TÜRKİYE CUMHURİYETİ'NİN YÜZÜNCÜ YILI

Advances in Electrical, Electronics and Software Engineering: Innovations Shaping the Future

Editör
Behiye Şahin

BİDGE

BIDGE Publications

Advances in Electrical, Electronics and Software Engineering: Innovations Shaping the Future

Editor: Dr. Behiye Şahin

ISBN: 978-625-372-578-5

Page Layout: Gözde YÜCEL

1st Edition:

Publication Date: 14.12.2024

BIDGE Publications,

Certificate No: 71374

www.bidgeyayinlari.com.tr - bidgeyayinlari@gmail.com

Krc Bilişim Ticaret ve Organizasyon Ltd. Şti.

Güzeltepe Mahallesi Abidin Daver Sokak Sefer Apartmanı No: 7/9 Çankaya / Ankara

Content

CHAPTER I

Detection Performance Evaluation and Real-Time Analysis of YOLOv8 for Military Vehicle Categorization

Furkan Yildirim[1]
Bora Burak Azer[2]
Fikrican Ciftci[3]

1. Introduction

In recent years, advancements in technology have transformed various fields, particularly in how information and intelligence are applied in modern practices. This is especially evident in the fields of transportation management and military operations, where object detection has become a cornerstone of operational success. Vehicle detection, particularly through aerial photography, plays a dual role in both civilian and military contexts (Ammour et al., 2017). In civilian applications, detecting and tracking vehicles significantly improve traffic flow, reduce

[1] Furkan Yildirim, Koluman Automotive Industry Inc., R&D Department, Ankara/Turkiye, Orcid: 0009-0009-5478-3008, furkan.yildirim@koluman.com
[2] Bora Burak Azer, Koluman Automotive Industry Inc., R&D Department, Ankara/Turkiye, Orcid: 0009-0004-3863-1526, burak.azer@koluman.com
[3] Fikrican Ciftci, Koluman Automotive Industry Inc., R&D Department, Ankara/Turkiye, Orcid: 0009-0004-0538-4995, fikrican.ciftci@koluman.com

congestion, and decrease accident rates in traffic management systems (Nellore & Hancke, 2016). On the military front, airborne object detection is a critical component of Battlefield Situational Awareness (BSA), which is essential for the success of future information warfare. With the increased availability of specialized UAVs and advancements in real-time data processing, reliance on aerial photography and object detection systems has grown exponentially (Bakirci & Bayraktar, 2024). However, challenges such as memory overload and processing speed persist. These limitations underscore the importance of adopting advanced methodologies like Convolutional Neural Networks (CNNs) to achieve efficient and accurate object detection in these critical applications (Krichen, 2023). Object detection involves identifying and locating specific objects within images or video streams, making it a fundamental aspect of intelligent systems. In the context of aerial photography, detecting vehicles from high altitudes is complex due to factors such as varying lighting conditions, occlusions, and the diversity of vehicle types (Veeranampalayam Sivakumar et al., 2020). Nevertheless, advances in deep learning have provided powerful tools to address these challenges. Specifically, CNNs have emerged as a leading technology for object detection, offering automatic and robust feature extraction capabilities. Unlike traditional machine learning approaches that require manual feature engineering, CNNs can learn hierarchical representations directly from raw data, enabling them to identify complex patterns and structures. This capability makes CNNs highly effective for airborne vehicle detection, where features such as vehicle shape, size, and orientation must be distinguished from complex backgrounds (Bakirci et al., 2024). The integration of CNNs into object detection systems has revolutionized video analysis, making it a foundational technology for intelligent transportation systems and military targeting. In intelligent transportation systems, the ability to detect and track vehicles in real time facilitates dynamic traffic management, reduces bottlenecks, and improves road safety (Alsubaei et al., 2022). By utilizing aerial imagery, traffic authorities can gain a comprehensive view of road networks, identify congestion areas, and allocate resources more effectively. For example, CNN-based detectors can classify vehicles and predict their trajectories, providing actionable insights for traffic flow optimization. These systems also support autonomous vehicle navigation by enabling real-time obstacle detection and collision prevention. The implementation of CNNs in this context highlights their role in enhancing the efficiency and safety of transportation infrastructures (Kaya et al., 2023).

In military operations, object detection plays a vital role in enhancing BSA by providing real-time insights into the battlefield environment. Accurate detection and tracking of enemy vehicles, personnel, and other assets are crucial for mission planning and execution. CNNs have proven effective in this field due to their ability to process large amounts of data from various sensor modalities, including optical, infrared, and radar imaging (Bakirci & Cetin, 2023). By analyzing aerial imagery using CNNs, military forces can identify potential threats, assess enemy movements, and optimize resource allocation. Furthermore, the end-to-end learning capability of CNNs enables the development of automated systems that can detect and classify objects with minimal human intervention, reducing response times and improving operational efficiency. This capability is particularly valuable in high-risk scenarios where timely and accurate information is critical for decision-making.

One of the key advantages of using CNNs for object detection is their scalability and adaptability. Modern CNN architectures, such as Faster R-CNN (Xu et al., 2022), YOLO (You Only Look Once) (Zhang et al., 2022; Terven et al., 2023), and SSD (Single Shot MultiBox Detector) (Bakirci & Bayraktar, 2024), are designed to balance accuracy and speed to address trade-offs in object detection tasks. For instance, YOLO uses a regression-based approach to directly predict object bounding boxes and class probabilities from feature maps, significantly reducing detection time. While this approach sacrifices some accuracy compared to region-based methods, it provides a practical solution for applications requiring real-time performance. In contrast, architectures like Faster R-CNN focus on maximizing detection accuracy by generating region proposals and refining them through multiple processing stages. This flexibility allows CNN-based systems to be adapted to specific application requirements, such as prioritizing speed for real-time traffic monitoring or accuracy for military surveillance.

The importance of object detection extends beyond its immediate applications, as it serves as a foundation for broader advancements in artificial intelligence and autonomous systems. For instance, in the development of unmanned aerial vehicles (UAVs), object detection is a critical component for navigation, obstacle avoidance, and mission execution (Tang et al., 2024). By incorporating CNNs, UAVs can autonomously identify and track objects of interest, enabling them to perform complex tasks such as search and rescue operations, infrastructure inspections, and environmental monitoring (Ozturk et al., 2021; Bakirci & Bayraktar, 2024). In the context of military applications, UAVs equipped with CNN-based object detection systems can provide continuous surveillance and reconnaissance capabilities, enhancing situational awareness and reducing risks for human personnel. The versatility of CNNs in processing diverse datasets and adapting to varying operational conditions highlights their value in advancing UAV technology and expanding potential use cases (Hong et al., 2019).

Despite their numerous advantages, CNN-based object detection systems face several challenges that must be addressed to fully realize their potential. One such challenge is the computational cost associated with training and deploying deep learning models. High-resolution aerial imagery requires significant processing power and memory, which can strain hardware resources and limit the scalability of CNN-based systems (Zhang et al., 2020). To mitigate this issue, researchers are exploring techniques such as model compression, quantization, and hardware acceleration to optimize CNN performance. Another challenge is the need for large, annotated datasets to effectively train CNNs (Wang et al., 2024; Bakirci & Cetin, 2022). In airborne object detection, obtaining labeled data that accurately represents diverse environmental conditions and object types can be labor-intensive and time-consuming. Synthetic data generation and transfer learning have emerged as promising solutions to address these limitations, enabling CNNs to achieve high performance with limited training data. Additionally, ethical considerations and privacy concerns must be taken into account when deploying object detection systems, particularly in civilian applications. The use of aerial photography for traffic monitoring and intelligent transportation systems raises questions about data privacy and the potential misuse of surveillance technologies. Establishing clear regulatory frameworks and implementing robust data protection measures are essential to address these concerns. Ensuring transparency and accountability in the design and deployment of object detection systems will be critical for gaining public trust and societal acceptance (Oh & Kang, 2017).

In conclusion, object detection is a fundamental technology that is transforming modern applications, particularly in intelligent transportation systems and military operations (Bakirci & Demiray, 2024). By leveraging the power of CNNs, these systems have achieved unprecedented levels of accuracy and efficiency, enabling

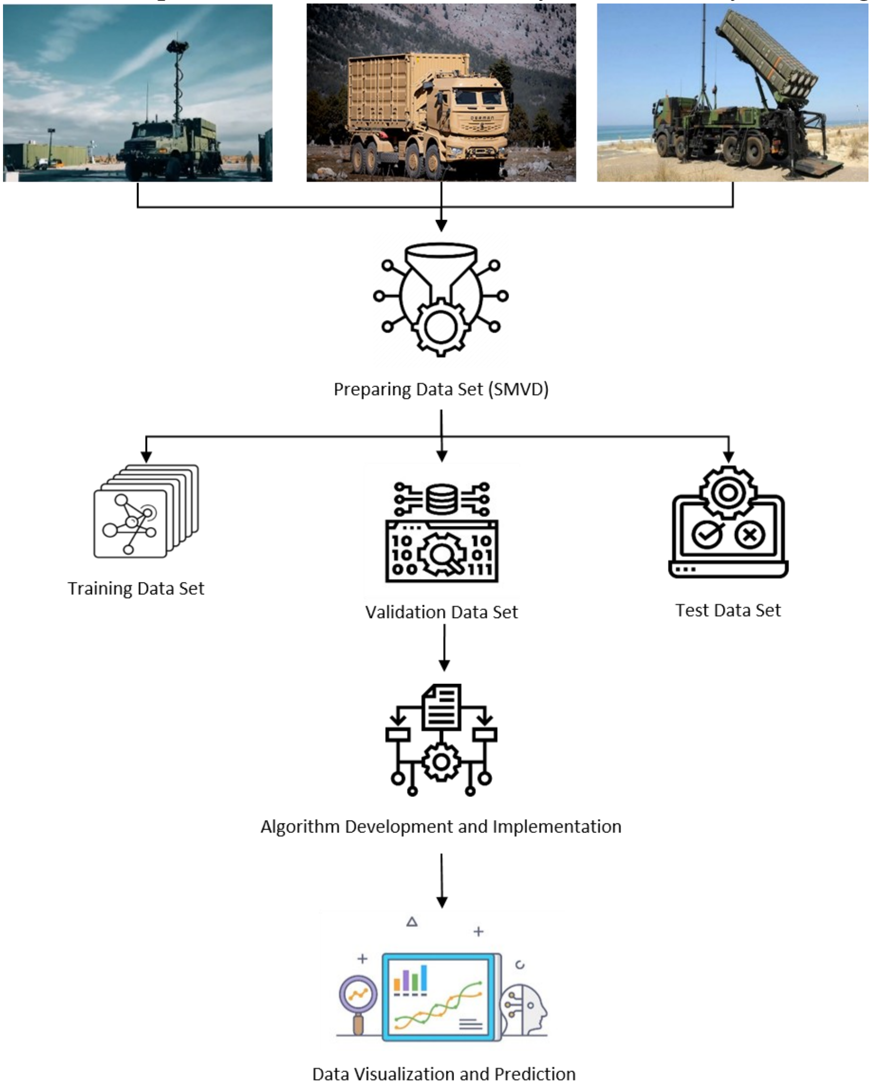


Figure 1: Workflow Diagram

real-time insights and decision-making. From optimizing traffic flow to enhancing battlefield situational awareness, the impact of CNN-based object detection extends across diverse fields, driving progress and innovation. However, addressing challenges related to computational cost, data availability, and ethical considerations will be essential to fully harness this technology's potential and ensure its responsible use in the future.

2. Method

This study investigates an application of the YOLOv8 model in military vehicle detection. The YOLOv8 model has a backbone and neck network for feature extraction and subsequent fusion of the extracted features given an input image. It interprets the semantic information from the extracted features and gives the location, size and class of the detected object in the image. Figure 1 shows the workflow diagram of the prepared study.

2.1. Object Detection Based on the YOLOv8 Model

YOLOv8 is the latest version of the You Only Look Once (YOLO) family and stands out as an advanced model, particularly in the field of object detection (Lou et al., 2023; Wang et al., 2023; Bakirci & Bayraktar, 2024). The primary goal of the YOLO series is to detect and classify objects in images quickly and accurately. YOLOv8 incorporates a series of innovations and improvements compared to previous versions to achieve this goal with higher accuracy, speed, and efficiency. The model typically uses deep learning techniques to classify objects in an image while simultaneously detecting their locations.

YOLOv8 is fundamentally based on a Convolutional Neural Network (CNN) architecture. In the first stage, the network uses a series of convolutional layers that divide the input image into smaller parts. These layers are designed to extract low-level features (such as edges and corners). This process enables the extraction of higher-level features in deeper layers. One of the key innovations of YOLOv8 lies in the optimizations designed to make this network more efficient. For example, the sparse convolution techniques used in the model reduce computational costs while increasing accuracy.

In YOLOv8, the network architecture is structured with "backbone" and "neck" sections. The backbone is the part responsible for extracting essential features from the image and is often supported by robust structures such as ResNet or EfficientNet (Bakirci & Bayraktar, 2024). The neck section, on the other hand, transforms these extracted features into higher-level abstractions, producing the outputs necessary for final detection. This stage utilizes structures like Feature Pyramid Networks (FPN) (Li et al., 2023), which help to more effectively detect objects of various sizes. Thanks to this feature, YOLOv8 can successfully detect both small and large objects. The model's output includes the class, confidence scores, and location information for each detected object.

YOLOv8 also adopts an "anchor-free" approach (Niu et al., 2024), offering a more flexible and precise method for object detection. This allows object boundaries to be determined more accurately, enabling the model to operate with fewer errors compared to previous YOLO versions. Additionally, YOLOv8 improves its overall accuracy through data augmentation techniques and regularization methods used during training. Techniques such as "mixup" allow the model to generalize better.

In terms of speed, YOLOv8 is highly suitable for real-time applications due to its optimized network structure. As a result, YOLOv8 offers significant improvements in object detection tasks, enabling faster and more accurate results (Bakirci & Bayraktar, 2024). By leveraging modern deep learning approaches and network architecture optimizations, YOLOv8 has secured an important position in the field of visual perception.

2.2. Advantages of Using the YOLOv8 Model

The YOLOv8 model (Dewi et al., 2024) holds significant importance in military vehicle detection tasks due to its ability to provide fast and accurate object detection, which plays a critical role, especially in real-time applications. Since military operations often involve time-sensitive and high-precision situations, accurately detecting military vehicles from images is vital for the success of operations. YOLOv8, with its fast processing capability, can instantly detect objects even in large-scale images, thereby accelerating decision-making processes. Moreover, the model's robust network structure and optimization techniques ensure accurate detection of military vehicles even under varying weather conditions and environmental challenges. YOLOv8's enhanced architecture offers the ability to detect both small and large objects simultaneously. This feature accounts for the different sizes of military vehicles and various detection challenges. The model's anchor-free structure allows for more precise boundary determination of objects and accurately detects military vehicles that overlap with complex terrains or natural obstacles. In addition, YOLOv8's low computational cost and high efficiency provide the capability to process data at high speeds in field applications, thereby increasing the effectiveness of critical tasks such as military vehicle detection. These features make YOLOv8 an indispensable tool for military vehicle detection (Bakirci & Bayraktar, 2024).

2.3. Dataset Creation

Due to the confidentiality of military studies, the dataset used in this study could not be homemade. Instead, publicly available and comprehensive datasets such as COCO and KITTI were utilized, which provide satisfactory results for the common vehicle target detection task. In previous research on military object detection, due to the lack of sufficient military datasets, scientists often had to create their own datasets. However, these custom datasets typically include a wide variety of military symbols and may not be entirely suitable for the military vehicle detection task focused on in this study. Therefore, to support this study, a new Special Military Vehicle Dataset (SMVD) has been created.

The SMVD developed in this study covers four common types of military vehicles: tanks, military trucks, infantry fighting vehicles, and command vehicles. These vehicles were obtained from various combat conditions such as desert, grassland, snow, and urban environments. The SMVD consists of 4,575 images of military vehicles in PNG format, with a resolution of 640×640 pixels. All images have been systematically numbered. Subsequently, using the LabelImg software, military vehicles in these images were annotated, resulting in a total of 8,586 accurate labels. To train and test detection models, 58% of the images were randomly selected for the training set, 19% for validation, and the remaining 23% were designated as the test set.

3. Results

Figure 2 presents selected visual examples of the detection results. The YOLOv8 model demonstrated strong performance in detecting tanks, with a precision score of 0.877, indicating its high ability to correctly identify tank instances without mislabeling other objects as tanks. The recall score of 0.819 signifies that the model successfully detected 81.9% of the actual tanks present in the test dataset, showcasing its effective sensitivity. The mean Average Precision (mAP) score of 0.854 further confirms that the model maintains high detection accuracy across varying confidence thresholds. The F1 score of 0.847, a harmonic mean of precision and recall, reflects the model's balanced capability in minimizing false positives and false negatives. Lastly, the detection time of 49 ms per frame underscores the model’s potential for near real-time tank detection, making it suitable for applications requiring quick response times, such as battlefield surveillance and reconnaissance missions. For military trucks, the model exhibited a precision of 0.822, slightly lower than that for tanks, suggesting a marginally increased likelihood of false positives. However, the recall score of 0.845 highlights the model's strong ability to identify most military trucks present in the dataset. The mAP score of 0.830 indicates consistent detection performance across varying thresholds, while the F1 score of 0.833 confirms a balanced trade-off between precision and recall. The detection time for military trucks was recorded at 59 ms per frame, slightly higher than for tanks, likely reflecting the increased complexity or diversity in the appearance of military trucks. Despite this, the model's performance remains highly efficient for real-time military truck detection scenarios.

Detection results for infantry fighting vehicles (IFVs) were comparatively weaker than for other vehicle types. A precision score of 0.779 points to a moderate likelihood of misclassifying other objects as IFVs. The recall score of 0.708 suggests that the model detected approximately 70.8% of the IFVs present in the dataset, indicating room for improvement in sensitivity. The mAP score of 0.713 confirms the model's relatively lower accuracy across confidence thresholds for this vehicle type. Despite these limitations, the F1 score of 0.742 shows a reasonable balance between precision and recall. The detection time of 67 ms, while higher than for tanks and trucks, remains within an acceptable range for many practical applications. The challenges in detecting IFVs could be attributed to their varied shapes, sizes, or camouflage patterns, highlighting the need for additional training data or algorithm optimization. The YOLOv8 model achieved robust results for military command vehicles, with a precision score of 0.861, indicating a low rate of false positives. The recall score of 0.874 further demonstrates the model's exceptional ability to identify nearly all military command vehicles in the dataset. The mAP score of 0.855 aligns closely with these findings, signifying consistent detection performance across thresholds. The F1 score of 0.867 emphasizes the model's effectiveness in maintaining both high precision and recall for this vehicle category. Additionally, the detection time of 51 ms per frame showcases the model's efficiency, making it well-suited for applications that require prompt identification of command vehicles, such as operational planning and real-time tactical analysis.

Figure 2: Detection results with YOLOv8

3.1. Performance Comparison

The detection performance of YOLOv8 varies noticeably across the four military vehicle categories: tanks, military trucks, IFVs, and military command vehicles. These variations can be attributed to differences in vehicle characteristics, dataset quality, and the inherent challenges of detecting certain features. Tanks exhibited the highest precision (0.877), reflecting the model's ability to accurately differentiate tanks from other objects. This superior performance may be due to the distinct structural features of tanks, such as their turret and caterpillar tracks, which make them visually unique. In contrast, IFVs had the lowest precision (0.779), likely because their designs are more varied and less distinct, leading to a higher probability of false positives. Military trucks (0.822) and command vehicles (0.861) fall between these extremes, suggesting moderate distinctiveness in their visual characteristics. Military command vehicles achieved the highest recall (0.874), indicating the model's ability to detect a large proportion of true instances. This may be due to the dataset containing sufficient examples of command vehicles with consistent visual patterns, such as antennas or specific body shapes, making them easier for the model to recognize. Conversely, IFVs scored the lowest recall (0.708), suggesting that the model struggled to detect many of the IFVs present in the dataset. The variability in IFV design, including differences in size and camouflage, might have contributed to this reduced sensitivity. Tanks (0.819) and military trucks (0.845) showed comparable recall scores, reflecting a balance between dataset representation and visual features. The mAP scores followed a similar trend, with tanks (0.854) and command vehicles (0.855) achieving the highest values, indicating consistent detection accuracy across confidence thresholds. Military trucks (0.830) showed slightly lower mAP, while IFVs (0.713) lagged significantly. The lower mAP for IFVs suggests that the model struggled to consistently detect this category across a range of scenarios and confidence levels, reinforcing the hypothesis of higher visual variability and less distinct features. The F1 score provides a balanced view of precision and recall. Military command vehicles (0.867) and tanks (0.847) achieved the highest F1 scores, underscoring the model's well-rounded performance for these categories. Military trucks (0.833) demonstrated slightly lower F1 scores due to reduced precision, while IFVs (0.742) had the lowest F1 score, reflecting significant challenges in both false positive and false negative rates. Detection times were fastest for tanks (49 ms) and command vehicles (51 ms), suggesting that these categories may have simpler or more distinctive features that the model processes efficiently. Military trucks (59 ms) and IFVs (67 ms) required longer detection times, likely due to increased visual complexity or intra-class variability. The higher detection time for IFVs in particular could indicate that the model is expending more computational effort to differentiate these vehicles from background objects or other categories.

The distinct structural features of tanks and command vehicles likely contributed to their higher precision, recall, and F1 scores. Features like the turret and tracks in tanks or antennas on command vehicles allow the model to identify these categories with minimal confusion. Conversely, IFVs' less distinct and more variable designs reduce the model's ability to differentiate them from other objects, negatively impacting precision and recall. The higher performance for command vehicles and tanks may also reflect better representation of these categories in the training dataset. Well-annotated, diverse, and adequately sized datasets can significantly enhance model learning and generalization. The lower performance for IFVs suggests potential gaps in the dataset, such as insufficient examples of IFVs or examples that fail to capture their diversity adequately. The lower detection performance for IFVs may also be attributed to the visual complexity and use of camouflage patterns, which make them harder to distinguish from backgrounds. Military trucks, while not as challenging as IFVs, still displayed slightly reduced precision and mAP due to their variable shapes and sizes. The differences in detection time suggest that the model requires varying levels of computational effort to process different vehicle categories. The higher detection time for IFVs may indicate that the model is performing additional calculations to resolve ambiguities caused by visual complexity.

4. Conclusion

This study evaluated the performance of the YOLOv8 object detection algorithm for identifying four distinct categories of military vehicles: tanks, military trucks, infantry fighting vehicles (IFVs), and military command vehicles. The results demonstrate that YOLOv8 achieves high detection accuracy for vehicle types with distinctive and consistent visual features, such as tanks and military command vehicles, as evidenced by their high precision, recall, and F1 scores. However, the model's performance declines for categories like IFVs, which exhibit greater visual variability and complexity, underscoring the challenges posed by less distinctive features and camouflage patterns. The findings also reveal differences in detection times across categories, with tanks and command vehicles benefiting from faster inference times compared to IFVs and military trucks. These variations suggest that the computational effort required for detection is influenced by the visual complexity and intra-class variability of the objects being detected. Overall, this study highlights YOLOv8's capability as a reliable tool for military vehicle detection, particularly in applications requiring real-time performance, such as battlefield reconnaissance and surveillance. However, it also underscores the need for further optimization, particularly for detecting visually complex or poorly represented vehicle categories. Future work could focus on augmenting training datasets with diverse examples, applying domain-specific transfer learning techniques, and fine-tuning the algorithm to enhance detection performance for challenging object categories. These insights contribute to advancing the application of deep learning-based object detection in military operations, paving the way for more robust and efficient surveillance systems.

5. References

Ammour, N., Alhichri, H., Bazi, Y., Benjdira, B., Alajlan, N., & Zuair, M. (2017). Deep Learning Approach for Car Detection in UAV Imagery. Remote Sensing, 9(4), 312. https://doi.org/10.3390/rs9040312

Nellore, K., & Hancke, G. P. (2016). A Survey on Urban Traffic Management System Using Wireless Sensor Networks. Sensors, 16(2), 157. https://doi.org/10.3390/s16020157

Bakirci, M., Bayraktar, I. (2024). Improving coastal and port management in smart cities with UAVs and deep learning. 2024 Mediterranean Smart Cities Conference (MSCC), pp. 1-6, Martil - Tetuan, Morocco. https://doi.org/10.1109/MSCC62288.2024.10697069

Krichen, M. (2023). Convolutional Neural Networks: A Survey. Computers, 12(8), 151. https://doi.org/10.3390/computers12080151

Veeranampalayam Sivakumar, A. N., Li, J., Scott, S., Psota, E., J. Jhala, A., Luck, J. D., & Shi, Y. (2020). Comparison of Object Detection and Patch-Based Classification Deep Learning Models on Mid- to Late-Season Weed Detection in UAV Imagery. Remote Sensing, 12(13), 2136. https://doi.org/10.3390/rs12132136

Bakirci, M., Dmytrovych, P., Bayraktar, I., Anatoliyovych, O. (2024). Challenges and advances in UAV-based vehicle detection using YOLOv9 and YOLOv10. 2024 IEEE 7th International Conference on Actual Problems of Unmanned Aerial Vehicles Development (APUAVD), pp. 317-321, Kyiv, Ukraine. https://doi.org/10.1109/APUAVD64488.2024.10765874

Alsubaei, F. S., Al-Wesabi, F. N., & Hilal, A. M. (2022). Deep Learning-Based Small Object Detection and Classification Model for Garbage Waste Management in Smart Cities and IoT Environment. Applied Sciences, 12(5), 2281. https://doi.org/10.3390/app12052281

Kaya, Ö., Çodur, M. Y., & Mustafaraj, E. (2023). Automatic Detection of Pedestrian Crosswalk with Faster R-CNN and YOLOv7. Buildings, 13(4), 1070. https://doi.org/10.3390/buildings13041070

Bakirci, M., Cetin, M. (2023). Reliability of MEMS accelerometers embedded in smart mobile devices for robotics applications. In: Garcia Marquez, F.P., Jamil, A., Eken, S., Hameed, A.A. (eds) Computational Intelligence, Data Analytics and Applications. ICCIDA 2022. Lecture Notes in Networks and Systems, vol 643, pp. 78-90, Springer, Cham. https://doi.org/10.1007/978-3-031-27099-4_7

Xu, X., Zhao, M., Shi, P., Ren, R., He, X., Wei, X., & Yang, H. (2022). Crack Detection and Comparison Study Based on Faster R-CNN and Mask R-CNN. Sensors, 22(3), 1215. https://doi.org/10.3390/s22031215

Zhang, Y., Guo, Z., Wu, J., Tian, Y., Tang, H., & Guo, X. (2022). Real-Time Vehicle Detection Based on Improved YOLO v5. Sustainability, 14(19), 12274. https://doi.org/10.3390/su141912274

Terven, J., Córdova-Esparza, D. -M., & Romero-González, J. -A. (2023). A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS. Machine Learning and Knowledge Extraction, 5(4), 1680-1716. https://doi.org/10.3390/make5040083

Bakirci, M., Bayraktar, I. (2024). Integrating UAV-based aerial monitoring and SSD for enhanced traffic management in smart cities. 2024 Mediterranean Smart Cities Conference (MSCC), pp. 1-6, Martil - Tetuan, Morocco. https://doi.org/10.1109/MSCC62288.2024.10696996

Tang, G., Ni, J., Zhao, Y., Gu, Y., & Cao, W. (2024). A Survey of Object Detection for UAVs Based on Deep Learning. Remote Sensing, 16(1), 149. https://doi.org/10.3390/rs16010149

Öztürk, A. E., & Erçelebi, E. (2021). Real UAV-Bird Image Classification Using CNN with a Synthetic Dataset. Applied Sciences, 11(9), 3863. https://doi.org/10.3390/app11093863

Bakirci, M., Bayraktar, I. (2024). Harnessing UAV technology and YOLOv9 algorithm for real-time forest fire detection. 2024 International Russian Automation Conference (RusAutoCon), pp. 95-100, Sochi, Russian Federation. https://doi.org/10.1109/RusAutoCon61949.2024.10694663

Hong, S. -J., Han, Y., Kim, S. -Y., Lee, A. -Y., & Kim, G. (2019). Application of Deep-Learning Methods to Bird Detection Using Unmanned Aerial Vehicle Imagery. Sensors, 19(7), 1651. https://doi.org/10.3390/s19071651

Zhang, N., Wei, X., Chen, H., & Liu, W. (2021). FPGA Implementation for CNN-Based Optical Remote Sensing Object Detection. Electronics, 10(3), 282. https://doi.org/10.3390/electronics10030282

Wang, B., Chen, Y., Yan, Z., & Liu, W. (2024). Integrating Remote Sensing Data and CNN-LSTM-Attention Techniques for Improved Forest Stock Volume Estimation: A Comprehensive Analysis of Baishanzu Forest Park, China. Remote Sensing, 16(2), 324. https://doi.org/10.3390/rs16020324

Bakirci, M., Cetin, M. (2022). Utilization of a vehicle's on-board diagnostics to reduce GPS-sourced positioning error. Innovations in Intelligent Systems and Applications Conference. September 7-9, Antalya, Turkey. https://doi.org/10.1109/ASYU56188.2022.9925443

Oh, S. -I., & Kang, H. -B. (2017). Object Detection and Classification by Decision-Level Fusion for Intelligent Vehicle Systems. Sensors, 17(1), 207. https://doi.org/10.3390/s17010207

Bakirci, M., Demiray, A. (2024). Enhancing attitude control in space stations through integrated robotic systems. 2024 International Russian Automation Conference (RusAutoCon), pp.

144-149, Sochi, Russian Federation. https://doi.org/10.1109/RusAutoCon61949.2024.10694461

Lou, H., Duan, X., Guo, J., Liu, H., Gu, J., Bi, L., & Chen, H. (2023). DC-YOLOv8: Small-Size Object Detection Algorithm Based on Camera Sensor. Electronics, 12(10), 2323. https://doi.org/10.3390/electronics12102323

Wang, G., Chen, Y., An, P., Hong, H., Hu, J., & Huang, T. (2023). UAV-YOLOv8: A Small-Object-Detection Model Based on Improved YOLOv8 for UAV Aerial Photography Scenarios. Sensors, 23(16), 7190. https://doi.org/10.3390/s23167190

Bakirci, M., Bayraktar, I. (2024). Transforming aircraft detection through LEO satellite imagery and YOLOv9 for improved aviation safety. 2024 26th International Conference on Digital Signal Processing and its Applications (DSPA), pp. 1-6. March 27-29, Moscow, Russian Federation. https://doi.org/10.1109/DSPA60853.2024.10510106

Li, Y., Fan, Q., Huang, H., Han, Z., & Gu, Q. (2023). A Modified YOLOv8 Detection Network for UAV Aerial Image Recognition. Drones, 7(5), 304. https://doi.org/10.3390/drones7050304

Niu, Y., Cheng, W., Shi, C., & Fan, S. (2024). YOLOv8-CGRNet: A Lightweight Object Detection Network Leveraging Context Guidance and Deep Residual Learning. Electronics, 13(1), 43. https://doi.org/10.3390/electronics13010043

Bakirci, M., Bayraktar, I. (2024). YOLOv9-enabled vehicle detection for urban security and forensics applications. 2024 12th International Symposium on Digital Forensics and Security (ISDFS), pp. 1-6, San Antonio, TX, USA. https://doi.org/10.1109/ISDFS60797.2024.10527304

Dewi, C., Manongga, D., Hendry, Mailoa, E., & Hartomo, K. D. (2024). Deep Learning and YOLOv8 Utilized in an Accurate Face Mask Detection System. Big Data and Cognitive Computing, 8(1), 9. https://doi.org/10.3390/bdcc8010009

Bakirci, M., Bayraktar, I. (2024). Boosting aircraft monitoring and security through ground surveillance optimization with YOLOv9. 2024 12th International Symposium on Digital Forensics and Security (ISDFS), pp. 1-6, San Antonio, TX, USA. https://doi.org/10.1109/ISDFS60797.2024.10527349

CHAPTER II

Investigating the Impact of Lighting Conditions on Crack Detection Algorithms in Mechanical Systems

Fikrican Ciftci[1]
Furkan Yildirim[2]
Bora Burak Azer[3]

1. Introduction

The reliability of mechanical components used in factories producing military vehicles is critically important for the performance and operational safety of these vehicles. In military applications, the structural integrity of these components is an indispensable requirement to withstand the harsh conditions encountered in the field (Yuan et al., 2019). However, microcracks or other structural defects that may develop over time in mechanical parts can lead to performance degradation in the short term and catastrophic failures in the long term (Wang et al., 2020). Therefore,

[1] Fikrican Ciftci, Koluman Automotive Industry Inc., R&D Department, Ankara/Turkiye, Orcid: 0009-0004-0538-4995, fikrican.ciftci@koluman.com
[2] Furkan Yildirim, Koluman Automotive Industry Inc., R&D Department, Ankara/Turkiye, Orcid: 0009-0009-5478-3008, furkan.yildirim@koluman.com
[3] Bora Burak Azer, Koluman Automotive Industry Inc., R&D Department, Ankara/Turkiye, Orcid: 0009-0004-3863-1526, burak.azer@koluman.com

developing fast, precise, and reliable methods for detecting cracks in mechanical components during production and maintenance processes is of great importance. In this context, artificial intelligence-based approaches, such as object detection algorithms, offer a more effective and efficient solution compared to traditional methods (Lou et al., 2024).

Traditional crack detection methods often rely on techniques such as ultrasonic testing, magnetic particle inspection, or visual inspection. While these methods can be effective, they typically require intensive human labor, and the error rate increases due to human factors. In visual inspection methods, factors such as the attention level, experience, and fatigue of operators tasked with detecting cracks directly influence the accuracy of the detection process (Rajesh et al., 2024). For instance, a small crack overlooked by an operator may grow over time, leading to irreversible damage. Furthermore, the high number of parts on the production line and the time constraints of the inspection process increase the likelihood of errors made by operators. These limitations associated with human factors clearly highlight the need for automated and intelligent systems, such as object detection algorithms (Bakirci et al., 2024).

Object detection algorithms can detect cracks on the surface of mechanical components with high accuracy using image processing and deep learning techniques (Ma et al., 2018). These algorithms analyze surface images of parts to quickly identify the size, shape, and location of cracks. Particularly, deep learning models like convolutional neural networks (CNN) (Bakirci & Bayraktar, 2024) are highly successful in distinguishing the complex and low-contrast visual features of cracks. By leveraging high-resolution images and annotations as training data, the sensitivity of these algorithms to crack detection can be continuously improved. Moreover, the ability of these systems to operate automatically minimizes human intervention, reduces error rates, and ensures more consistent processes (Fan et al., 2020).

In addition to reducing error rates, the integration of object detection algorithms also saves time and costs in production processes. For example, while manual inspection of each mechanical component on a production line can take hours, automated systems can complete this task in seconds (Munawar et al., 2021). Furthermore, the ability to generate numerical reports on detected defects makes quality control processes more objective and standardized. This allows for regular monitoring of crack size and growth trends, enabling more effective planning of preventive maintenance strategies.

In conclusion, the detection of cracks in mechanical components using object detection algorithms in factories producing military vehicles enhances reliability and efficiency by mitigating human error risks (Maslan & Cicmanec, 2023). These methods, which overcome the limitations of human observation, play a crucial role not only in defect detection but also in the overall optimization of production and maintenance processes. Such artificial intelligence-based solutions have the potential to enhance the field performance of military vehicles while minimizing production costs and time loss. Therefore, the broader adoption of object detection algorithms in crack detection has become a strategic necessity for the modern defense industry.

Deep learning-based object detection algorithms, particularly models like Convolutional Neural Networks (CNN) and the R-CNN family, have revolutionized image analysis and object detection (Bakirci & Bayraktar, 2024). Two-stage object detection detectors offer significant advantages, especially in applications requiring high accuracy. These detectors primarily identify potential object regions (region proposals) in the first stage and perform a more detailed analysis in the second stage to classify and localize the objects. Models like R-CNN (Region-based CNN), Fast R-CNN, Faster R-CNN, and Mask R-CNN are leading examples of this approach (Lee & Park, 2022; Afzaal et al., 2021). The R-CNN family introduces innovative approaches to improve detection accuracy. Initially, R-CNN extracts features from the selected regions using CNNs and classifies these regions. While this method provides high accuracy, it has a longer processing time due to running separate CNNs for each region. Fast R-CNN accelerates this process by extracting a feature map from the entire image using a single CNN and analyzing region proposals on this map. This reduces computational cost and increases the model's accuracy (Ren et al., 2018). Taking it further, Faster R-CNN introduces Region Proposal Networks (RPN), making the process of identifying potential object regions entirely learnable. This innovation makes two-stage detectors more suitable for real-time applications.

The greatest advantage of two-stage detectors is their ability to detect small, complex, and overlapping objects with high accuracy. The focus on potential object regions in the first stage prevents unnecessary processing of irrelevant areas, allowing more resources to be allocated for analysis. These models excel in scenarios requiring high accuracy, such as defense, medical, and industrial automation. Additionally, models like Mask R-CNN provide object segmentation capabilities, enabling pixel-level differentiation of objects, expanding the scope of two-stage detectors to applications requiring detailed analysis. Consequently, two-stage object detection detectors based on CNN and the R-CNN family are powerful tools in object detection due to their high accuracy, flexibility, and detailed analysis capabilities (Bakirci & Bayraktar, 2024). In complex scenes, these models' precision offers performance that surpasses traditional methods.

2. Methodology

With the increasing interest in deep learning, significant developments and advancements have been achieved in the field of object detection. Among these advancements, Faster R-CNN, a member of the widely popular R-CNN family, stands out as a convolutional neural network (CNN)-based object detection algorithm. Faster R-CNN distinguishes itself from previous-generation algorithms by producing faster and more accurate results, bringing it to the forefront (Avola et al., 2021). It is particularly preferred for its ability to combine object detection and classification functions, achieving both high accuracy and fast operation. The primary goal of the Faster R-CNN algorithm is not only to detect objects and details in image data but also to develop an integrated network architecture that accurately localizes these objects and details (Zhang et al., 2019).

Faster R-CNN significantly improves its speed and accuracy and reduces overall system complexity with the introduction of the Region Proposal Network (RPN), a key innovation absent in previous CNN architectures and the Fast R-CNN algorithm. This enhancement makes it superior to other algorithms within the R-CNN family. Faster R-CNN, which addresses and improves upon the shortcomings of its predecessors, R-CNN and Fast R-CNN, consists of several core components and convolutional layers. These components and layers include Feature Extraction (Backbone Network), Region Proposal Network (RPN), ROI Pooling Layer, Classification, and Localization Phases (Bakirci & Bayraktar, 2024). The integration of these interrelated components and convolutional layers enables Faster R-CNN to provide rapid and accurate object detection capabilities. Among these, the Region Proposal Network and the Fast R-CNN detector stand out as the primary components.

As shown in Figure 1, the input image passes through various layers of Faster R-CNN, and the desired object in the image is detected. The working principle of Faster R-CNN can be examined under four main headings:

1. Feature Extraction (Backbone Network): Faster R-CNN utilizes the CNN architecture for feature extraction. The primary purpose of feature extraction is to identify and transform the significant and meaningful features of an image, enabling the model to better recognize and distinguish objects. The backbone network processes the low-level pixel data of an image step by step, transforming it into higher-level representations that are more meaningful for object recognition. After the input image is processed by the CNN architecture, a feature map is generated.

2. Region Proposal Network (RPN): The RPN is a component that makes strong predictions about where objects might be located within an image. The primary task of the RPN is to generate thousands of region proposals that potentially contain objects within an image. These proposals are represented by bounding rectangular frames of varying sizes and aspect ratios. Subsequently, the RPN predicts the likelihood of each frame containing an object and estimates how well it matches the actual object.

3. ROI Pooling Layer: The ROI pooling layer plays a critical role in classifying the regions proposed by the RPN during the object detection process. It reduces region proposals of varying sizes to a fixed-size matrix, standardizing each region proposal. The main advantage of the ROI pooling layer is its ability to simplify the network's complexity, making the classification process faster and more efficient.

4. Classification Phase: Each region proposal generated by the RPN is classified into its respective object class. The goal of this

step is to ensure that the identified objects are accurately categorized, making the process faster and more efficient. At the end of the classification phase, probability values are calculated for each proposed region, and confidence scores are generated for their respective object classes.

5. Localization Phase: The localization phase refines the bounding boxes determined by the RPN to improve their accuracy. This involves adjusting the center points, widths, and heights of the bounding boxes to ensure a more precise fit around the objects. As a result, more accurately framed bounding boxes are produced during the object detection process.

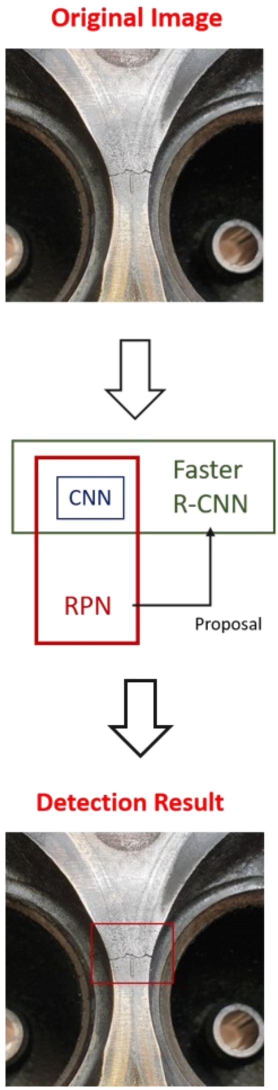
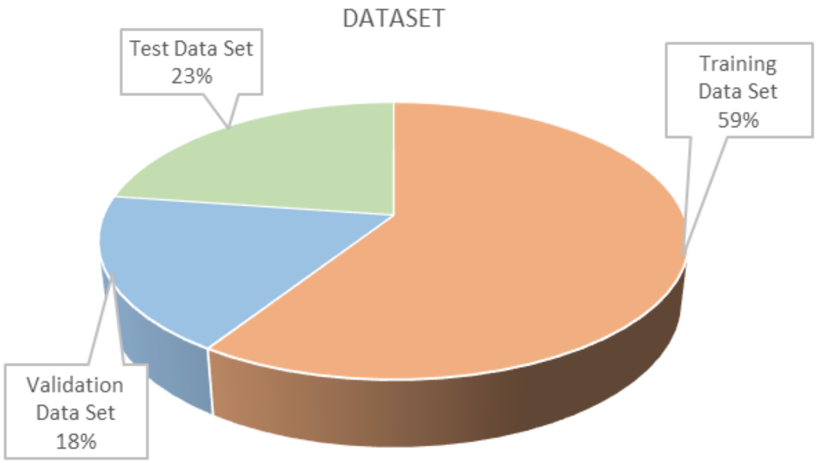


Figure 1: Faster R-CNN detection process.

3. Dataset



Graph 1: Weighted distributions of the dataset for training, validation, and testing phases. A total of 59% of the dataset is allocated for training, 18% for validation, and 23% for testing.

In this study on crack detection in mechanical components, a meticulous approach was adopted to establish a reliable and comprehensive dataset. Initially, the experimental data collection process involved the use of mechanical components both with and without cracks. These components were subjected to various loading conditions, and the resulting cracks were detected using imaging techniques, particularly high-resolution cameras and thermal imaging systems. Images were captured from different angles and under varying lighting conditions to enhance diversity.

The raw data obtained were processed to improve the accuracy and diversity of the dataset. However, since only a limited amount of data could be collected under laboratory conditions, data augmentation techniques were employed to increase the size and variety of the dataset. During the data augmentation process, the original images were subjected to transformations such as rotation, cropping, brightness and contrast adjustments, blurring, and noise addition. Additionally, for symmetrical components, reflection and scaling methods were applied to ensure spatial diversity in the images. This approach enabled the model to effectively learn under different conditions and crack locations.

As a result, the constructed dataset provided a robust foundation for training a deep learning model focused on crack detection in mechanical components. The dataset aimed to better reflect real-world scenarios and enhance the model's generalization capacity. The final dataset consists of a total of 1,875 images. The weight distribution for the training, validation, and testing phases of the dataset is presented in Graph 1.

4. Data Augmentation

Image processing techniques play a critical role in materials engineering, quality control, damage detection, and various research fields. In this study, different image processing techniques were applied to an original image of a material surface and analyzed in detail. Examining visual data in depth is crucial for detecting cracks and structural defects on surfaces. The original image, labeled as Image 1, was analyzed first, followed by adjustments such as brightness, contrast, sharpness, and color transformations. The effects of these operations are thoroughly examined below.

The original image, represented as Image 1, shows a material surface with a visible crack. This image serves as the reference point for analysis, as no processing has been applied, and it has been evaluated in its raw form. While the surface texture and crack regions are perceivable to the naked eye in the original image, further image processing techniques were applied to make these details clearer and more prominent. In Image 2, the brightness was increased by 50%. This operation enhanced the visibility of darker areas in the image but resulted in some loss of detail. Excessive brightness can cause information loss, especially in lighter parts of the surface. In Image 3, the brightness was reduced by 50%, darkening the image. While this enhanced details in the darker regions of the surface, it reduced overall visibility. In this case, the crack regions became more pronounced, but understanding the overall surface structure became more challenging.

In Image 4, the sharpness was increased by 100%. This operation highlighted fine details on the surface, especially the crack edges, making them more distinct. Increasing sharpness improved the analyzability of the image and made the surface texture details more visible. However, excessive sharpness can sometimes introduce artifacts or artificial errors into the image. When sharpness was reduced by 100%, as seen in Image 7, blurring occurred. In this case, surface details and crack visibility decreased significantly. Reduced sharpness can make analysis difficult, particularly for low-resolution images.

In Image 5, contrast was increased by 50%. Enhancing the contrast emphasized the difference between light and dark areas on the surface, making the cracks more visible. The tonal difference around the crack became sharper with this process. Conversely, when contrast was reduced by 50%, as shown in Image 6, the difference between light and dark areas diminished, reducing the visibility of cracks on the surface. The image became more homogeneous, but detail loss occurred.

In Image 8, a grayscale filter was applied, removing color information and converting the image into grayscale. Grayscale conversion is particularly useful for analyses where color information is not important. This operation allows for a better understanding of the surface's intensity distribution. Grayscale conversion provides significant advantages in analyzing cracks and surface texture based on tonal differences.

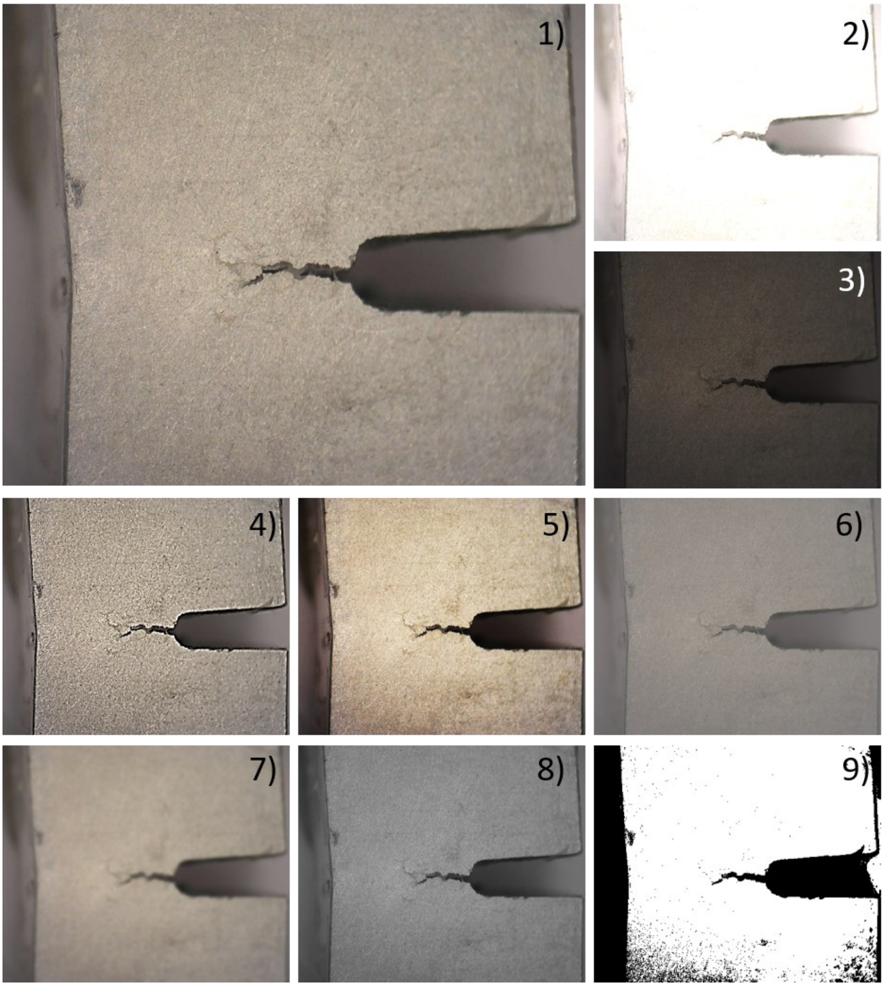


Figure 2: Images of data augmentation using various methods.

Image 9 was created using a combination of thresholding and edge detection techniques. Thresholding classifies pixels as either black or white based on a specific intensity value. This process separates cracks and cut regions from the background. Edge detection, on the other hand, analyzes changes in intensity to define object boundaries. As a result, the edges of the crack region were clearly highlighted, making it easier to analyze structural defects on the surface.

This study examined the effects of different image processing techniques on an image of a material surface. Adjustments such as brightness, contrast, sharpness, and grayscale transformations highlighted surface details or simplified the image for analysis. Finally, thresholding and edge detection techniques provided the most beneficial outcome by defining the precise boundaries of the crack regions. Such image processing methods play a critical role in materials engineering, damage analysis, and quality control applications.

5. Result

Model	Metric				
	Precision	**Recall**	**mAP**	**F1 Score**	**Inference Time**
Faster R-CNN	0.879	0.817	0.803	0.846	53.4 ms
SSD	0.780	0.721	0.737	0.749	24.4 ms
YOLOv5	0.803	0.755	0.776	0.778	21.6 ms

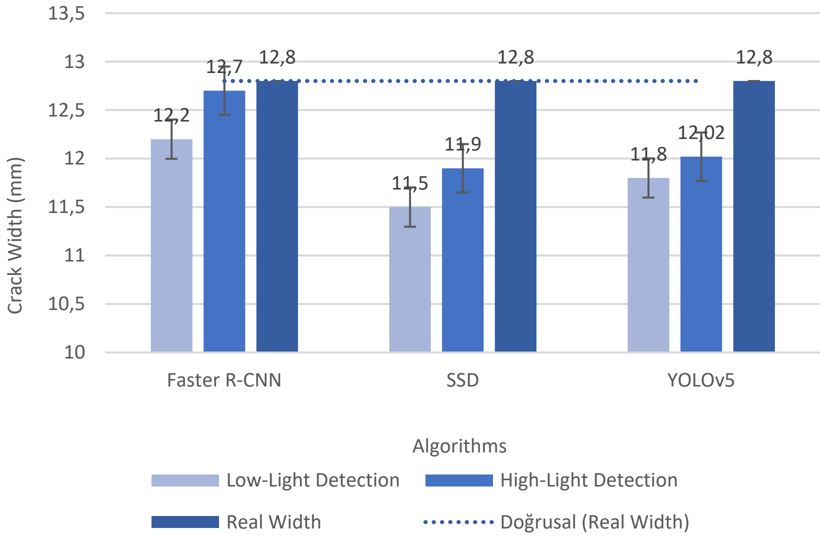
Table 1: Comparison of the Faster R-CNN algorithm with SSD and YOLOv5 in terms of performance metrics.

Table 1 presents the performance metrics of three different deep learning algorithms (Faster R-CNN, SSD, YOLOv5) used for crack detection in mechanical parts. Table 1 includes the accuracy, recall, mean average accuracy (mAP), F1 score, and inference times of these algorithms (Bakirci & Bayraktar, 2024). Precision indicates how many cracks a model detects correctly. In other words, it is the ratio of correctly detected positives (true positives) to the total positive detections (true positives + false positives). A high precision indicates that the model has a low tendency to produce false positives. Recall indicates how many total true cracks the model can detect. In other words, it is the ratio of correctly detected positives (true positives) to the total true positives (true positives + false negatives). A high recall indicates that the model has a low probability of missing positive examples. mAP (Mean Average Precision) measures the average accuracy performance of the model at different threshold values. It is used to understand the impact of the balance between precision and recall values on the overall performance. mAP is a standard often used to compare the performance of object detection models. F1 Score is a balance metric between precision and recall. It expresses the harmonic mean of precision and recall values and focuses on reducing both false positives and false negatives. F1 score provides a more comprehensive evaluation in imbalanced datasets or in cases where precision and recall values differ. Inference Time refers to the time it takes for the model to perform its prediction on an image. Inference time is critical for the usability of the model in real-time applications.

When we first evaluate the performance of the algorithms in terms of precision; Faster R-CNN (0.879) has the highest precision value. This shows that the majority of the cracks it detects are correct. YOLOv5 (0.803) is lower than Faster R-CNN but higher than SSD (0.780). YOLOv5 is better than SSD in limiting false positives. SSD has the lowest precision value compared to the other two algorithms, which means it produces more false positives. Faster R-CNN (0.817) provided the highest recall value by detecting a large portion of the real cracks. YOLOv5 (0.755) is lower than Faster R-CNN but higher than SSD (0.721). This means that YOLOv5 missed fewer cracks. SSD exhibited the lowest performance in terms of recall, missing some of the positive examples. Faster R-CNN (0.803) achieved the highest mAP value and showed the most accurate detection performance overall. YOLOv5 (0.776) outperformed SSD

(0.737) in mAP value and showed a balanced performance. SSD is behind the other two algorithms in mAP, indicating less overall accuracy. Faster R-CNN (0.846) achieved the best balance between precision and recall and achieved the highest F1 score. YOLOv5 (0.778) again showed a more balanced performance with a better F1 score than SSD (0.749). SSD was behind the other two algorithms in F1 score, showing a weaker balance between accuracy and recall. YOLOv5 (21.6 ms) has the fastest inference time and is quite suitable for real-time applications. SSD (24.4 ms) is a bit slower than YOLOv5, although it performs similarly to YOLOv5 in terms of speed. Faster R-CNN (53.4 ms) was considerably slower in inference time compared to the other two algorithms and is limited for real-time applications.

Faster R-CNN achieved the highest values in precision, recall, mAP and F1 scores, and showed superior performance in terms of crack detection accuracy. However, it is the algorithm with the longest inference time (53.4 ms) and its usability in real-time applications is limited. Therefore, it can be preferred in applications that require high accuracy. YOLOv5 provided a balanced performance between speed and accuracy. Since it has the shortest inference time (21.6 ms), it is quite advantageous in real-time applications. It achieved better results compared to SSD in precision, recall, mAP and F1 scores and ranked second in terms of overall performance. YOLOv5 is ideal for scenarios where accuracy and speed requirements are balanced. Although SSD has the lowest values in both accuracy and recall metrics, it provided fast performance with inference time (24.4 ms). It can be used in applications where accuracy tolerance is lower but speed is at the forefront. However, it is at the end of the preference order due to its limited accuracy performance compared to other algorithms. As a result, Faster R-CNN is recommended for situations where precision is critical, YOLOv5 for real-time applications, and SSD for applications requiring low precision where speed is a priority.



Graph 2: Detection size results of a crack width measured as 12.8 mm in a mechanical part obtained with Faster R-CNN, SSD and YOLOv5 models under sufficient and insufficient light conditions.

Graph 2 evaluates the performance of three different algorithms (Faster R-CNN, SSD and YOLOv5) used for crack detection in mechanical parts in low light and high light conditions. The graph compares the width values measured by these algorithms for a crack with an original width of 12.8 mm. At the same time, the accuracy of the measurement results of the algorithms is compared with the real crack width (12.8 mm).

Faster R-CNN measured the crack width as 12.2 mm on average in low light conditions, while this value increased to 12.7 mm in high light conditions. In both cases, some deviation from the real width (12.8 mm) was observed. However, the measurement in high light conditions gave a closer result than in low light conditions. This shows that the algorithm works more accurately in high light conditions.

The SSD algorithm showed a significant deviation from the real width by measuring the crack width as 11.5 mm in low light conditions. This value was measured as 11.9 mm in high light conditions. The deviation of SSD from the true width in both conditions is higher compared to the other algorithms. This shows that SSD is less sensitive than the other two algorithms in measuring the crack width.

YOLOv5 measured the crack width as 11.8 mm in low light conditions, while this value was measured as 12.02 mm in high light conditions. The algorithm gave a result that was quite close to the true width, especially in high light conditions. However, its performance in low light conditions showed a slightly lower accuracy compared to the true width. Nevertheless, YOLOv5's performance showed a better sensitivity compared to SSD.

The graph clearly shows the closeness of the measurement results of all algorithms to the true crack width and the performance changes in different light conditions. Faster R-CNN and YOLOv5 produced results closer to the true width, especially in high light conditions. SSD showed significant deviations in low and high light conditions and showed lower performance compared to the other two algorithms. These results reveal the effect of light conditions on the algorithm performance in applications that require precise measurements such as crack detection. In addition, it is understood that Faster R-CNN and YOLOv5 provide more reliable results in high light conditions, but SSD’s performance is more limited in such cases. In future studies, it can be suggested to use techniques such as data augmentation or different preprocessing methods to improve the performance of the algorithms in low light conditions. This approach can contribute to obtaining more consistent and reliable results in real scenarios.

6. Conclusion

This study demonstrates the strengths and limitations of three prominent object detection algorithms, Faster R-CNN, SSD, and YOLOv5, in the context of crack detection in mechanical components. Faster R-CNN excels in precision, recall, and mAP, making it the most accurate model for applications where detection quality is paramount. However, its slower inference time limits its applicability in real-time scenarios. YOLOv5 emerges as a strong contender, offering the fastest inference time while maintaining a balanced performance across all metrics. It is particularly well-suited for real-time industrial applications where speed and accuracy must coexist. SSD, despite being faster than Faster R-CNN, falls short in precision and recall, indicating limited utility in accuracy-critical settings. The analysis also underscores the importance of lighting conditions in crack detection, as all models show varying degrees of sensitivity to changes in illumination. Future research should focus on improving the robustness of these algorithms to diverse operational conditions and exploring hybrid approaches that combine the strengths of multiple models. These advancements will pave the way for more reliable and efficient crack detection systems in mechanical engineering and beyond.

7. References

Yuan, Y., Ma, S., Wu, J., Jia, B., Li, W., & Luo, X. (2019). Frequency Feature Learning from Vibration Information of GIS for Mechanical Fault Detection. Sensors, 19(8), 1949. https://doi.org/10.3390/s19081949

Wang, T., Tan, B., Lu, M., Zhang, Z., & Lu, G. (2020). Piezoelectric Electro-Mechanical Impedance (EMI) Based Structural Crack Monitoring. Applied Sciences, 10(13), 4648. https://doi.org/10.3390/app10134648

Lou, C., Tinsley, L., Duarte Martinez, F., Gray, S., & Honarvar Shakibaei Asli, B. (2024). Optimized AI Methods for Rapid Crack Detection in Microscopy Images. Electronics, 13(23), 4824. https://doi.org/10.3390/electronics13234824

Rajesh, S., Jinesh Babu, K. S., Chengathir Selvi, M., & Chellapandian, M. (2024). Automated Surface Crack Identification of Reinforced Concrete Members Using an Improved YOLOv4-Tiny-Based Crack Detection Model. Buildings, 14(11), 3402. https://doi.org/10.3390/buildings14113402

Bakirci, M., Dmytrovych, P., Bayraktar, I., Anatoliyovych, O. (2024). Multi-class vehicle detection and classification with YOLO11 on UAV-captured aerial imagery. 2024 IEEE 7th International Conference on Actual Problems of Unmanned Aerial Vehicles Development (APUAVD), pp. 191-196, Kyiv, Ukraine. https://doi.org/10.1109/APUAVD64488.2024.10765862

Ma, L., Li, Y., Li, J., Wang, C., Wang, R., & Chapman, M. A. (2018). Mobile Laser Scanned Point-Clouds for Road Object

Detection and Extraction: A Review. Remote Sensing, 10(10), 1531. https://doi.org/10.3390/rs10101531

Bakirci, M., Bayraktar, I. (2024). Assessment of YOLO11 for ship detection in SAR imagery under open ocean and coastal challenges. 2024 21st International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE), pp. 1-6, Mexico City, Mexico. https://doi.org/10.1109/CCE62852.2024.10770926

Fan, Z., Li, C., Chen, Y., Wei, J., Loprencipe, G., Chen, X., & Di Mascio, P. (2020). Automatic Crack Detection on Road Pavements Using Encoder-Decoder Architecture. Materials, 13(13), 2960. https://doi.org/10.3390/ma13132960

Munawar, H. S., Hammad, A. W. A., Haddad, A., Soares, C. A. P., & Waller, S. T. (2021). Image-Based Crack Detection Methods: A Review. Infrastructures, 6(8), 115. https://doi.org/10.3390/infrastructures6080115

Maslan, J., & Cicmanec, L. (2023). A System for the Automatic Detection and Evaluation of the Runway Surface Cracks Obtained by Unmanned Aerial Vehicle Imagery Using Deep Convolutional Neural Networks. Applied Sciences, 13(10), 6000. https://doi.org/10.3390/app13106000

Bakirci, M., Bayraktar, I. (2024). Improving coastal and port management in smart cities with UAVs and deep learning. 2024 Mediterranean Smart Cities Conference (MSCC), pp. 1-6, Martil - Tetuan, Morocco. https://doi.org/10.1109/MSCC62288.2024.10697069

Lee, Y. -S., & Park, W. -H. (2022). Diagnosis of Depressive Disorder Model on Facial Expression Based on Fast R-CNN. Diagnostics, 12(2), 317. https://doi.org/10.3390/diagnostics12020317

Afzaal, U., Bhattarai, B., Pandeya, Y. R., & Lee, J. (2021). An Instance Segmentation Model for Strawberry Diseases Based on Mask R-CNN. Sensors, 21(19), 6565. https://doi.org/10.3390/s21196565

Ren, Y., Zhu, C., & Xiao, S. (2018). Small Object Detection in Optical Remote Sensing Images via Modified Faster R-CNN. Applied Sciences, 8(5), 813. https://doi.org/10.3390/app8050813

Bakirci, M., Bayraktar, I. (2024). Integrating UAV-based aerial monitoring and SSD for enhanced traffic management in smart cities. 2024 Mediterranean Smart Cities Conference (MSCC), pp. 1-6, Martil - Tetuan, Morocco. https://doi.org/10.1109/MSCC62288.2024.10696996

Avola, D., Cinque, L., Diko, A., Fagioli, A., Foresti, G. L., Mecca, A., Pannone, D., & Piciarelli, C. (2021). MS-Faster R-CNN: Multi-Stream Backbone for Improved Faster R-CNN Object Detection and Aerial Tracking from UAV Images. Remote Sensing, 13(9), 1670. https://doi.org/10.3390/rs13091670

Zhang, L., Zhang, Y., Zhang, Z., Shen, J., & Wang, H. (2019). Real-Time Water Surface Object Detection Based on Improved Faster R-CNN. Sensors, 19(16), 3523. https://doi.org/10.3390/s19163523

Bakirci, M., Bayraktar, I. (2024). Comparative performance of YOLOv9 and YOLOv10 for vehicle detection towards real-time traffic surveillance with UAVs. 2024 21st International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE), pp. 1-6, Mexico City, Mexico. https://doi.org/10.1109/CCE62852.2024.10771048

Bakirci, M., Bayraktar, I. (2024). Harnessing UAV technology and YOLOv9 algorithm for real-time forest fire detection. 2024 International Russian Automation Conference (RusAutoCon), pp. 95-100, Sochi, Russian Federation. https://doi.org/10.1109/RusAutoCon61949.2024.10694663

CHAPTER III

Precision and Sustainability in Military Manufacturing through Advanced Object Detection

Bora Burak Azer[1]
Fikrican Ciftci[2]
Furkan Yildirim[3]

1. Introduction

In factories that mass produce in the military field, the detection of small mechanical parts with object detection methods is a very important issue. This importance stems from the need for the production process to be carried out quickly, reliably and without errors. In the production of military equipment, it is critical that each part is produced in accordance with quality standards, the assembly is done correctly and the final product exhibits the expected performance (Wang et al., 2024). Therefore, the detection and control of small mechanical parts is very important in terms of both

[1] Bora Burak Azer, Koluman Automotive Industry Inc., R&D Department, Ankara/Turkiye, Orcid: 0009-0004-3863-1526, burak.azer@koluman.com
[2] Fikrican Ciftci, Koluman Automotive Industry Inc., R&D Department, Ankara/Turkiye, Orcid: 0009-0004-0538-4995, fikrican.ciftci@koluman.com
[3] Furkan Yildirim, Koluman Automotive Industry Inc., R&D Department, Ankara/Turkiye, Orcid: 0009-0009-5478-3008, furkan.yildirim@koluman.com

quality management and efficiency. First of all, the detection of small mechanical parts with object detection methods is important to minimize error rates (Chalklen et al., 2020). Errors that may occur during the production process can increase production costs and disrupt delivery processes. Traditional manual inspection methods are not effective enough in modern military production facilities because they are both open to human error and time-consuming. At this point, artificial intelligence-supported object detection systems can detect each part on the production line in milliseconds and reduce error rates to much lower levels (Bakirci & Bayraktar, 2024). These methods prevent larger problems that may occur later in the process by detecting faulty parts at an early stage. In addition, the detection of small parts supports automation in the production process and increases efficiency. With the spread of Industry 4.0 technologies, robotic systems and sensor-based solutions have come to the fore in military production facilities (Wang et al., 2021). Object detection systems are a critical part of these automation processes. Advanced technology methods such as cameras, laser scanning devices or ultrasonic sensors can check the size, shape, surface quality and assembly suitability of parts (Andronie et al., 2023 : Bakirci, 2023). These detections are integrated with other machines in the production line, allowing the parts to be automatically separated or included in the assembly process. In this way, the need for human intervention is reduced and the production process becomes faster and more efficient (Papadaki et al., 2023).

Moreover, the detection of small mechanical parts is used to strengthen quality control mechanisms. The importance of production quality in military applications is indisputable, because these equipment are used in critical missions and any error can lead to serious losses. Object detection systems evaluate each manufactured part according to the specified standards and immediately separate the non-conforming parts (Bakirci &

Bayraktar, 2024). In this way, only flawless products reach the end user. In addition, these systems allow for multiple data analysis, allowing for the determination of the root causes of quality problems and the possibility of making improvements in future production cycles. In addition, these technologies support environmentally friendly production processes (Khan et al., 2023). Early detection of faulty parts reduces waste from production and ensures more efficient use of resources. Systems working with recyclable materials offer both economic and ecological benefits. This has become a strategic advantage for the military production sector with the increase in environmental awareness. Finally, the application of object detection methods increases data collection and analysis capacity. In modern military production facilities, data obtained from each production stage is used to optimize system performance and for future projections (Hughes-Riley & Dias, 2018). Object detection systems create a very valuable data pool by recording each step of the production process. This data pool can be analyzed with artificial intelligence and machine learning algorithms to develop more rational solutions. This enables military production processes to become more innovative and competitive. As a result, the detection of small mechanical parts with object detection methods in factories that perform mass production in the military field is of great importance both technically and strategically (Nasim et al., 2024). These methods increase quality by reducing error rates, support automation in the production line, reduce costs and offer an environmentally friendly approach. In addition, they contribute to the more effective future production cycles thanks to data collection and analysis opportunities. All these benefits make object detection technologies an indispensable component in modern military production facilities (Bakirci & Cetin, 2023).

Object detection methods play a critical role in various industrial and academic fields in today's technological ecosystem.

These methods are usually developed as a combination of image processing, artificial intelligence, and sensor-based technologies and aim to quickly and accurately detect features such as size, shape, color, and texture of objects (Bistron & Piotrowski, 2021). Object detection has important areas of use not only in the manufacturing sector, but also in many different application areas such as security, health, transportation, and agriculture. One of the main advantages of these technologies is that they increase the accuracy rate by reducing human-induced errors (Meng et al., 2020 : Liu et al., 2024). Especially in production processes that require high precision, object detection systems enable the early detection of faulty products, thus saving time and cost. For example, a robotic system on the production line automatically sorting a faulty component both increases labor efficiency and ensures the quality of the final product (Bakirci & Toptas, 2022). Another important aspect of object detection methods is that they support automation processes. These systems optimize communication between robots and machines in accordance with Industry 4.0 principles (Ku et al., 2022: Elsisi et al., 2021). Equipped with tools such as cameras, laser scanners and ultrasonic sensors, these systems have real-time data processing capacity (Han et al., 2017). This allows faster and more efficient operations in production lines. In addition, object detection technologies work integrated with big data analytics to monitor and improve operational processes. The collected data can be used in strategic decision-making mechanisms for future processes. This ensures that industrial systems remain continuously innovative and competitive (Bakirci, 2023). As a result, object detection methods stand out as a technological solution that responds to today's complex industrial needs. The importance of these systems in achieving goals such as accuracy, efficiency and sustainability is great.

YOLOv7 (Zheng et al., 2024) is a deep learning-based network architecture that plays an important role in the field of object detection. This model is part of the "You Only Look Once" (YOLO) family and aims to achieve faster and more accurate results than its predecessors. The design of YOLOv7 is optimized to provide particularly fast processing times and high accuracy rates. In this paper, the network architecture and basic features of YOLOv7 will be explained in an academic language.

2. Methodology

Object detection is a fundamental task for computer vision and artificial intelligence applications. YOLO is one of the most effective and popular algorithms in this field, and stands out with its ability to detect objects in images in a single pass. Various improvements have been made since the first version of the YOLO algorithm, and each new version has increased the accuracy, speed, and efficiency of the model (Bakirci et al., 2024). YOLOv7 (Zhang et al., 2023) is the final link in this process and offers more improvements and optimizations compared to previous versions. YOLOv7 comes with many optimizations. These include super-resolution, improvements in optimization algorithms, balancing the model in terms of speed and accuracy, and more efficient training techniques. The balance between speed and accuracy of the model is an important factor, especially in real-time applications. The structural improvements of YOLOv7 allow the model to use less computational resources in the training and testing phases, making it faster and more efficient. YOLOv7 is a model that performs significantly better than its previous versions. It is extremely effective in real-time object detection applications, especially in areas such as video streaming and autonomous vehicles. YOLOv7 is capable of accurately detecting both small objects and large objects, making it suitable for various industrial applications (Zhou et al. 2023).

2.1. Object Detection Based on the YOLOv7 Model

YOLOv7, whose network architecture is given in Figure 1, has an architecture consisting of three main components: backbone, neck, and head. Each of these components helps the model to provide high accuracy and efficiency in object detection.

Backbone is the basic network structure that extracts features from images. The backbone used in YOLOv7 is a structure called CSPDarknet53 (Cross-Stage Partial Darknet). This structure uses a strategy that improves the information flow between layers to increase the performance of deep learning models. CSPDarknet53 provides higher efficiency compared to Darknet structures in previous versions (Zhao & Zhu, 2023).

Neck is a structure that makes the feature maps obtained from the backbone more meaningful. In YOLOv7, a combination of FPN (Feature Pyramid Network) and PANet (Path Aggregation Network) is used. This structure integrates features at different resolution levels and allows the model to detect small and large objects simultaneously. While FPN is used to detect objects at different scales, PANet improves the information flow of the network and provides better results.

The head section is the output section of the network and produces the locations, classes and confidence scores of the detected objects. In YOLOv7, this section is optimized to make more precise predictions. The model estimates the rectangular coordinates (with class labels) for each object, and these predictions are usually made with a "grid"-based approach. The head section of YOLOv7 produces more accurate and faster results than the head structures in previous versions.

YOLOv7 is an efficient and fast deep learning model that has made significant progress in the object detection task. The network architecture and structural improvements used increase the accuracy of the model while optimizing the processing speed. This model, which is especially applicable in the fields of autonomous driving, safety monitoring and robotics, can be an important reference point for future object detection solutions. YOLOv7 stands out as one of the most advanced technologies in the field of deep learning and computer vision.YOLOv8 is fundamentally based on a Convolutional Neural Network (CNN) architecture. In the first stage, the network uses a series of convolutional layers that divide the input image into smaller parts. These layers are designed to extract low-level features (such as edges and corners). This process enables the extraction of higher-level features in deeper layers. One of the key innovations of YOLOv8 lies in the optimizations designed to make this network more efficient. For example, the sparse convolution techniques used in the model reduce computational costs while increasing accuracy.

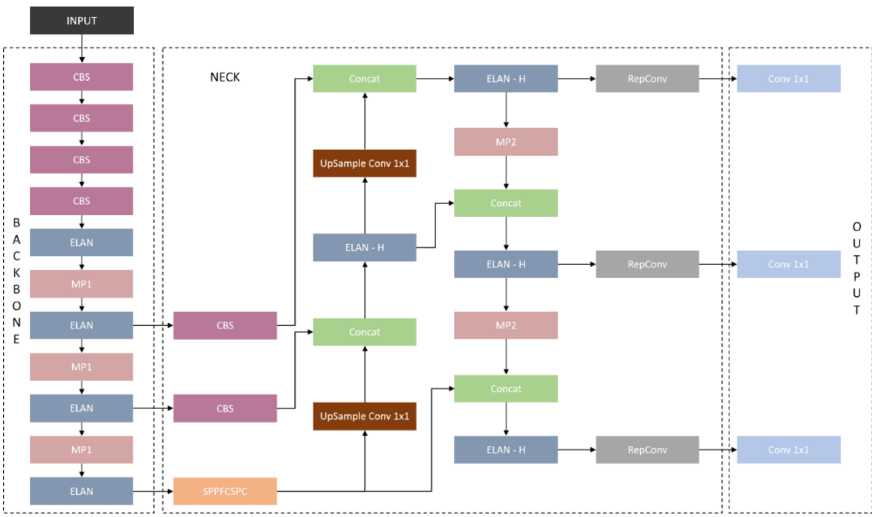


Figure 1: YOLOv7 Network Architecture

2.2. Loss Function

YOLOv7 uses various loss functions to optimize the object detection task. During the training process of the model, three main loss components are calculated for accurate object detection: coordinate loss, class loss, and object existence loss.

Each of these components is optimized from different perspectives to improve the accuracy and efficiency of the model. Coordinate loss is used to accurately estimate the locations of detected objects. YOLOv7 estimates four coordinates (x, y, width, height) for each object. This loss is usually calculated using ciou (Complete Intersection over Union):

$$L_{coord} = 1 - IoU\ (\hat{B}, B)$$

Here, $\hat{B}$ is the predicted box of the model and B is the true box. IoU (Intersection over Union) shown in Figure 2 measures the overlap ratio between the predicted box and the true box. Class loss tries to determine the correct class of the detected objects. In YOLOv7, this loss is calculated using cross-entropy loss:

$$L_{class} = \sum_{i} y_i \log(\widehat{y_i})$$

Here, y_i respresents the true class label and $\widehat{y_i}$ represents the class probabilities predicted by the model.

Object existence loss measures the error of the model in estimating the confidence score of the object existence. This loss is calculated by binary cross-entropy loss:

$$L_{object} = \sum_{i} y_i \log(\widehat{y_i}) + (1 - y_i) \log(1 - \widehat{y_i})$$

Here, y_i is the actual label (1 veya 0) indicating the existence of the object, and $\hat{y}_i$ is the confidence score estimated by the model.

YOLOv7 achieves high accuracy in object detection by minimizing the sum of these losses. The combination of these losses optimizes each component of the model and provides more accurate and efficient results in object detection.

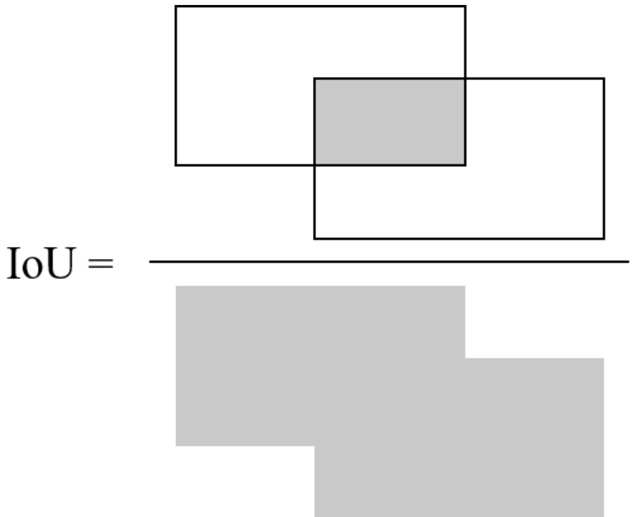


Figure 2: Illustration of the concept of IoU.

3. Dataset

We created a dataset consisting of 1046 bolt images as the basis for our proposed algorithm. To increase the generalization ability of the algorithm, data augmentation techniques were applied to expand the dataset to a total of 2800 images. These augmentation methods included random rotations, cropping, flipping, and changes in color channels. Each augmented image was annotated according to the YOLO dataset format to facilitate object detection tasks.

The dataset was divided into three subsets, namely training, validation, and test sets, with a distribution ratio of 6:2:2. This division ensures that the training set has sufficient data for model learning, while the validation and test sets provide reliable metrics to evaluate performance and prevent overfitting (Bakirci & Bayraktar, 2024). To maintain experimental consistency and fairness, the same initial training parameters were used in all experiments.

Before training, the resolution of all input images was standardized to 640x640 pixels. Each model was trained for 250 epochs to provide sufficient learning opportunities. Training hyperparameters were carefully selected to optimize performance. The initial learning rate was set to 0.001 to allow for efficient gradient descent updates, while a momentum coefficient of 0.935 was used to stabilize training and speed up convergence. Additionally, a weight decay coefficient of 0.0004 was applied to regularize the model and prevent overfitting by penalizing large weights.

4. Hardware and Software Configurations

The experimental environment, including hardware and software configurations, is detailed in Table 1 to ensure reproducibility. The detection of mechanical components, specifically bolts in this case, was carried out on a system with a high-performance hardware and software configuration, enabling efficient and accurate model training and inference. The hardware utilized includes an NVIDIA RTX 3060 GPU, a graphics processing unit equipped with 12GB of GDDR6 memory, which is particularly well-suited for deep learning tasks, especially real-time object detection models like YOLOv7. The CUDA cores and Tensor cores present in the RTX 3060 ensure accelerated computations for parallel processing of matrix operations, which are critical for neural network training and inference.

The CPU used in the system is the Intel Xeon E5, which is widely recognized for its high computational power and multi-threading capabilities. Xeon processors are typically utilized in workstations and server-grade systems for intensive computational workloads, making them suitable for pre-processing data, running auxiliary scripts, and managing the overall pipeline during training. Additionally, the system is equipped with 128GB DDR5 RAM, which provides ample memory for handling large datasets, caching operations, and ensuring smooth operation during model training and testing phases. This significant memory capacity is particularly advantageous when dealing with high-resolution images or large-scale object detection tasks that require substantial memory for batch processing.

The operating system running on the hardware is Mac OS, which offers a stable and secure environment for deep learning development. While Mac OS is not as common as Linux-based systems for deep learning tasks, it remains a suitable platform for running frameworks such as PyTorch, given proper GPU support and CUDA configuration. The deep learning framework utilized in this study is PyTorch, a widely adopted library for developing machine learning and deep learning models. PyTorch is preferred for its dynamic computational graph, ease of use, and extensive community support. It is highly optimized for GPU-based operations and integrates seamlessly with CUDA, enabling faster matrix computations for YOLOv7 inference.

The software environment was further supported by Python version 3.10.8, which serves as the programming language for implementing the YOLOv7 model. Python's versatility, wide range of libraries, and compatibility with PyTorch make it an ideal choice for deep learning research. In particular, Python allows for efficient pre-processing of images, annotation management, and post-processing of model outputs.

In summary, the combination of an NVIDIA RTX 3060 GPU, Intel Xeon E5 CPU, 128GB DDR5 RAM, and PyTorch on Python 3.10.8 running on Mac OS provided a robust and efficient environment for the YOLOv7-based object detection task. The GPU's computational power accelerated model inference, while the CPU and RAM ensured smooth handling of large datasets and image processing tasks. This configuration highlights the importance of high-performance hardware and optimized software environments in achieving accurate and efficient results in real-world object detection applications. Future studies with even more advanced GPUs or distributed systems could further enhance the performance of similar tasks, especially in industrial and military production environments requiring real-time detection capabilities.

Parameter	Configuration
GPU	NVIDIA RTX 3060
CPU	Intel Xeon E5
RAM	128GB DDR5
Operating System	Mac OS
DL Framework	PyTorch
Python Version	3.10.8

Table 1: Hardware and Software Configurations

5. Test Results

This study presents an evaluation of the detection of mechanical components used in the production of military vehicles with the YOLOv7 model. In the provided image which shows in Fig. 2, bolts of various sizes and positions were detected by the YOLOv7 model and are shown with blue rectangles. When examining the overall performance of the algorithm, it can be observed that the model successfully identifies and classifies many bolts. However, in cases where some bolts are partially obscured by other objects, the model fails to detect these bolts.

The YOLOv7 (You Only Look Once) model is a deep learning-based algorithm developed for real-time object detection. One of the key advantages of this model is its ability to make fast and accurate predictions for multi-object detection. However, there are certain limitations to this success. In particular, the visibility problem (occlusion), where objects overlap and obscure one another, can reduce detection performance. In this image, the high density of bolts and the partial occlusion caused by neighboring mechanical components have significantly affected the YOLOv7 model's detection performance.

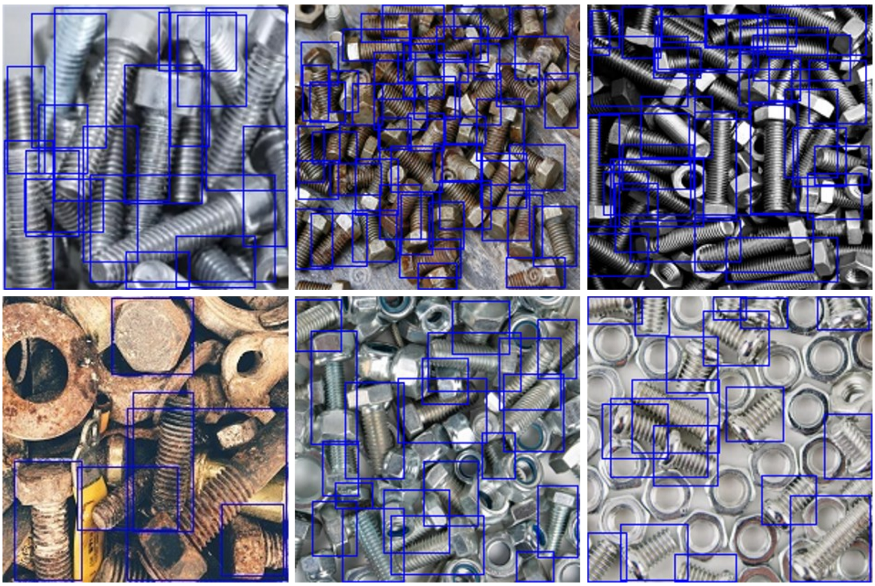


Figure 3: Detection examples with YOLOv7 model.

The impact of the occlusion problem on detection accuracy is a common challenge in computer vision studies. Single-stage object detection algorithms like YOLOv7 are highly effective at detecting large objects and those with distinct edges. However, they are prone to errors in conditions involving occlusion, small objects, or complex backgrounds. This phenomenon is also observed in the presented image. Bolts positioned close to each other may cause the model to perceive multiple objects as a single bolt or fail to detect some of them.

Furthermore, the presence of varying color tones and material surfaces in the image is another factor affecting the overall performance of the model. The different reflections of corroded or rusted bolts have made it difficult for the algorithm to distinguish these objects. On the other hand, shiny metallic surfaces have also led to misleading results in the shape recognition process.

In conclusion, although the YOLOv7 model successfully detects most of the bolts in the image, factors such as occlusion, complex arrangements, and surface reflections have caused some bolts to remain undetected. These results emphasize the need for continued research to improve object detection algorithms for industrial applications. In future studies, enriching datasets, using multi-angle perspectives to address the occlusion problem, and supporting models like YOLOv7 with multi-layer learning strategies will be beneficial.

5.1. Test Results in Numerical

This study presents an evaluation of the detection of mechanical components used in the production of military vehicles with the YOLOv7 model. In the provided image, bolts of various sizes and positions were detected by the YOLOv7 model and are shown with blue rectangles. When examining the overall performance of the algorithm, it can be observed that the model successfully identifies and classifies many bolts. However, in cases where some bolts are partially obscured by other objects, the model fails to detect these bolts. The quantitative performance evaluation for this test yields the following results: Precision is measured as 0.899, Recall as 0.844, the mean Average Precision (mAP) as 0.863, and the F1-Score as 0.871, with a mean inference speed of 28 FPS (frames per second). Precision, in this context, refers to the ratio of correctly predicted bolts to the total predicted bolts, indicating the model's ability to avoid false positives.

A Precision value of 0.899 suggests that 89.9% of the predicted bolts are accurate, which is a strong result for the YOLOv7 model. Recall, on the other hand, measures the ratio of correctly detected bolts to the total number of ground-truth bolts present in the image. With a Recall value of 0.844, the model successfully identifies 84.4% of all bolts, although some bolts remain undetected due to occlusion or complex spatial arrangements. The mAP metric, a widely used measure in object detection tasks, represents the mean of Average Precision values across all object classes and Intersection over Union (IoU) thresholds. An mAP value of 0.863 indicates that the model achieves a high level of accuracy when both Precision and Recall are considered together, further demonstrating its robustness. The F1-Score, which is the harmonic mean of Precision and Recall, combines these two metrics into a single value to balance the trade-off between false positives and false negatives. The F1-Score of 0.871 shows that the YOLOv7 model achieves a well-balanced performance in detecting bolts under the current conditions. Furthermore, the mean inference speed, reported as 28 FPS, highlights the model's capability to process frames in near-real time, making it suitable for industrial applications where high-speed object detection is required.

Despite these strong performance metrics, the challenges posed by occlusion, overlapping bolts, and varying material surfaces, such as corrosion and reflections, limit the model's ability to achieve perfect detection results. Specifically, bolts partially hidden behind other mechanical parts or in regions with high visual complexity reduce the Recall metric, as they remain undetected. Additionally, reflective surfaces and inconsistent lighting conditions may cause minor inaccuracies in localization, affecting the Precision score. Overall, the YOLOv7 model demonstrates a high level of accuracy and speed in detecting bolts, as evidenced by the reported metrics. However, addressing the issues of occlusion and improving the model's robustness against visual complexities will be crucial for further enhancing its performance in real-world industrial environments. Future work can focus on augmenting the training dataset with occlusion-heavy images, utilizing multi-view detection techniques, or integrating post-processing algorithms to mitigate false negatives and improve overall detection efficiency.

4. Conclusion

In conclusion, the integration of advanced object detection methods, particularly the YOLOv7 algorithm, plays a transformative role in enhancing the efficiency, accuracy, and reliability of military production processes. The study demonstrates that the ability of YOLOv7 to detect small mechanical components, such as bolts, not only minimizes error rates but also strengthens quality control mechanisms and promotes automation within Industry 4.0 environments. This advancement has been validated through rigorous experimentation, where the model achieved high precision (0.899), recall (0.844), and mean average precision (0.863) metrics, showcasing its robust performance in detecting objects even under challenging conditions like occlusion and surface variations. However, limitations such as reduced detection performance in occluded environments and the influence of reflective or corroded surfaces underscore the need for further enhancements in algorithmic design and dataset enrichment.

Moreover, the deployment of YOLOv7 contributes significantly to sustainable and resource-efficient production processes. By reducing waste through early fault detection and optimizing resource utilization, these technologies align with contemporary environmental objectives while maintaining the stringent quality standards required in military applications. The real-time capabilities of YOLOv7, as evidenced by its mean inference speed of 28 FPS, further position it as an indispensable tool for industrial settings demanding high-speed object detection.

The study underscores the critical role of robust hardware and optimized software environments in achieving these outcomes. Utilizing high-performance configurations such as the NVIDIA RTX 3060 GPU and Intel Xeon E5 CPU, coupled with PyTorch's dynamic deep learning capabilities, provided a stable foundation for implementing YOLOv7 effectively. This emphasizes the importance of a synergistic relationship between technological infrastructure and algorithmic advancements.

Looking ahead, addressing the current model's limitations through strategies such as multi-angle perspectives, improved handling of occlusion, and integration of multi-layer learning frameworks holds promise for further enhancing its detection capabilities. Additionally, augmenting the training datasets with diverse scenarios and leveraging big data analytics for continuous optimization can significantly elevate the model's efficacy. These advancements will not only bolster its applicability in military production but also extend its relevance across diverse industrial and academic domains.

Ultimately, the findings of this study affirm that the adoption of cutting-edge object detection technologies like YOLOv7 is pivotal in advancing the precision, efficiency, and sustainability of modern industrial systems. As industries continue to embrace automation and artificial intelligence, the role of such technologies will only grow, paving the way for more innovative, competitive, and environmentally conscious production methodologies.

5. References

Wang, W., Chen, J., Han, G., Shi, X., & Qian, G. (2024). Application of Object Detection Algorithms in Non-Destructive Testing of Pressure Equipment: A Review. Sensors, 24(18), 5944. https://doi.org/10.3390/s24185944

Chalklen, T., Jing, Q., & Kar-Narayan, S. (2020). Biosensors Based on Mechanical and Electrical Detection Techniques. Sensors, 20(19), 5605. https://doi.org/10.3390/s20195605

Bakirci, M., Bayraktar, I. (2024). Refining transportation automation with convolutional neural network-based vehicle detection via UAVs. 2024 International Russian Automation Conference (RusAutoCon), pp. 150-155, Sochi, Russian Federation. https://doi.org/10.1109/RusAutoCon61949.2024.10694108

Wang, D., Wang, J. -G., & Xu, K. (2021). Deep Learning for Object Detection, Classification and Tracking in Industry Applications. Sensors, 21(21), 7349. https://doi.org/10.3390/s21217349

Andronie, M., Lăzăroiu, G., Iatagan, M., Hurloiu, I., Ștefănescu, R., Dijmărescu, A., & Dijmărescu, I. (2023). Big Data Management Algorithms, Deep Learning-Based Object Detection Technologies, and Geospatial Simulation and Sensor Fusion Tools in the Internet of Robotic Things. ISPRS International Journal of Geo-Information, 12(2), 35. https://doi.org/10.3390/ijgi12020035

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

Papadaki, A., & Pateraki, M. (2023). 6D Object Localization in Car-Assembly Industrial Environment. Journal of Imaging, 9(3), 72. https://doi.org/10.3390/jimaging9030072

Bakirci, M., Bayraktar, I. (2024). Comparative performance of YOLOv9 and YOLOv10 for vehicle detection towards real-time traffic surveillance with UAVs. 2024 21st International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE), pp. 1-6, Mexico City, Mexico. https://doi.org/10.1109/CCE62852.2024.10771048

Khan, S. A., Lee, H. J., & Lim, H. (2023). Enhancing Object Detection in Self-Driving Cars Using a Hybrid Approach. Electronics, 12(13), 2768. https://doi.org/10.3390/electronics12132768

Hughes-Riley, T., & Dias, T. (2018). Developing an Acoustic Sensing Yarn for Health Surveillance in a Military Setting. Sensors, 18(5), 1590. https://doi.org/10.3390/s18051590

Nasim, M., Mumtaz, R., Ahmad, M., & Ali, A. (2024). Fabric Defect Detection in Real World Manufacturing Using Deep Learning. Information, 15(8), 476. https://doi.org/10.3390/info15080476

Bakirci, M., Cetin, M. (2023). Improving position-time trajectory accuracy in vehicle stop-and-go scenarios by using a mobile robot as a testbed. Journal of Control Engineering and Applied Informatics, 25(3), 35-44. https://doi.org/10.61416/ceai.v25i3.8365

Bistron, M., & Piotrowski, Z. (2021). Artificial Intelligence Applications in Military Systems and Their Influence on Sense of Security of Citizens. Electronics, 10(7), 871. https://doi.org/10.3390/electronics10070871

Meng, Z., Zhang, M., & Wang, H. (2020). CNN with Pose Segmentation for Suspicious Object Detection in MMW Security Images. Sensors, 20(17), 4974. https://doi.org/10.3390/s20174974

Liu, Z., Chen, C., Huang, Z., Chang, Y. C., Liu, L., & Pei, Q. (2024). A Low-Cost and Lightweight Real-Time Object-Detection Method Based on UAV Remote Sensing in Transportation Systems. Remote Sensing, 16(19), 3712. https://doi.org/10.3390/rs16193712

Bakirci, M., Toptas, B. (2022). Kinematics and autoregressive model analysis of a differential drive mobile robot. (IEEE) 4th International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA). pp. 1-6. June 9-11, Ankara, Turkey. https://doi.org/10.1109/HORA55278.2022.9800071

Ku, B., Kim, K., & Jeong, J. (2022). Real-Time ISR-YOLOv4 Based Small Object Detection for Safe Shop Floor in Smart Factories. *Electronics*, *11*(15), 2348. https://doi.org/10.3390/electronics11152348

Elsisi, M., Tran, M. -Q., Mahmoud, K., Lehtonen, M., & Darwish, M. M. F. (2021). Deep Learning-Based Industry 4.0 and Internet of Things towards Effective Energy Management for Smart Buildings. *Sensors*, *21*(4), 1038. https://doi.org/10.3390/s21041038

Han, X., Zhong, Y., & Zhang, L. (2017). An Efficient and Robust Integrated Geospatial Object Detection Framework for High Spatial Resolution Remote Sensing Imagery. *Remote Sensing*, *9*(7), 666. https://doi.org/10.3390/rs9070666

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

Zheng, K., Liang, H., Zhao, H., Chen, Z., Xie, G., Li, L., Lu, J., & Long, Z. (2024). Application and Analysis of the MFF-YOLOv7 Model in Underwater Sonar Image Target Detection. *Journal of Marine Science and Engineering*, *12*(12), 2326. https://doi.org/10.3390/jmse12122326

Zhang, Y., Sun, Y., Wang, Z., & Jiang, Y. (2023). YOLOv7-RAR for Urban Vehicle Detection. *Sensors*, *23*(4), 1801. https://doi.org/10.3390/s23041801

Zhou, S., Cai, K., Feng, Y., Tang, X., Pang, H., He, J., & Shi, X. (2023). An Accurate Detection Model of *Takifugu rubripes* Using an Improved YOLO-V7 Network. *Journal of Marine Science and Engineering*, *11*(5), 1051. https://doi.org/10.3390/jmse11051051

Bakirci, M., Dmytrovych, P., Bayraktar, I., Anatoliyovych, O. (2024). Multi-class vehicle detection and classification with YOLO11 on UAV-captured aerial imagery. 2024 IEEE 7th International Conference on Actual Problems of Unmanned Aerial Vehicles Development (APUAVD), pp. 191-196, Kyiv, Ukraine. https://doi.org/10.1109/APUAVD64488.2024.10765862

Zhao, L., & Zhu, M. (2023). MS-YOLOv7:YOLOv7 Based on Multi-Scale for Object Detection on UAV Aerial Photography. *Drones*, *7*(3), 188. https://doi.org/10.3390/drones7030188

Bakirci, M., Bayraktar, I. (2024). Assessment of YOLO11 for ship detection in SAR imagery under open ocean and coastal challenges. 2024 21st International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE), pp. 1-6, Mexico City, Mexico. https://doi.org/10.1109/CCE62852.2024.10770926


MATRIX