Comparison of CNN-Based Architectures for Detection of Different Object Classe

Nataliya Bilous *, Vladyslav Malko, Marcus Frohme *, Alina Nechyporenko

Posted Date: 17 September 2024

doi: 10.20944/preprints202409.1310.v1

Keywords: deep learning; object detection; neural network; YOLO; SSD; EfficientDet

Article

Comparison of CNN-Based Architectures for Detection of Different Object Classes

Nataliya Bilous [1,2,*], Vladyslav Malko [1], Marcus Frohme [2,*] and Alina Nechyporenko [1,2]

[1] Kharkiv National University of Radio Electronics, Kharkiv, Ukraine
[2] Division Molecular Biotechnology and Functional Genomics, Technical University of Applied Sciences Wildau, 15745 Wildau, Germany
* Correspondence: nataliya.bilous@nure.ua; mfrohme@th-wildau.de; Tel.: +380679190676

Abstract: Detecting people and technical objects in various situations, such as natural disasters and warfare, is critical to search and rescue operations and the safety of the civilian. A fast and accurate detection for people and equipment can significantly increase the effectiveness of search and rescue missions and provide timely assistance to people. Computer vision and deep learning technologies play a key role in detecting the required objects due to their ability to analyse big volumes of visual data in real time. The performance of the neural networks such as YOLOv4-v8, Faster R-CNN, SSD, and EfficientDet has been analysed using COCO2017, SARD, SeaDronesSee, and VisDrone2019 datasets. Main criteria for comparison were mAP, Precision, Recall, F1-Score, and the ability of the neural network to work in real time. The most important metrics for evaluating the efficiency and performance of models for a given task are accuracy (mAP), F1-Score, and processing speed (FPS). These metrics allow us to evaluate both the accuracy of object recognition and the ability to use the models in real-world environments where high processing speed is important. Although different neural networks perform better on certain types of metrics, YOLO outperforms them on all metrics, showing the best results of mAP-0.88, F1 - 0.88, and FPS - 48, so the focus was on these models.

Keywords: deep learning; object detection; neural network; YOLO; SSD; EfficientDet

1. Introduction

The detection of people and technical objects in various critical situations, such as natural disasters and military operations, plays a key role in search and rescue operations and ensures the safety of civilians. Fast and accurate search for people and equipment significantly increases the efficiency of rescue missions and enables timely assistance to victims. Modern computer vision and deep learning technologies, with their ability to analyze large amounts of visual data in real time, are becoming indispensable tools in these processes. In recent years, there has been significant progress in the development of neural networks, which has opened new possibilities for image analysis and processing. This study aims to comparatively analyze several state-of-the-art neural networks to identify the most productive models for object recognition tasks in complex environments. The COCO2017, SARD, SeaDronesSee and VisDrone2019 datasets were used as data sources, which provide diverse and realistic scenarios for testing the models. The neural networks considered for the study were YOLOv4-v8, Faster R-CNN, SSD and EfficientDet. The main criteria for comparison were the average detection accuracy (mAP), precision (Precision), completeness (Recall), F1-Score and real-time ability of the neural network. These metrics were chosen for a reason: they allow for a comprehensive evaluation of both object recognition accuracy and model performance in real-world environments where high processing speed is crucial. The results of the comparative analysis showed that different neural networks exhibit different levels of performance depending on the type of metric. However, the YOLO family of models (v4-v8) showed superior performance in all key metrics. YOLO is characterized by high accuracy, fast processing speed and robustness to noise and interference, making these models most suitable for use in real-world scenarios requiring high performance. This research highlights the importance of applying neural networks to object recognition tasks in complex and critical environments. State-of-the-art models such as YOLO can

 doi:10.20944/preprints202409.1310.v

greatly improve the efficiency of search and rescue operations and provide higher civilian safety. Further development and deployment of these technologies opens new opportunities for automation and efficiency improvements in various fields such as agriculture, autonomous transport and environmental monitoring. The aim of this research is to compare the performance of different neural network models (YOLOv4-v8, Faster R-CNN, SSD and EfficientDet) in human and technical object detection tasks in different environments such as water, road and complex environments to determine the most efficient model. The subject of the study is deep learning algorithms and models used for detecting and classifying objects in images, including people and technical objects, in a variety of environmental conditions. The object of the study is various neural network models such as YOLOv4-v8, Faster R-CNN, SSD and EfficientDet and their performance in solving object detection problems in complex environments. The relevance of the study stems from the growing need for efficient detection and monitoring systems that can operate in real-time in various environments including water and road scenes. Such systems are important for security, rescue operations, autonomous vehicles and other critical applications.

2. Review of the Literature

Object detection in images using neural networks has become one of the key challenges in computer vision. In recent years, there have been many studies aimed at developing and improving deep learning methods for recognizing different classes of objects. Let us review the main advances and approaches that formed the basis of our research. The YOLO model [1], proposed by Joseph Redmon and colleagues in 2016, was a breakthrough in object detection. YOLO is characterized by high-speed image processing and real-time capability, which makes it particularly useful for tasks requiring fast response. Subsequent versions (YOLOv2, YOLOv3, YOLOv4 and onward to YOLOv8) have made significant improvements in accuracy and robustness to various lighting conditions and noise. Studies have shown that YOLO works effectively in a variety of applications including traffic monitoring, security and autonomous systems. Faster R-CNN [2], proposed by Shao Jin and colleagues in 2015, is an improved version of R-CNN and Fast R-CNN. The main difference of Faster R-CNN is the utilization of Region Proposal Network (RPN), which greatly speeds up the object detection process. Despite its high accuracy, this model has a relatively low processing speed compared to YOLO, which limits its application in real time. The SSD model [3] proposed by Wei Liu and colleagues in 2016 also focuses on high speed image processing. SSD utilizes multi-resolution feature maps to detect objects at different scales, which achieves a high level of accuracy with low latency. SSD is widely used in tasks related to autonomous driving and robotics. EfficientDet [4], proposed in 2020, is a family of models based on EfficientNet. The main focus of EfficientDet is on computational efficiency and resource efficiency, making it ideal for use in computationally constrained environments. EfficientDet shows high accuracy and performance, especially in the context of mobile and embedded systems.

Saieshan Reddy [5] in his research has conducted a comprehensive analysis of three convolutional neural network (CNN) based object detection algorithms for vehicle detection task. The algorithms considered include Faster R-CNN, YOLO v3 and SSD. Reddys results show that SSD outperforms other algorithms in terms of average accuracy (mAP 0.92) and detection time (0.5 seconds per image), despite higher values of average loss. YOLO v3 also performs well, striking a balance between accuracy and speed (mAP 0.81 and 1.16 seconds per image). Faster R-CNN, although it has high accuracy (mAP 0.76), is inferior in terms of processing speed (5.1 seconds per image), which limits its real-time applications. Chaoyue Sun et al.[6] presented the MRD-YOLO model for object detection in challenging road conditions using multispectral images. Current visible light-based algorithms often face problems of misses and false positives in poor visibility. To address these problems, multispectral images combining RGB, and infrared channel data have been used. MRD-YOLO includes a BIC-Fusion module for efficient information fusion, a SAConv module for improved detection of objects of different scales, and an AIFI framework for better utilization of semantic information. Experiments on FLIR_Aligned and M3FD datasets have shown that MRD-YOLO outperforms existing algorithms in terms of detection accuracy in challenging road conditions,

 doi:10.20944/preprints202409.1310.v

avoiding misses and false positives. Thus, MRD-YOLO significantly improves object detection in autonomous vehicle systems by providing reliable detection in a variety of challenging environments.

Yanyan Dai et al.[7] proposed a novel approach to object detection and tracking for autonomous vehicles by combining YOLOv8 and LiDAR. Traditional methods based on computer vision often encounter difficulties under varying illumination and complex environments. To improve results, the authors combined YOLOv8, which provides high accuracy object detection in RGB images, with LiDAR, which provides 3D distance information. The paper presents a solution for calibrating data between sensors, filtering ground points from LiDAR clouds, and managing computational complexity. A filtering algorithm trained on YOLOv8, a calibration method for converting 3D coordinates to pixel coordinates, and object clustering based on the merged data are developed. Experiments on an Agilex Scout Mini robot with Velodyne LiDAR and Intel D435 camera confirmed the effectiveness of the approach, improving the performance and reliability of object detection and tracking systems. Nataliya Bilous et al.[8] developed a method to identify and compare human postures and exercises based on 3D joint coordinates. This method is intended for analyzing human movements, especially in medicine and sports, where it is important to evaluate the correctness of exercise performance. The method uses pose descriptions in the form of logical statements and is robust to data errors, independent of shooting angle and human proportions. Joint coordinates are corrected for the length of the bones connecting them, which improves positional accuracy and eliminates the need for outlier processing. To remove errors, a method of averaging the graph along each axis is used to group consecutive points so that the difference between the maximum and minimum value does not exceed the error. The groups are then filtered, leaving only those where both points are smaller, or both are larger. The proposed method only requires a state-of-the-art smartphone and does not impose restrictions on the way exercise videos are captured. The method was tested on UTKinect-Action3D, SBU-Kinect-Interaction v2.0, Florence3D, JHMDB and NTU RGB+D 120 datasets, as well as on a proprietary dataset collected using ARKit and BlazePose. The results showed high motion tracking accuracy and robustness of the method to data errors, making it suitable for applications in various scenarios including fitness, rehabilitation and health monitoring. Daud Khan et al.[9] proposed the integration of YOLOv3 and MobileNet SSD algorithms to improve real-time object detection. The research aims to overcome the problems of blur, noise and rotational distortion that affect the detection accuracy in real-world environments. YOLOv3 provides high speed and accuracy by detecting multiple objects in a single image, while MobileNet SSD balances speed and accuracy on resource constrained devices. The authors attempt to improve the accuracy of object localization and performance of algorithms in various applications such as augmented reality, robotics, surveillance systems, and autonomous vehicles. The integration of these technologies improves safety, provides immediate response to threats, and allows robots to better interact with their environment. The authors also compared lightweight versions of YOLOv3 and YOLOv4 in terms of daytime and nighttime accuracy. Experiments showed significant improvements in the performance and reliability of object detection systems when these algorithms were integrated. Further optimization including hardware acceleration and object tracking techniques are recommended to improve real-time detection capabilities. In their research, Dinesh Suryavanshi et al.[10] compare the YOLO and SSD algorithms for real-time object detection. The objective is to carry out a comparative analysis of the accuracy, speed, and efficiency of two popular deep learning-based algorithms. The COCO (Common Objects in Context) dataset was employed for the evaluation of performance. The YOLO algorithm is distinguished by its high speed and capacity to detect multiple objects in a single image in a single iteration of the network, rendering it well-suited for real-time, performance-intensive tasks such as traffic monitoring and autonomous vehicles. YOLO is distinguished by its high accuracy and low background error, rendering it an effective choice for a diverse array of applications. In contrast, the SSD (Single Shot Detector) algorithm is capable of detecting multiple objects in a single iteration. However, in contrast to YOLO, it employs a more intricate architectural design comprising multiple convolutional layers, with the objective of enhancing the precision of the detection process. The SSD algorithm is particularly effective for tasks

 doi:10.20944/preprints202409.1310.v

that require more accurate object recognition, such as forensic analysis and landmark detection. The research revealed that YOLO exhibits superior speed and real-time performance compared to SSD; however, SSD displays enhanced accuracy in complex environments. The selection of an appropriate algorithm is dependent upon the specific accuracy and speed requirements of the intended application. Therefore, YOLO is more appropriate for applications that necessitate rapid detection, whereas SSD may be preferable for tasks that demand more precise object recognition. Akshatha K.R. et al.[11] investigated the use of Faster R-CNN and SSD algorithms for detecting people in aerial thermal images. The main objective of the study is to evaluate the performance of these algorithms under conditions where objects (people) are small in size and images have low resolution, which often causes problems for standard object detection algorithms. The study utilized two standard datasets, OSU Thermal Pedestrian and AAU PD T. The algorithms were tuned with different network architectures such as ResNet50, Inception-v2, and MobileNet-v1 to improve detection accuracy and speed. The results showed that the Faster R-CNN model with ResNet50 architecture achieved the highest accuracy, showing an average detection accuracy (mAP) of 100% for the OSU test dataset and 55.7% for the AAU PD T dataset. At the same time, the SSD with MobileNet-v1 architecture delivered the fastest detection rate of 44 frames per second (FPS) on the NVIDIA GeForce GTX 1080 GPU. Algorithm parameters such as anchor scaling and network steps were fine-tuned to improve performance. These tweaks resulted in a 10% increase in mAP accuracy for the Faster R-CNN ResNet50 model and a 3.5% increase for the Inception-v2 SSD. Thus, the study shows that Faster R-CNN is more suitable for tasks requiring high accuracy, while SSD is better suited for applications requiring high real-time detection rates. Heng Zhang et al. [12] presented an improved Faster R-CNN algorithm for real-time vehicle detection based on frame difference and spatiotemporal context. Unlike the original approach that uses anchors, the new method uses the frame difference to highlight regions of interest. The spatial context enhances the expression of target information, while the temporal context improves the detection performance. Experiments showed the high performance of the proposed method with low sensitivity to background changes. Shengxin Gao et al. [13] investigated the performance of a pedestrian detection model based on the Faster R-CNN algorithm. The model was trained on the Caltech Pedestrian dataset and showed an average precision (AP) of 51.9%, indicating high accuracy in pedestrian detection tasks. The processing time per image was only 0.07 seconds, making the model suitable for real-time applications. The model also consumes only 158 MB of memory, making it easy to implement on resource-constrained devices. The study emphasizes that even with relatively low AP, the model demonstrated fast processing speed, making it useful for applications such as surveillance systems and autonomous vehicles. The Faster R-CNN algorithm incorporates a region proposal network (RPN) to generate regions likely to contain pedestrians and uses a convolutional neural network for accurate localization and classification. The study applied data augmentation techniques such as random flipping, scaling, and pruning to improve the generalization ability of the model. The results showed that the model achieved the highest accuracy at the 45th training epoch, demonstrating an AP of 51.9%. Despite this, the model managed to maintain a high inference rate and compact memory size, which emphasizes its suitability for real-world applications. In the future, it is possible to improve the performance of the model by introducing more advanced architectures and carefully tuning the hyperparameters. This study provides valuable insights for further developments in computer vision and emphasizes the potential of using the Faster R-CNN algorithm in real-world pedestrian detection applications. Zbigniew Omiotek et al. [14] in their study investigated the application of the Faster R-CNN algorithm to detect dangerous objects in surveillance camera images. The researchers developed a customized dataset including images of baseball bats, pistols, knives, machetes, and rifles under different lighting and visibility conditions. Using different backbone networks, the best results were achieved with ResNet152, which showed an average detection accuracy (mAP) of 85% and AP of 80% to 91% depending on the object. The average detection rate was 11-13 frames per second, making the model suitable for public safety systems. The study emphasizes that most existing solutions overestimate their effectiveness due to the insufficient quality of the datasets used. The dataset created presents objects in a variety of conditions, including partial overlap, blur, and poor lighting,

 doi:10.20944/preprints202409.1310.v

to better reflect real-world conditions. This improves the robustness of the results and the detection accuracy. The Faster R-CNN algorithm with different backbone networks such as ResNet152 has been tested and shown to perform well. The average accuracy for baseball bat detection was 87.8%, 91.3% for pistol, 80.8% for knife, 79.7% for machete, and 85.3% for rifle. Despite the high accuracy, the model managed to keep the processing speed at 11-13 frames per second. The experiments showed that the Faster R-CNN with ResNet152 backbone network is most effective for public safety monitoring tasks, which allows us to recommend it for use in video surveillance systems. The authors also note that further research may include increasing the number of training examples and using data augmentation techniques to improve the accuracy and reliability of the model. Tong Bai et al. [15] presented a method for recognizing vehicle types in images using an improved Faster R-CNN model. As the number of vehicles increases, traffic jams, accidents and vehicle-related crimes increase, which requires improved traffic management methods. The study improves three aspects of the model: combining features from different layers of the convolutional network, using contextual features, and optimizing bounding boxes to improve recognition accuracy. A dataset with images of cars, SUVs and vans was used in the experiments. The results showed that the improved model effectively identifies vehicle types with high accuracy. The average recognition accuracy (AP) for the three vehicle types was 83.2%, 79.2% and 78.4%, respectively, which is 1.7% higher than the traditional Faster R-CNN model. These improvements make the proposed model suitable for use in intelligent transportation systems, improving the accuracy and reliability of traffic management and safety. Yanfeng Wang et al. [16] presented the TransEffiDet method for aircraft detection and classification in aerial images, combining the EfficientDet algorithm and Transformer module. The main objective of the study was to improve the accuracy and reliability of object detection in low light and high background conditions. The EfficientDet model is used as the core network, providing efficient fusion of feature maps of different scales. Transformer analyzes global features to extract long-term dependency. Experimental results show that TransEffiDet achieves an average detection accuracy (mAP) of 86.6%, which is 5.8% higher than EfficientDet, demonstrating high accuracy and robustness in military applications. The generated MADAI dataset, which includes images of military and civilian aircraft, will be published with the study. Key achievements include integration of multi-scale and multi-dimensional features, reduced computational complexity, and improved object recognition accuracy. In the future, the researchers plan to apply this method to military target detection and explore the use of additional Transformer layers to improve accuracy. Munteanu, D. et al. [17] have carried out a research to develop a real-time automatic sea mine detection system using three deep learning models: YOLOv5, SSD, and EfficientDet. In the current armed conflicts and geopolitical tensions, navigation safety is jeopardized by the large number of sea mines, especially in maritime conflict areas. The study utilized images captured from drone, submarine and ship cameras. Due to the limited number of sea mine images available, the researchers applied data augmentation techniques and synthetic image generation, creating two datasets: for floating mines and underwater mines. The models were trained and tested on these datasets and evaluated for accuracy and processing time. The results showed that the YOLOv5 algorithm provided the best detection accuracy for both floating and underwater mines, achieving high performance with a relatively short training time. The SSD model also performed well, although it required more computational resources. The EfficientDet model, one of the newest neural networks, showed high performance in detection tasks due to innovations introduced in previous models. Tests on handheld devices such as the Raspberry Pi have shown that the system can be used in real-world scenarios, providing mine detection with a latency of about 2 seconds per frame. These results highlight the potential of using deep learning to improve maritime safety in conflict situations. The research presented by O. Hramm et al [18]. describes a customizable solution for cell segmentation using the Hough transform and watershed algorithm. The paper emphasizes the importance of developing flexible algorithms that can adapt to different datasets and imaging conditions to improve recognition accuracy and reliability. The algorithm proposed by the authors has many tunable parameters that can optimize the segmentation process for different types of images. The Hough transform is used to define circles, which helps in clustering overlapping cells, while the watershed algorithm segments cells even if they overlap or are

 doi:10.20944/preprints202409.1310.v

in complex conditions. Major achievements of the research include the creation of a robust algorithm that can be effectively applied to image processing with large amounts of noise and artifacts. This is particularly relevant to biological and medical analysis tasks where image quality often leaves much to be desired. The applicability of this research to our work is to use the proposed methods to improve the accuracy and robustness of deep learning models in detecting people and technical objects in complex environments such as water, roads and other environments. The results and conclusions of this work will be used to optimize our models and adapt them to different application scenarios, thus ensuring high accuracy and robustness of monitoring and security systems. Thus, the study provides valuable data and methods that can be directly applied in our research to improve the performance of neural network models, providing high accuracy and reliability in real-world scenarios.

In their work, Bilous et al. [19] presented methods for detecting and comparing body poses in video streams, focusing on the evaluation of different libraries and methods designed to analyze and compare body poses in real time. The authors investigated the performance and accuracy of libraries such as OpenPose, PoseNet, and BlazePose in order to identify the most effective solutions for pose recognition. They emphasized that accurate detection and comparison of body poses is important for applications such as exercise monitoring, safety, and medical diagnostics. The research involved testing each of the libraries on different datasets to evaluate their performance and accuracy. BlazePose combined with the weighted distance method performed best, outperforming OpenPose and PoseNet on a number of measures. The authors emphasize that the methods and algorithms proposed in their work can be used in a variety of applications, including video surveillance systems, exercise control, and medical monitoring. The results of the research confirm that BlazePose is a powerful and efficient solution for real-time pose recognition. In conclusion, the authors emphasize the importance of selecting appropriate libraries and methods to achieve high accuracy and performance in real-world applications. In their work, Bilous N. et al. [20] presented a method for tracking human head movements using control points. This method reduces the number of vectors to be recorded to the minimum number needed to describe head movements. The study includes a comparison of existing face vector detection methods and demonstrates the advantage of regression methods, which show significant accuracy and independence from illumination and partial face occlusion. The main goal of the work is to create a method that can track head movements and record only significant head direction vectors. The authors propose a control point method that significantly reduces the set of head direction vectors describing motion. According to the results, the regression-based methods showed significantly better accuracy and independence from illumination and partial face occlusion. The results of the study confirm the applicability of the control point method for human motion tracking and show that head vector detection methods from 2D images can compete with RGBD-based methods in terms of accuracy. Thus, when combined with the proposed approach, these methods present fewer limitations in use than RGBD-based methods. This method can be useful in various fields such as human-computer interaction, telemedicine, virtual reality, and 3D sound reproduction. Moreover, head position detection can be used to compare the exercises performed by a person to a certain standard, which is useful for rehabilitation facilities, fitness centers, and entertainment venues.

3. Materials and Methods

3.1. Datasets

Four main datasets, COCO2017, SARD, SeaDronesSee and VisDrone2019, were used to train the selected neural networks and evaluate their performance in different environments. These datasets provide unique images and annotations that allow the models to learn to recognize and classify objects in diferent scenarios. COCO2017 (Common Objects in Context) [21] is one of the most widely used and well-known datasets in computer vision. It includes over 200,000 images annotated for more than 80 object categories such as people, cars, bicycles, animals, and household objects. The annotations include not only bounding boxes but also segmentation masks, allowing COCO2017 to be used for object detection and segmentation tasks. This dataset is characterized by a variety of

 doi:10.20944/preprints202409.1310.v

scenes and objects collected in real-world environments with different variations in lighting, poses, and backgrounds. This makes COCO2017 ideal for comprehensive evaluation of object recognition algorithms and provides models with a rich training experience. SARD[22] is a specialized dataset designed for maritime object detection tasks. The dataset includes drone and ship images that cover a wide range of scenarios related to search and rescue operations. SARD includes annotations for various objects such as boats, life rafts, and people in the water. The images in this dataset vary in lighting and weather conditions, allowing models to learn to recognize objects in complex and variable marine environments. This dataset is particularly valuable for the development of algorithms designed for marine rescue and aquatic monitoring. The SeaDronesSee [23] dataset is designed for monitoring aquatic areas using drones. It contains images of people on water, various marine objects and natural phenomena such as waves and foam. Annotations in SeaDronesSee include object detection and segmentation tasks, allowing models to recognize and classify different objects in complex aquatic environments. This dataset is used to develop algorithms that can perform effectively under the changing lighting and dynamic scenes that characterize marine environments. SeaDronesSee provides realistic scenarios for testing and improving models aimed at safety and rescue in the water. VisDrone2019 is one of the largest and most diverse datasets collected from drones. It includes images captured under a variety of lighting and weather conditions, making it ideal for evaluating algorithm performance in complex traffic scenes and urban environments. VisDrone2019 [24] contains annotations for more than 10 object categories, including cars, pedestrians, cyclists, and other vehicles. The dataset is used for object detection, tracking, and segmentation tasks, allowing for a comprehensive evaluation of the algorithms in real-world environments. Annotations include bounding boxes, segmentation masks, and object traces, making it one of the most comprehensive and versatile datasets for drone-related computer vision tasks.

3.2. Data Preparation

Data preparation is a critical step to ensure successful training of neural network models. In this study, several key steps were performed to prepare the data, which ensured high quality input data and contributed to improved model performance. The first step was image preprocessing. All images were normalized to bring them to a uniform scale and range of pixel values. Normalizing the images improved the stability and speed of model training by reducing the impact of differences in brightness and contrast. In addition, all images were resized to a uniform size, ensuring compatibility with the neural network model architecture and allowing the use of fixed-size batches. Data augmentation played a key role in increasing the size of the training set and improving the generalization ability of the models. Various techniques such as rotation, scaling, shifting, changing the brightness and contrast of the images were used in the augmentation process. These techniques generated additional examples from the original images, which reduced the risk of overtraining the models and improved their ability to recognize objects under different conditions.

After data preprocessing and augmentation, the datasets were divided into three parts: training, validation and test samples in an 80:10:10 ratio. The training sample was used to directly train the models, the validation sample was used to monitor the training process and prevent overtraining, and the test sample was intended for the final evaluation of the models' performance. The partitioning of the data into these samples was done randomly, while maintaining the proportions of object classes in each part, which ensured an unbiased evaluation of the models. Special attention was paid to the processing of annotations for images. The annotations contained information about the location and types of objects in the images (Figure 1), which is necessary to train the models on detection and classification tasks. The image shows that the partitioning includes both foreground and background, which provides data diversity and helps the model learn to distinguish objects in complex environments. This is especially important when creating datasets for real-time autonomous control and monitoring tasks, where recognition accuracy plays a key role. This annotation helps models improve object recognition in complex scenes where overlaps, partial visibility, and different viewing angles are present. All annotations were brought to a common format compatible with the deep learning frameworks used, which simplified the process of loading and using the data in the

 doi:10.20944/preprints202409.1310.v

models. Special scripts were developed to efficiently load the data into memory and transfer it to the GPU. These scripts provided fast and seamless loading of large amounts of data, minimizing latency and providing high performance when training models. The scripts also included methods for dynamic data augmentation during training, further increasing the diversity of training examples and improving the generalizability of the models.

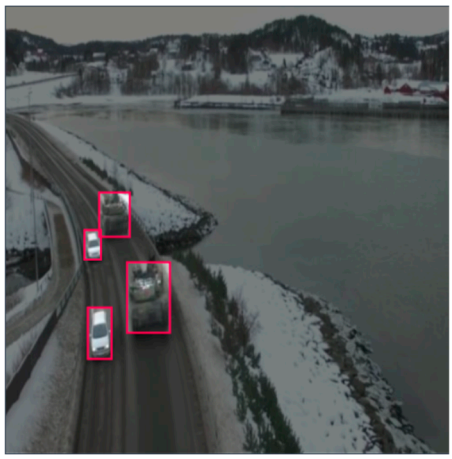

Figure 1. Annotated image of a technical object for object detection model training.

Utilizing an integrated approach to data training that included preprocessing, augmentation, proper partitioning into samples, and efficient data loading played a key role in achieving high model performance. This approach allowed us to create high-quality input data for training, which in turn ensured the stability and accuracy of neural network models under different conditions.

3.3. Neural Network Models

For this research, several modern neural network models have been selected, each with unique characteristics and advantages that make them suitable for different object detection tasks. The models considered include YOLOv4-v8, Faster R-CNN, SSD, and EfficientDet. The YOLO model was chosen for its high-speed image processing and real-time capability. This model is characterized by the fact that it processes the whole image in one pass of the network, which allows high speed. YOLOv4-v8 versions were used in this study. Each subsequent version of YOLO brought improvements in accuracy and robustness to different lighting conditions and noise, making these models the most productive in real-world environments. YOLOv4 [25] offered improvements over previous versions by introducing a new detection method and utilizing techniques such as CSPNet and PANet, greatly improving the accuracy and speed of the model. In the YOLOv5 [26] version, image processing techniques and learning algorithms have been improved to achieve even better performance. The YOLOv6 [27] and YOLOv7 [28] versions continue to enhance the model architecture, including improvements in handling different object scales and optimizing computational resources. YOLOv8 [29] is the latest version in this study, which integrates the latest developments in deep learning and provides the highest accuracy and speed among all YOLO versions.

The Faster R-CNN model was chosen for its high accuracy in object detection tasks. The main advantage of Faster R-CNN is the use of Region Proposal Network (RPN), which significantly speeds up the object detection process compared to previous versions of R-CNN. However, despite its high accuracy, the processing speed of this model is slower compared to YOLO, which limits its application in tasks requiring real-time processing. Faster R-CNN uses RPN to generate region suggestions, which are then processed by a convolutional network to accurately localize and classify objects. The model finds wide application in tasks where high accuracy is important, such as video surveillance, autonomous driving, and image analysis in medical applications.

SSD (Single Shot MultiBox Detector) was selected for its focus on high-speed image processing. The model utilizes multiple feature maps to detect objects at different scales to achieve a high level

 doi:10.20944/preprints202409.1310.v

of accuracy with low latency. SSD is widely used in tasks related to autonomous driving and robotics due to its ability to work efficiently in real-world environments. The SSD utilizes multiple feature maps of different resolutions, which enables the model to efficiently detect objects of different sizes and scales in the image. The SSD model demonstrates high performance with low latency, making it suitable for applications requiring real-time processing.

EfficientDet is a model based on the EfficientNet architecture that emphasizes computational and resource efficiency. This model is ideal for use in environments with limited computational resources. EfficientDet exhibits high accuracy and performance, especially in the context of mobile and embedded systems, making it an important component for resource-intensive tasks. EfficientDet utilizes a novel BiFPN (Bidirectional Feature Pyramid Network) architecture to efficiently fuse features from different layers to achieve high accuracy at low computational cost. EfficientDet is widely used in mobile and embedded systems where it is important to minimize resource consumption while maintaining high object recognition accuracy.

Each of these models was trained and evaluated using training datasets. Popular deep learning frameworks such as TensorFlow and PyTorch were used in the training process. The main metrics for evaluating the performance of the models were mAP (mean accuracy), precision, accuracy, completeness, F1-Score and FPS (frames per second), which enabled a comprehensive evaluation of both object recognition accuracy and the models' ability to perform in real-world environments. Thus, the study provided a comparative analysis of the performance of state-of-the-art neural network models, which allowed us to identify their strengths and weaknesses in different application scenarios.

3.4. Training Models

Training neural network models is a key step in the neural network development and optimization process. In this study, we considered various state-of-the-art neural network architectures, including YOLOv4-v8, Faster R-CNN, SSD, and EfficientDet, to determine their performance in human and technical object detection tasks in different environments. Experiments were conducted for each model to train and evaluate them. The model training process involved several steps. First, the models were initialized with pre-configured hyperparameters. They were then trained on the training dataset, with regular evaluation on the validation dataset to monitor progress and prevent overfitting. After training was completed, the models were evaluated on the test dataset on key metrics such as mAP (mean accuracy), precision, accuracy, completeness, F1-Score and FPS (frames per second). These metrics allowed for a comprehensive evaluation of both object recognition accuracy and the models' ability to perform in real-world conditions.

3.4.1. Computational Platform

To ensure efficient training and performance evaluation of neural network models, a high-performance computing platform was utilized in this study. The choice of hardware and software plays a critical role in performing complex deep learning tasks, especially when dealing with large datasets and complex model architectures. Compute nodes med-IoT-KI serve equipped with powerful 4x NVIDIA Tesla A100 GPUs with 80 GB of memory were used to train the models. This GPU was chosen for its high performance and ability to process large amounts of data. The CPU used was a dual-processor AMD EPYC 7413 with 24 cores at 2.65 GHz, providing multi-threading and high performance for data processing and GPU coordination. The system was also equipped with 512GB of RAM to handle large batches of data and maintain high processing speeds. For data storage, 2TB SSD disks were used to provide fast access to data and models, reducing latency during training. Modern software tools and libraries were used for model development, training and evaluation. The operating system chosen was Ubuntu 20.04 LTS for its stability, security, and extensive support for scientific computing tools. The deep learning frameworks used were TensorFlow 2.4 and PyTorch 1.8, which provide powerful tools for developing and training neural networks and also support GPU operation. The NumPy, Pandas, OpenCV and Scikit-learn libraries were used for data preprocessing, result analysis and visualization. Environment optimization and tuning was performed to ensure

 doi:10.20944/preprints202409.1310.v

stable and efficient operation of the computing platform. NVIDIA CUDA 11.2 and cuDNN 8.1 drivers were installed to ensure compatibility with the used GPUs and deep learning frameworks. TensorFlow and PyTorch were configured to use all available GPUs, maximizing performance and speeding up model training. All necessary libraries and dependencies were installed and updated to ensure their relevance and compatibility with deep learning frameworks.

Preparing the data for training involved several key steps. First, the OpenCV and NumPy libraries were used to normalize, resize, and augment the images. Next, the datasets were divided into training, validation, and test samples to provide an unbiased assessment of model performance. Finally, scripts were developed to efficiently load the data into memory and transfer it to the GPU for model training. The use of a high-performance computing platform and optimized software significantly accelerated the model training process and ensured high accuracy and stability of the results. The environment setup and data preparation played a key role in achieving high performance, which confirms the importance of choosing the right hardware and software tools for deep learning tasks.

3.4.2. Configuring Hyperparameters

Hyperparameter tuning is a key step in the training of neural network models, as choosing the right hyperparameter values directly affects the performance, accuracy, and generalization ability of the model. In this study, optimal hyperparameter values were selected for each of the models based on multiple experiments and analyzing the results. The initial learning rate for all models was set at 0.01. This value provided a sufficient convergence rate, allowing the model to train quickly while preventing jump changes in weights that can occur if the learning rate is too high. During training, the learning rate could be adjusted based on monitoring the errors and convergence of the model to ensure the best results. The batch size, which determines the number of images processed per iteration, was chosen to be 64. This value was chosen considering the availability of computational resources and the necessary balance between gradient estimation accuracy and memory utilization efficiency. Larger packet sizes produce more accurate gradient estimates but require more memory, whereas smaller packets produce noisier estimates but require fewer resources. The number of epochs to train each model was set to 30. This value was chosen based on learning curve analysis and validation. An insufficient number of epochs can lead to undertraining of the model when it fails to learn all the patterns in the data. On the other hand, too many epochs can lead to overfitting when the model starts to overfit to the training data, losing its ability to generalize to new data. Therefore, 30 epochs was the optimal value, providing a balance between training and generalization. The Adam optimizer, which combines the advantages of adaptive learning rate and moment methods, was used to update the model weights. Adam automatically adjusts the learning rate for each parameter, allowing the model to converge faster and adapt to different types of data. This optimizer has shown high performance and stability on a wide range of deep learning tasks.

To prevent overfitting in the YOLOv4-v8 models, a Dropout method with a neuron outage probability of 0.5 was used. This method consists of randomly turning off a certain percentage of neurons during training, which prevents the model from being dependent on specific features and improves its generalization ability. Dropout helps models avoid overfitting and improves their performance on validation and test data. The Faster R-CNN model used SGD (Stochastic Gradient Descent) optimizer with a momentum of 0.9. This optimizer allows past gradients to be taken into account when updating the weights, resulting in more stable and faster training. L2-regularization with a factor of 0.0005 was also applied to the model to prevent increasing weights and overfitting. L2-regularization adds a penalty to the loss function for large values of weights, which helps to control the complexity of the model and improve its generalization ability. Dropout with probability 0.4 was applied in the SSD model, which helped to improve the generalization ability of the model. The RMSprop optimizer used in this model provided adaptive updating of weights and good convergence. RMSprop automatically adjusts the learning rate for each parameter, allowing the model to adapt to the data faster and improve its performance. EfficientDet, based on the EfficientNet architecture, used L2 regularization with a factor of 0.0001 to prevent increasing weights and

 doi:10.20944/preprints202409.1310.v

overfitting. This regularization helped control the complexity of the model and improve its ability to generalize. EfficientDet emphasized computational and resource efficiency, making it ideal for use in computationally constrained environments.

An early stopping method was used for all models, which stopped training if performance on the validation dataset stopped improving over 10 epochs. This method avoided unnecessary training time and prevented overfitting of models, while ensuring high accuracy and stability. Hyperparameter tuning was conducted by trial and error using cross-validation to ensure the best results. Optimizing the hyperparameters achieved a balance between accuracy, performance, and generalization ability of the models, which was critical for real-world object recognition tasks.

4. Results

In this research, experiments were carried out to compare the performance of different neural network models for object detection tasks including people and technical objects (military vehicles, cars) in different environments such as water, road and other complex environments. YOLOv4-v8, Faster R-CNN, SSD and EfficientDet models were considered. The use of trained datasets COCO2017, SARD, SeaDronesSee and VisDrone2019 allowed for a comprehensive evaluation of the models under different conditions and scenarios.

4.1. Performance Evaluation

Experimental results showed that the models exhibit different levels of performance in human and technical object detection tasks under different conditions. The table below summarizes the comparative accuracy, completeness, average accuracy at threshold 0.5 (mAP@0.5) and F1-Score for each architecture after 30 training epochs (Table 1):

Table 1. Architectures performance.

Architecture	Precision	Recall	mAP@0.5	F1-Score
YOLOv4	0.81	0.83	0.82	0.82
YOLOv5	0.73	0.73	0.73	0.73
YOLOv6	0.62	0.62	0.62	0.62
YOLOv7	0.68	0.68	0.68	0.68
YOLOv8	0.87	0.89	0.88	0.88
Faster R-CNN	0.84	0.82	0.83	0.83
SSD	0.76	0.75	0.75	0.75
EfficientDet	0.82	0.80	0.81	0.81

From the above data, we can see that YOLOv8 demonstrates the highest performance among all models, showing the best results in all key metrics. Faster R-CNN also shows high accuracy but is inferior to YOLOv8 in processing speed.

4.2. Processing Speed

One of the key aspects for real-time object detection tasks is processing speed. The table below shows the FPS (frames per second) for each architecture (Table 2):

Table 2. Processing speed for each architecture

Architecture	FPS
YOLOv4	40
YOLOv5	45
YOLOv6	42
YOLOv7	43
YOLOv8	48
Faster R-CNN	10

 doi:10.20944/preprints202409.1310.v

SSD	35
EfficientDet	30

YOLOv8 demonstrates the highest processing speed among all architectures, which makes it the most suitable for tasks requiring real time. Faster R-CNN, despite its high accuracy, is significantly inferior in terms of speed, which limits its application in such tasks.

4.3. Classification Errors

The analysis of object classification errors has revealed important aspects for model optimization. Partially detected objects, misclassification and completely missed objects are typical errors that occur during the detection of people and technical objects in different environments. The table below summarizes the error analysis for different models (Table 3):

Table 3. Detection errors for each architecture.

Architecture	Partially detected	Misclassified	Missed
YOLOv4	10%	7%	6%
YOLOv5	11%	8%	6%
YOLOv6	21%	12%	11%
YOLOv7	10%	6%	6%
YOLOv8	8%	4%	4%
Faster R-CNN	9%	5%	6%
SSD	14%	9%	8%
EfficientDet	11%	7%	7%

YOLOv8 significantly outperforms other models in all error rates, which indicates its high efficiency and accuracy in the task of detecting people and technical objects in various environments. Figure 2 shows a street scene where the YOLOv8 model recognized several cars. On the left is a black car, which the model recognized with probability 0.92, circled by an orange rectangle and labeled “car”. In front of the black car is a silver car, recognized by the model with probability 0.66, also circled by an orange rectangle and labeled “car”. Partially visible in the right foreground is a silver car, identified by the model with a probability of 0.93, circled by a green rectangle and labeled “truck”. The image demonstrates how YOLOv8 performs in a complex street traffic environment. It is important to note here that YOLOv8 is able to handle objects of different types and sizes, detecting them in real time. This allows YOLOv8 to be used in traffic control systems, autonomous vehicles and security applications where instant and accurate object recognition is required. The scene shown also highlights YOLOv8's ability to handle different lighting and background conditions, which is essential for urban applications. This allows the system to recognize and respond quickly to dynamic changes in the road conditions.

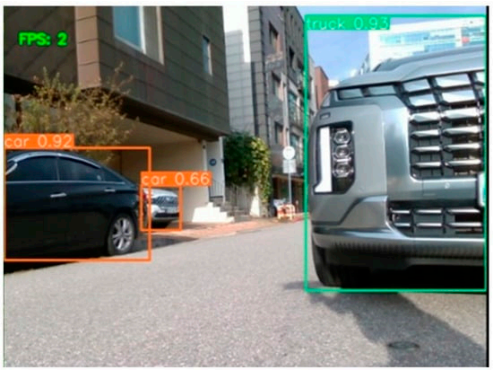

Figure 2. Detection and classification of vehicles on an urban road scene using YOLOv8 model.

 doi:10.20944/preprints202409.1310.v

Figure 3 shows a scene at a military training ground where the YOLOv8 model recognized two tanks. In the foreground is a large tank, which the model identified with a probability of 0.925. The tank is circled by a red rectangle and labeled “tank”. In the distance, on the road leading up the hill, you can see a second tank, recognized by the model with a probability of 0.907, also circled by a red rectangle and labeled “tank”.

The image demonstrates the YOLOv8 model's performance in recognizing and classifying military vehicles in open terrain, with probabilities for each recognized object.

Figure 3. Detection and classification of military vehicles on a terrain using YOLOv8 model.0.

Experimental results confirmed that YOLOv8 is the most effective model for detecting people and technical objects in different environments, providing high accuracy, speed and ability to adapt to different scenarios. Faster R-CNN showed high accuracy, but its low processing speed limits its application in real time. SSD and EfficientDet also showed decent results but are inferior to YOLOv8 in terms of accuracy and processing speed. Thus, YOLOv8 is the preferred choice for security monitoring and complex environment analysis tasks, providing new opportunities for further improvements and developments in computer vision and deep learning.

5. Discussion

The results of our study showed that YOLOv4-v8, Faster R-CNN, SSD, and EfficientDet models exhibit different levels of performance in human and technical object detection tasks in different environments. YOLOv8 proved to be the most efficient model, providing high accuracy and processing speed, making it the preferred choice for real-time tasks. Faster R-CNN showed high accuracy, but its low processing speed significantly limits its application in tasks where responsiveness is important. SSD and EfficientDet also performed well but are inferior to YOLOv8 in challenging environments such as water and road scenes. Our experiments showed that environmental conditions have a significant impact on the performance of the models. For example, complex water scenes with reflections and variable lighting conditions pose additional challenges for the models. In such environments, YOLOv8 performed best due to its ability to adapt to a variety of conditions and its high object recognition accuracy. On roads with many moving objects and a variety of backgrounds, YOLOv8 also performed well, outperforming other models in terms of accuracy and processing speed. This makes it ideal for applications in traffic monitoring systems and autonomous vehicles. The classification error analysis showed that the main problems arise from partially detected objects, misclassification and missed objects. Partially detected objects are often found in complex environments such as water, where reflections and variable lighting conditions are present. Misclassification can be due to insufficient image clarity, or the complexity of scenes where multiple objects are present. Missing objects are more common in low visibility conditions or where there are multiple overlaps. Despite the high performance of YOLOv8, there are several areas for further improvements. One of them is to improve the handling of complex scenes with many objects and overlaps. This can be achieved by using additional training data and applying reinforcement learning techniques. It is also worth considering using more complex architectures to improve accuracy and reduce classification errors. The models used in this study have a wide range of applications in various fields. YOLOv8, due to its high accuracy and processing speed, is ideal for security

 doi:10.20944/preprints202409.1310.v

monitoring tasks such as water and road surveillance, as well as for applications in autonomous vehicle systems. Faster R-CNN, despite its low processing speed, can be effective in applications where high accuracy is required and real-time is not necessary, such as medical imaging and video surveillance. SSD and EfficientDet can also be useful in computationally constrained environments due to their efficiency and speed. In conclusion, our study shows that YOLOv8 is the most effective model for human and technical object detection tasks in various environments due to its high accuracy, processing speed and ability to adapt to a variety of scenarios. Faster R-CNN, SSD and EfficientDet also performed well, but are inferior to YOLOv8 in key performance metrics. These results emphasize the importance of selecting the right model for specific tasks and conditions, which is critical for the successful application of computer vision and deep learning technologies in real-world scenarios.

6. Conclusions

In this research, comprehensive experiments were performed to compare the performance of different neural network models in human and technical object detection tasks under a variety of conditions. The models considered included YOLOv4-v8, Faster R-CNN, SSD and EfficientDet. The experimental results allowed us to draw several key conclusions. First, the YOLOv8 model performed the best among all the models considered, demonstrating high accuracy and processing speed. These characteristics make it the preferred choice for real-time tasks such as water and road safety monitoring and autonomous vehicle systems. YOLOv8 has also shown high robustness to challenging environmental conditions such as reflections on water and variable lighting conditions. Second, Faster R-CNN demonstrated high object recognition accuracy, but its low processing speed limits real-time applications. This model can be effective in applications where high accuracy is required but processing speed is not critical, such as medical imaging and video surveillance. SSD and EfficientDet also performed well, especially in computationally constrained environments. These models strike a balance between accuracy and performance, making them suitable for mobile and embedded systems where resource efficiency is important. The analysis of classification errors showed that the main problems arise from partially detected objects, misclassification and missing objects. These errors are most common in complex environments such as water and road scenes with many objects. Further improvement of the models requires a focus on making them more robust to such conditions, which can be achieved by using additional training data and applying reinforcement learning techniques. Data training played a key role in the success of model training. Normalization, augmentation, and proper partitioning of data into training, validation, and test samples ensured high quality input data and contributed to improved model performance. Efficient data loading and optimization of the computational platform also played an important role in achieving high performance. In conclusion, the results of this study emphasize the importance of selecting the appropriate model for specific tasks and conditions. YOLOv8 is the most effective model for human and technical object detection tasks in various environments due to its high accuracy, processing speed, and ability to adapt to a variety of scenarios. These findings are critical for the successful application of computer vision and deep learning technologies in real-world scenarios, providing new opportunities for further improvements and developments in this field..

7. Acknowledgments

The research was performed in collaboration with research laboratory of Kharkiv National University of Radio Electronics and Technical University of Applied Sciences Wildau and financial support of Volkswagen Foundation. Server used for research is med-IoT-KI, TH Wildau, project 85053663 "AI-supported IoT environment for medical examinations" MWFK program to promote the infrastructure for research, development and innovation (InfraFEI) with funds from the European Regional Development Fund (ERDF).

References

1. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); IEEE: Las Vegas, NV, USA, June 2016; pp. 779–788.
2. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks 2015.
3. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. **2015**, doi:10.48550/ARXIV.1512.02325.
4. Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and Efficient Object Detection. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); IEEE: Seattle, WA, USA, June 2020; pp. 10778–10787.
5. Reddy, S.; Pillay, N.; Singh, N. Comparative Evaluation of Convolutional Neural Network Object Detection Algorithms for Vehicle Detection. *J. Imaging* **2024**, *10*, 162, doi:10.3390/jimaging10070162.
6. Sun, C.; Chen, Y.; Qiu, X.; Li, R.; You, L. MRD-YOLO: A Multispectral Object Detection Algorithm for Complex Road Scenes. *Sensors* **2024**, *24*, 3222, doi:10.3390/s24103222.
7. Dai, Y.; Kim, D.; Lee, K. An Advanced Approach to Object Detection and Tracking in Robotics and Autonomous Vehicles Using YOLOv8 and LiDAR Data Fusion. *Electronics* **2024**, *13*, 2250, doi:10.3390/electronics13122250.
8. Bilous, N.; Svidin, O.; Ahekian, I.; Malko, V. A Skeleton-Based Method for Exercise Recognition Based On 3D Coordinates of Human Joints. *IJ-AI* **2024**, *13*, 581, doi:10.11591/ijai.v13.i1.pp581-596.
9. Daud Khan; Muhammad Waqas; Mohsin Tahir; Shahab Ul Islam; Muhammad Amin; Atif Ishtiaq; Latif Jan Revolutionizing Real-Time Object Detection: YOLO and MobileNet SSD Integration. *JCBI* **2023**, *6*, 41–49.
10. Suryavanshi, D. A Comparative Study of Object Detection Using YOLO and SSD Algorithms. *IJSREM* **2023**, *07*, doi:10.55041/IJSREM25639.
11. Akshatha, K.R.; Karunakar, A.K.; Shenoy, S.B.; Pai, A.K.; Nagaraj, N.H.; Rohatgi, S.S. Human Detection in Aerial Thermal Images Using Faster R-CNN and SSD Algorithms. *Electronics* **2022**, *11*, 1151, doi:10.3390/electronics11071151.
12. Zhang, H.; Shao, F.; Chu, W.; Dai, J.; Li, X.; Zhang, X.; Gong, C. Faster R-CNN Based on Frame Difference and Spatiotemporal Context for Vehicle Detection. *SIViP* **2024**, doi:10.1007/s11760-024-03370-3.
13. Gao, S. Exploration and Evaluation of Faster R-CNN-Based Pedestrian Detection Techniques. *ACE* **2024**, *32*, 185–190, doi:10.54254/2755-2721/32/20230208.
14. Omiotek, Z.; Zhunissova, U. Dangerous Items Detection in Surveillance Camera Images Using Faster R-CNN. *Preprints* **2024**, doi:10.20944/preprints202406.1090.v1.
15. Bai, T.; Luo, J.; Zhou, S.; Lu, Y.; Wang, Y. Vehicle-Type Recognition Method for Images Based on Improved Faster R-CNN Model. *Sensors* **2024**, *24*, 2650, doi:10.3390/s24082650.
16. Wang, Y.; Wang, T.; Zhou, X.; Cai, W.; Liu, R.; Huang, M.; Jing, T.; Lin, M.; He, H.; Wang, W.; et al. TransEffiDet: Aircraft Detection and Classification in Aerial Images Based on EfficientDet and Transformer. *Computational Intelligence and Neuroscience* **2022**, *2022*, 1–10, doi:10.1155/2022/2262549.
17. Munteanu, D.; Moina, D.; Zamfir, C.G.; Petrea, Ștefan M.; Cristea, D.S.; Munteanu, N. Sea Mine Detection Framework Using YOLO, SSD and EfficientDet Deep Learning Models. *Sensors* **2022**, *22*, 9536, doi:10.3390/s22239536.
18. Hramm, O.; Bilous, N.; Ahekian, I. Configurable Cell Segmentation Solution Using Hough Circles Transform and Watershed Algorithm. In Proceedings of the 2019 IEEE 8th International Conference on Advanced Optoelectronics and Lasers (CAOL); IEEE: Sozopol, Bulgaria, September 2019; pp. 602–605.
19. Bilous, N.V.; Ahekian, I.A.; Kaluhin, V.V. DETERMINATION AND COMPARISON METHODS OF BODY POSITIONS ON STREAM VIDEO. *RIC* **2023**, 52, doi:10.15588/1607-3274-2023-2-6.
20. Rakova, A.O.; Bilous, N.V. REFERENCE POINTS METHOD FOR HUMAN HEAD MOVEMENTS TRACKING. *RIC* **2020**, *0*, 121–128, doi:10.15588/1607-3274-2020-3-11.
21. Lin, T.-Y.; Maire, M.; Belongie, S.; Bourdev, L.; Girshick, R.; Hays, J.; Perona, P.; Ramanan, D.; Zitnick, C.L.; Dollár, P. Microsoft COCO: Common Objects in Context 2015.
22. Sambolek, S.; Ivasic-Kos, M. SEARCH AND RESCUE IMAGE DATASET FOR PERSON DETECTION - SARD 2021.
23. Varga, L.A.; Kiefer, B.; Messmer, M.; Zell, A. SeaDronesSee: A Maritime Benchmark for Detecting Humans in Open Water. **2021**, doi:10.48550/ARXIV.2105.01922.
24. Zhu, P.; Wen, L.; Du, D.; Bian, X.; Fan, H.; Hu, Q.; Ling, H. Detection and Tracking Meet Drones Challenge 2021.
25. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection 2020.
26. Jocher, G. Ultralytics YOLOv5 2020.
27. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications. **2022**, doi:10.48550/ARXIV.2209.02976.

 doi:10.20944/preprints202409.1310.v

28. Wang, C.-Y.; Bochkovskiy, A.; Liao, H.-Y.M. YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. *arXiv preprint arXiv:2207.02696* **2022**.
29. Jocher, G.; Chaurasia, A.; Qiu, J. Ultralytics YOLOv8 2023.