31st International Conference on Field-Programmable Logic and Applications (pp. 76-82). IEEE Press.
14. Bakirci, M. (2024). Evaluating the impact of unmanned aerial vehicles (UAVs) on air quality management in smart cities: A comprehensive analysis of transportation-related pollution. *Computers and Electrical Engineering,* 119, Part A, 109556. https://doi.org/10.1016/j.compeleceng.2024.109556
15. Shuai, Q., Wu, X. (2020). Object detection system based on SSD algorithm. In 2020 International Conference on Culture-Oriented Science & Technology (pp. 141-144). IEEE Press.
16. Bakirci, M. (2025). Vehicular mobility monitoring using remote sensing and deep learning on a UAV-based mobile computing platform. *Measurement,* 244, 116579. https://doi.org/10.1016/j.measurement.2024.116579

17. Jiang, P., Ergu, D., Liu, F., Cai, Y., Ma, B. (2022). A review of YOLO algorithm developments. *Procedia Computer Science*, 199, 1066-1073.
18. Giro, R.A., Bernasconi, G., Giunta, G., Cesari, S. (2021). A data-driven pipeline pressure procedure for remote monitoring of centrifugal pumps. *Journal of Petroleum Science and Engineering*, 205(108845).
19. Yan, Y., Liu, D., Gao, B., Tian, G.Y., Cai, Z.C. (2020). A deep learning-based ultrasonic pattern recognition method for inspecting girth weld cracking of gas pipeline. *IEEE Sensors Journal*, 20(14), 7997-8006.
20. Bakirci, M., Dmytrovych, P., Bayraktar, I., Anatoliyovych, O. (2024). Multi-class vehicle detection and classification with YOLO11 on UAV-captured aerial imagery. In *2024 IEEE 7th International Conference on Actual Problems of Unmanned Aerial Vehicles Development (APUAVD)* (pp. 191-196). IEEE Press. https://doi.org/10.1109/APUAVD64488.2024.10765862
21. Chen, W., Huang, H., Peng, S. et al. (2021). YOLO-face: a real-time face detector. Visual Computing, 37, 805–813. https://doi.org/10.1007/s00371-020-01831-7
22. Zhang, F., Liu, H., Ma, Z., Yang, Y., Wan, X. (2020). Study of UAV Application in Wireless Sensor Networks. In *2020 3rd International Conference on Mechanical, Electronics, Computer, and Industrial Technology* (pp. 343-348). IEEE Press.
23. Garg, D., Goel, P., Pandya, S., Ganatra, A., Kotecha, K. (2018). A Deep Learning Approach for Face Detection using YOLO. In *2018 IEEE Punecon* (pp. 1-4). IEEE Press.
24. Bakirci, M., Bayraktar, I. (2024). The cutting-edge YOLO11 for advanced aircraft detection in synthetic aperture radar (SAR) imagery. In *2024 8th International Symposium on Innovative Approaches in Smart Technologies (ISAS)* (pp. 1-6). IEEE Press. https://doi.org/10.1109/ISAS64331.2024.10845222
25. Zhao, Y., Shu, Q. (2025). Debiased hybrid contrasitve learning with hard negative mining for unsupervised person re-identification. *Digital Signal Processing*, 156(104826).
26. Gupta, L., Jain, R., Vaszkun, G. (2016). Survey of Important Issues in UAV Communication Networks. *IEEE Communications Surveys & Tutorials*, 18(2), 1123-1152.
27. Bakirci, M. (2023). Data-driven system identification of a modified differential drive mobile robot through on-plane motion tests. *Electrica,* 23(3), 619-633. https://doi.org/10.5152/electrica.2023.22164
28. Tsouros, D.C., Bibi, S., Sarigiannidis, P.G. (2019). A Review on UAV-Based Applications for Precision Agriculture. Information, 10, 349. https://doi.org/10.3390/info10110349
29. Li, T., Hong, B. (2020). An optimized YOLO method for object detection. In 2020 16th International Conference on Computational Intelligence and Security (pp. 30-34). IEEE Press.
30. Chen, Z., Guo, H., Yang, J., Jiao, H., Feng, Z., Chen, L., Gao, T. (2022). Fast vehicle detection algorithm in traffic scene based on improved SSD. *Measurement*, 201(111655).
31. Li, Y., Huang, H., Xie, Q., Yao, L., Chen, Q. (2018). Research on a surface defect detection algorithm based on MobileNet-SSD. *Applied Sciences*, 8(9), 1678.
32. Bakirci, M. (2023). Simulation of autonomous driving for a line-following robotic vehicle: determining the optimal manoeuvring mode. *Elektronika ir Elektrotechnika,* 29(6), 4-11. https://doi.org/10.5755/j02.eie.32364
33. Liu, B., Zhang, X., Gao, Z., Chen, L. (2017). Weld defect images classification with VGG16-based neural network. In International Forum on Digital TV and Wireless Multimedia Communications (pp. 215-223). IEEE Press.
34. Guan, Q., Wang, Y., Ping, B., Li, D., Du, J., Qin, Y., Xiang, J. (2019). Deep convolutional neural network VGG-16 model for differential diagnosing of papillary thyroid carcinomas in cytological images: A pilot study. *Journal of Cancer*, 10(20), 4876-4885.
35. Kaur, T., Gandhi, T.K. (2019). Automated brain image classification based on VGG-16 and transfer learning. In 2019 International Conference on Information Technology (pp. 94-98). IEEE Press.
36. Alippi, C., Disabato, S., Roveri, M. (2018). Moving convolutional neural networks to embedded systems: the AlexNet and VGG-16 Case. In 2018 17th ACM-IEEE International Conference on Information Processing in Sensor Networks (pp. 212-223). IEEE Press.
37. Bakirci, M., Bayraktar, I. (2024). YOLOv9-enabled vehicle detection for urban security and forensics applications. In *2024 12th International Symposium on Digital Forensics and Security (ISDFS)* (pp. 1-6). IEEE Press. https://doi.org/10.1109/ISDFS60797.2024.10527304
38. Bagaskara, A., Suryanegara, M. (2021). Evaluation of VGG-16 and VGG-19 deep learning architectures for classifying dementia people. In 2021 4th International Conference of Computer and Informatics Engineering (pp. 1-4). IEEE Press.
39. Yu, G., Fan, H., Zhou, H., Wu, T., Zhu, H. (2020). Vehicle target detection method based on improved SSD model. *Journal of Artificial Intelligence*, 2(3), 125.
40. Bakirci, M., Bayraktar, I. (2024). Boosting aircraft monitoring and security through ground surveillance optimization with YOLOv9. In *2024 12th International Symposium on Digital Forensics and Security (ISDFS)* (pp. 1-6). IEEE Press. https://doi.org/10.1109/ISDFS60797.2024.10527349