information

Article

Deep Learning-Based Multiple Droplet Contamination Detector for Vision Systems Using a You Only Look Once Algorithm

Youngkwang Kim [1,†], Woochan Kim [1,†], Jungwoo Yoon [1], Sangkug Chung [1,*] and Daegeun Kim [2,*]

[1] Department of Mechanical Engineering, Myongji University, Yongin 17058, Republic of Korea; ygkim@mju.ac.kr (Y.K.); wckim@mju.ac.kr (W.K.); yoon000912@mju.ac.kr (J.Y.)
[2] Microsystems, Inc., Yongin 17058, Republic of Korea
* Correspondence: skchung@mju.ac.kr (S.C.); dgkim@microsystems.co.kr (D.K.)
† These authors contributed equally to this work.

Abstract: This paper presents a practical contamination detection system for camera lenses using image analysis with deep learning. The proposed system can detect contamination in camera digital images through contamination learning utilizing deep learning, and it aims to prevent performance degradation of intelligent vision systems due to lens contamination in cameras. This system is based on the object detection algorithm YOLO (v5n, v5s, v5m, v5l, and v5x), which is trained with 4000 images captured under different lighting and background conditions. The trained models showed that the average precision improves as the algorithm size increases, especially for YOLOv5x, which showed excellent efficiency in detecting droplet contamination within 23 ms. They also achieved an average precision (mAP@0.5) of 87.46%, recall (mAP@0.5:0.95) of 51.90%, precision of 90.28%, recall of 81.47%, and F1 score of 85.64%. As a proof of concept, we demonstrated the identification and removal of contamination on camera lenses by integrating a contamination detection system and a transparent heater-based cleaning system. The proposed system is anticipated to be applied to autonomous driving systems, public safety surveillance cameras, environmental monitoring drones, etc., to increase operational safety and reliability.

Keywords: object detection; classification; contamination detection; autonomous driving systems; machine leaning

Citation: Kim, Y.; Kim, W.; Yoon, J.; Chung, S.; Kim, D. Deep Learning-Based Multiple Droplet Contamination Detector for Vision Systems Using a You Only Look Once Algorithm. *Information* **2024**, *15*, 134. https://doi.org/10.3390/info15030134

Academic Editors: Dejiu Chen, Fredrik Warg, Anders Thorsén and Anders Cassel

Received: 31 January 2024
Revised: 19 February 2024
Accepted: 27 February 2024
Published: 28 February 2024

1. Introduction

Traffic accidents are one of the primary causes of death worldwide. These accidents are often caused by factors such as driver inattentiveness, failure to follow traffic rules, and distractions [1,2]. Particularly, the lack of safety mechanisms in vehicles has been identified as a principal factor contributing to traffic accidents [3]. For these reasons, the automobile industry is adopting electronic safety devices. The U.S. National Highway Traffic Safety Administration (NHTSA) passed a law requiring vehicles to have rear visibility starting in 2018 [4], and Europe is enacting similar regulations to enhance vehicle safety features.

As a result, the automotive industry is rapidly growing research vision-based recognition systems that use optical sensors, like cameras and lidar, which act as the vehicle's 'eyes'. These systems assist in safe driving by detecting the driving environment, objects, and potential hazards and by providing warning signals [5]. Such capabilities are essential for making safe decisions and responding quickly, even in abnormal situations. The effectiveness of these systems is highly dependent on the quality of image acquisition devices such as cameras [6,7]. However, these devices are vulnerable to contamination, such as rain, snow, and fog, due to exposure to the external environment.

To solve this issue, various active cleaning technologies are being developed. These technologies employ methods such as electrowetting [8,9], surface acoustic waves [10,11], and heat [12,13] to remove contamination from lens surfaces. For instance, Lee et al. [14]

developed a cleaning device using electrowetting, applying an electrical signal to control the interfacial tension of droplets on the surface, thus moving and removing them. Song et al. [15] developed a cleaning device that uses surface acoustic waves and applies an electric signal to the device to push and remove droplets in the direction of surface acoustic waves. Park et al. [16] developed a cleaning device using heat to evaporate and remove droplets on the surface. As such, various active cleaning technologies are being developed, and the application of these devices in real-world scenarios requires the development of contamination detection technology so that the cleaning device can detect contamination and operate on its own.

Robbins and Nelson [17] developed a system for detecting the presence of contamination on the lens of cameras equipped with digital image sensors. This system comprises a light source, an image sensor, and a light source located between the lens cover and the digital image sensor. When contamination occurs on the lens, the light emitted from the light source is scattered by the contamination. The image sensor detects this scattered light and converts it into a digital signal. Utilizing this signal, the system determines the presence of lens contamination and provides a warning to the user. Zhang et al. [18] researched detecting camera lens contamination using a static region segmentation algorithm and a wave decomposition-based blurred edge detection algorithm. The static region segmentation algorithm detects non-moving regions obscured by contamination in the images acquired through a camera, and the blurred edge detection algorithm identifies the blurred outlines of objects due to contamination. However, Robbins and Nelson's system requires precise control of the light source and sensitivity adjustment of the image sensor, and Zhang's algorithm requires complex processing to accurately identify static areas and blurry edges. This process requires a high level of technical knowledge and resources and has limitations, such as being inapplicable in a variety of lighting conditions. To overcome these issues, we attempted to solve these problems by utilizing deep learning algorithms that are easier to develop and more adaptable to different environments.

Recently, with the advancement of Graphic Processing Unit (GPU) technology, deep learning algorithms detection technologies have attracted attention that can precisely analyze various objects and situations in complex environments. Chi Cheng Lai et al. [19] developed a windshield rain detection system using deep learning on more than 150 k global background images of rainy situations. Huanjie Tao et al. [20] developed a pixel-level supervised learning neural network to build an advanced detection system that can recognize forest smoke. Similarly, Yining Cao et al. [21] developed an MCS-YOLO algorithm to implement a high-precision real-time object detection system optimized for autonomous driving environments. Despite these advances, there is a lack of development of specialized contamination detection systems that precisely analyze and recognize the location and characteristics of contamination to develop self-cleaning systems that can effectively clean various contaminants on the surface of a camera lens.

This paper proposes an advanced droplet contamination cleaning system for improving the visibility of automotive vision systems. This system consists of a transparent heater-based advanced droplet cleaning device and a deep learning-based contamination detection system utilizing YOLO (You Only Look Once), an object detection model based on CNN architecture [22,23]. The advanced droplet contamination cleaning system precisely recognizes the location of droplet contamination and enables customized cleaning technology based on it, enabling continuous high-definition image acquisition without degrading the performance of the camera lens. The operational scenario of this system is shown in Figure 1. If the camera installed in a vehicle is contaminated, the analytical model trained on the contamination receives this distorted image. The YOLO algorithm embedded within this model analyzes the image to determine the presence of contamination. When contamination is detected, the analysis image is transmitted to the vehicle's display to visually inform the driver of the contamination situation. Concurrently, a digital signal is transmitted to the vehicle's internal lens cleaning system to remove the contamination attached to the lens. This research focuses on using deep learning technology for the

detection of camera lens contamination. Ultimately, through a demonstration, it shows the enhancement of camera visibility through integration with a previously developed lens cleaning system [24].

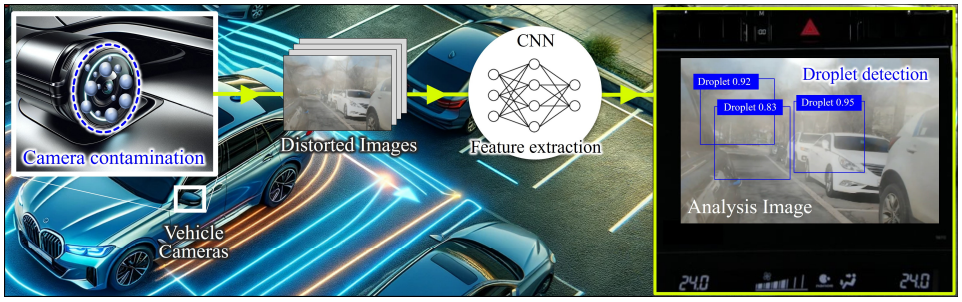

Figure 1. Scenario for the proposed deep learning-based contamination detection system.

2. Network

2.1. Convolution Neural Network (CNN)

A CNN is a deep learning algorithm inspired by the structure and functional principles of the human brain. Deep learning algorithms have evolved from Artificial Neural Networks (ANNs) to Deep Neural Networks (DNNs) and finally to CNNs. An ANN consists of an input layer that receives data, a hidden layer where data processing occurs, and an output layer that produces the final results. However, ANNs were limited to low accuracy and slow learning times. To solve these problems, DNNs were developed. DNNs are more complex, featuring two or more deep hidden layers [25,26]. DNNs autonomously create classification labels and categorize data, producing optimal results through repeated processes. While DNNs excel in object classification and recognition, they struggle with processing high-dimensional data that require extensive computation.

CNNs were developed based on the foundational concepts of DNNs to overcome these limitations [27,28]. CNNs are specifically structured to efficiently learn local features within an image. This structure reduces computational requirements and enhances image processing accuracy. The CNN architecture is used for image classification, recognition, and detection and is based on a paper by Yann LeCun et al. in 1998 [29]. The operation of CNN architecture is divided into two major steps (Figure 2) [30,31]. The first step is a feature extraction. This step includes a convolutional layer and a pooling layer. The convolutional layer uses a kernel filter to compute a convolutional operation on the input image to extract the underlying features. Pooling reduces the dimensionality of the feature data and converts the two-dimensional data into a one-dimensional array. The second step is classification, in which the fully connected layer learns the relationships between the extracted features and uses them to determine which category the image belongs to, outputting a classification result. Through this process, CNN architecture contributes to reducing the complexity of the data while retaining important information, resulting in high accuracy and efficient image processing.

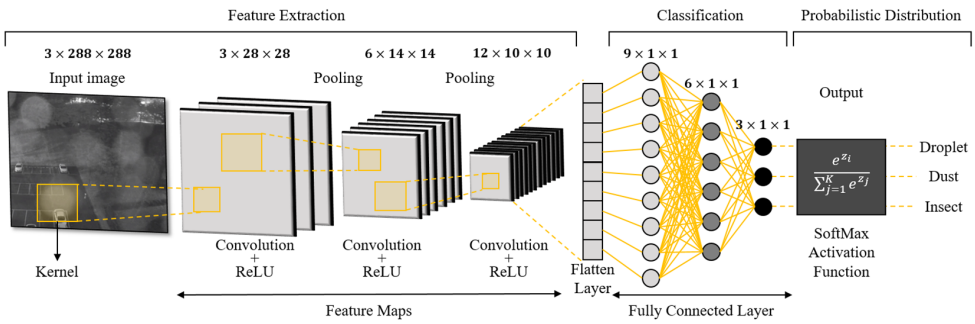

Figure 2. CNN architecture for image classification [32].

2.2. You Only Look Once (YOLO) [33]

The YOLO (You Only Look Once) series is a prominent example of a single-stage detector in object detection technology. Single-stage detectors generally comprise three main components: the backbone, neck, and head. The backbone extracts both low-level and high-level features from the images. The neck fuses the features extracted by the backbone. This process enriches the semantic information of the features and transfers them to the head, as well as bridging the gap between feature extraction (achieved by the backbone) and object detection (achieved by the head), ensuring a smooth transition of informative features. The head is the final stage of object detection and classification. This stage performs the head based on the features received from the neck, predicting the location and class of objects.

YOLO divides the image into grids and predicts the presence or absence of objects and their bounding boxes within each grid region. This treats the object detection problem as a single regression problem, allowing the model to predict the location and class of the object with just one inference on the input image. Each grid cell is responsible only for objects centered within its area and predicts many bounding boxes and a confidence score for them. Ultimately, each bounding box provides a five-dimensional output that represents the location of the object (coordinates) and the probability that the object exists. The five-dimensional outputs are the coordinates of the object's center point (x, y), the width and height of the bounding box (w, h), and the probability that the object exists within that box. Here, x and y are coordinates that represent the location of the center of the object within the image, and w and h are the width and height of the bounding box surrounding the object. Confidence represents the probability that an object exists within the box, which is a metric of the model's detection performance. These five dimensions of information allow YOLO to accurately predict the location, size, and probability of the existence of objects in an image. Early versions of YOLO were very fast in inference speed but had limitations in terms of accuracy. However, subsequent versions, such as YOLOv3, YOLOv4, and YOLOv5, have greatly improved accuracy through various technical improvements.

2.3. YOLOv5

YOLOv5 is widely favored in various research circles due to its seamless integration with specific libraries and frameworks, its ability to perform rapid inferences, and its notable accuracy [34–36]. An overview of the YOLOv5 architecture is presented in Figure 3. At its core, YOLOv5 employs CSPDarknet, an enhanced version of darknet, augmented with a cross-stage partial network (CSPNet) [37]. CSPNet efficiently addresses the issue of redundant gradient information in the network, enhancing the learning process while preserving accuracy and reducing complexity. The spatial pyramid pooling (SPP) block, a key component of the backbone, broadens the receptive field and isolates critical features from the base network. This block generates feature maps from the input image through its convolutional layers. The YOLOv5 architecture's neck features a Path Aggregation Network (PANet), which facilitates optimal information flow. PANet incorporates an innovative Feature Pyramid Network (FPN) design, with layers arranged in both bottom-up and top-down configurations, enhancing the transfer of low-level features within the algorithm. This structure is particularly effective in boosting localization accuracy at lower layers, thus enabling more precise object localization [38]. The head of the YOLOv5 architecture is tailored for multi-level predictions, generating feature map outputs at three distinct levels. This multi-tiered approach allows the detection of objects of varying sizes, from small to large, with rapid inference speeds and high accuracy. Such capabilities render YOLOv5 exceptionally suitable for real-time object detection systems [39,40].

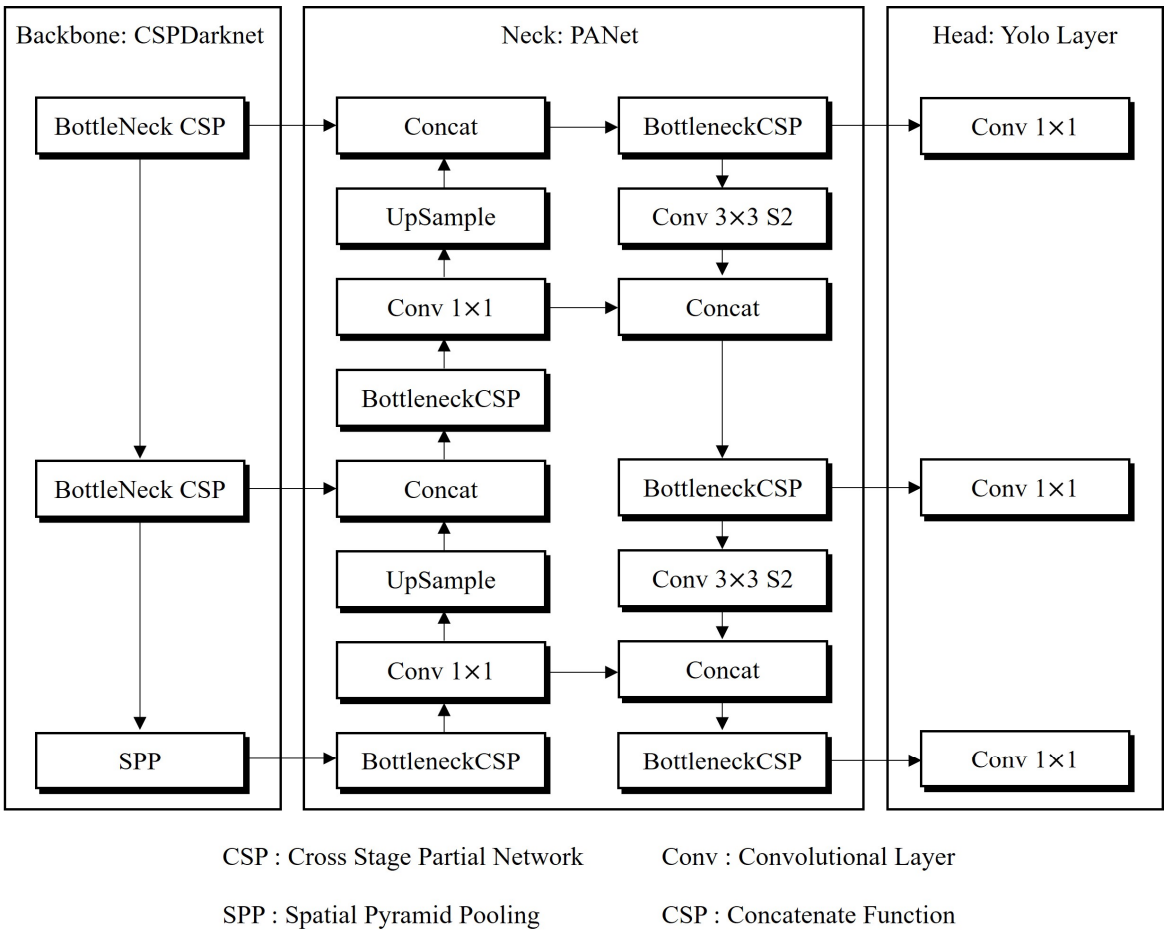

Figure 3. The general architecture of YOLOv5 [41].

3. Methodology

The development of a deep learning-based real-time droplet detector using the YOLO algorithm involves the process of building a development environment and training the algorithm in that environment. The hardware configuration utilized included a computer equipped with a 12th Gen Intel® Core™ i7-12700 processor, an NVIDIA RTX4080 graphics card, and 32 GB of RAM. The software environment, operating on the Windows system, extensively used various programs, including Anaconda 3-2021.11, Python 3.7, PyCharm-Professional-2021, and PyQt 5.15.4. Training of the YOLO algorithm consists of three steps, as shown in the sections below.

3.1. Step 1. Create an Image Dataset

This dataset was created by collecting images where the lens was contaminated by multiple droplets under various background conditions and labeling the collected images as 'droplet'. The dataset contains a bounding box containing information about the location and size of the droplet. To collect images contaminated by multiple droplets, a glass measuring 20 mm across and 20 mm long was attached to the lens of a mobile IP camera. The surface of the glass was coated with a 1 µm thick CYTOP as the hydrophobic coating. By increasing the contact angle between the droplet and the glass plate through the hydrophobic coating, the boundary, shape, and size of the droplet can be accurately determined. Next, a large number of droplets were sprayed onto the glass surface using a droplet spray device under various hue, saturation, and brightness conditions, and 4000 images of the same size of 640 × 640 pixels were collected. Figure 4 shows representative samples.

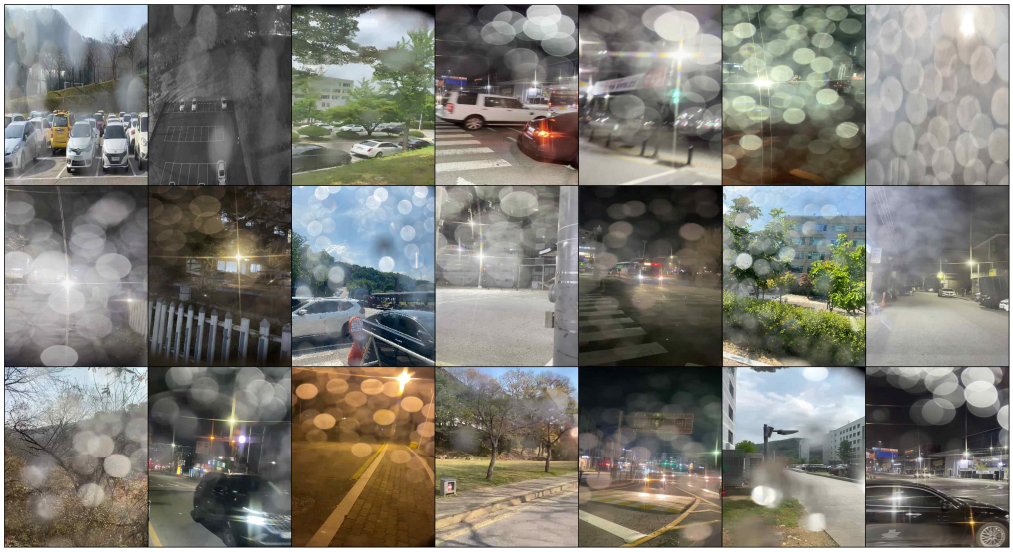

Figure 4. Images of the dataset for training and validation.

We used DarkLabel [42], an open-source graphical image labeling tool to label the collected images with 'droplet'. Image labeling provides the information a model needs to identify and classify objects within an image. The results of labeling are the coordinates, sizes, and bounding boxes. As shown in Figure 5, each droplet within the digital image was manually labeled to generate a label file containing the coordinate information of the object. This file documents the object's class, as well as the height and width of the bounding box for each object. In this scenario, as there is only one class, all droplets are labeled under the same class. The file format is textual (.txt).

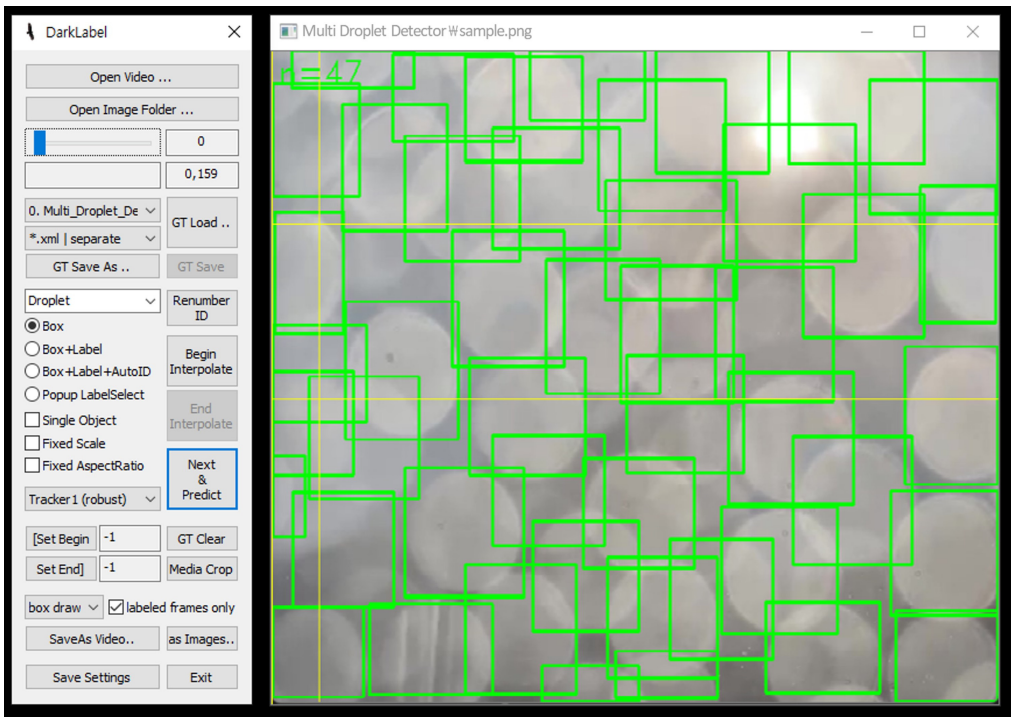

Figure 5. Image annotation tool DarkLabel [42].

3.2. Step 2. Training of Algorithms

The second step involves setting the training variables and learning droplets from the dataset using the YOLO algorithm. The training utilized a dataset of 4000 images,

partitioned into training and validation sets at a 7:3 ratio, amounting to 2800 images for training and 1200 for validation. In our research, YOLO (v5n, v5s, v5m, v5l, and v5x) was trained by tuning the batch size and number of epochs. We experimented with batch sizes (4, 8, and 16), observing that training time was inversely proportional to batch size. However, model performance, such as model accuracy and loss rates, were unaffected by variations in batch size. Based on these results, we trained the model with a batch size of 16. Additionally, the image resolution was set to 640 × 640 pixels for computational efficiency and precision accuracy [43]. The training of the detection model was performed with the YOLO (v5n, v5s, v5m, v5l, and v5x) algorithm, each using a different number of convolutional layers and filters [44]. YOLOv5n features the least number of convolutional layers and filters, optimizing the model size and computational demands, whereas YOLOv5x has the most, enhancing accuracy and complex feature learning capability. Table 1 shows the main training parameters used in training.

Table 1. The detailed training strategies applied to model training and information used in the training process.

Model	Training Parameters	Optimizer	Learning Rate	Momentum	Image Size	Batch Size	Epochs
YOLOv5n	1,900,000	SGD	0.001	0.937	640 × 640	16	150
YOLOv5s	7,200,000	SGD	0.001	0.937	640 × 640	16	150
YOLOv5m	21,200,000	SGD	0.001	0.937	640 × 640	16	150
YOLOv5l	46,500,000	SGD	0.001	0.937	640 × 640	16	150
YOLOv5x	86,700,000	SGD	0.001	0.937	640 × 640	16	150

In the training process, repetition can be divided into two stages: First, we apply the training dataset to the algorithm, and the algorithm automatically adjusts the weights according to the loss values. The loss value is calculated through the GIOU Loss function [45,46]. Subsequently, the validation dataset is applied, and the loss value is recalculated with updated weights, serving as a key indicator of model performance. The final model was constructed using PyTorch [47,48].

3.3. Step 3. Quantitatively Evaluation of the Detection Performance

In the third step, the performance of the model developed through algorithm training is evaluated using four performance metrics: precision, recall, F1 score, and average precision [49]. Firstly, precision and recall are calculated using Equations (1) and (2):

$$\text{Precision} = \frac{TP}{TP + FP} \tag{1}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2}$$

TP (True Positive) represents the number of objects accurately identified, while FP (False Positive) refers to the count of objects incorrectly identified, and FN (False Negative) signifies the number of objects that were not detected. The quantification of these objects is based on the IoU (Intersection over Union) metric, which involves calculating and thresholding. IoU evaluates the accuracy of detection by comparing the extent of overlap between the predicted bounding boxes and the actual ones, as illustrated in Figure 6 [50]. Following this, we compute the F1 score, a balanced measure of precision and recall, using the formula presented in Equation (3).

$$\text{F1} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{3}$$

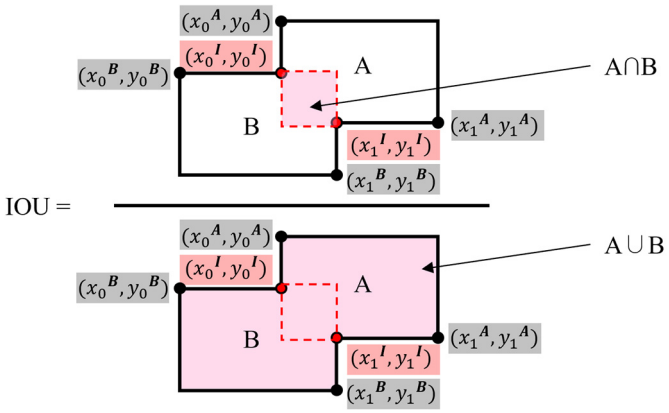

Figure 6. IoU calculation.

Lastly, AP (Average Precision) is computed through Equation (4), with mAP (Mean Average Precision) derived by averaging the AP across categories.

$$\mathrm{AP} = \sum_{i=1}^{N} precision(k) \times \Delta recall(k) \tag{4}$$

In Equation (4), 'n' denotes the total number of images in the dataset, *precision*(k) is the precision at the kth image, and Δ*recall*(k) is the recall difference between the $k-1$th and kth images.

4. Result

4.1. Training Loss Analysis of Architectures

Figure 7 illustrates the training and validation loss for the five architectures. Training and validation loss tended to be inversely proportional to the number of epochs, number of convolution layers, and number of filters. Training and validation loss observed decreases as epochs increase and for architecture with more convolution layers and filters. From these results, it was confirmed that the model trained with YOLOv5x, which is a complex architecture, can achieve lower training and validation loss than other trained models and has the lowest training loss of 0.0348 at epoch 150.

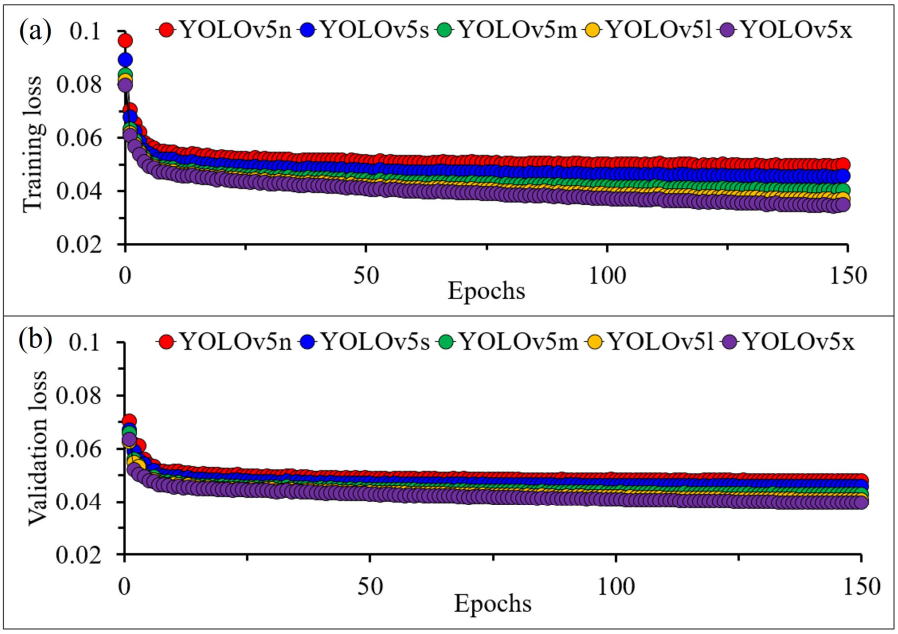

Figure 7. Loss of five different architectures: (**a**) Training; (**b**) Validation.

4.2. Compare the Detection Performance of Trained Models

Figure 8 and Table 2 detail the quantitative performance evaluations of trained models with the YOLO (v5n, v5s, v5m, v5l, and v5x) architectures. The results show that the precision and inference time of the trained models is proportional to the number of convolution layers and filters. Architectures with more convolution layers and filters have more computation capability, which improves precision by identifying more complex features but also increases inference time. As a result, the trained model with the YOLOv5n architecture demonstrates the lowest performance with 76.07% mAP@0.5, 34.87% mAP@0.5:0.95, 77.81% precision, 64.66% recall, and a 70.62% F1 score. In contrast, the trained model with the YOLOv5x architecture demonstrates the highest performance among others with 87.46% mAP@0.5, 51.90% mAP@0.5:0.95, 90.28% precision, 81.47% recall, and 85.64% F1 score. Also, the inference times of trained models with YOLO (v5n, v5s, v5m, v5l, and v5x) architecture were recorded as 3.3 ms, 4.8 ms, 8.9 ms, 14.7 ms, and 23 ms, respectively.

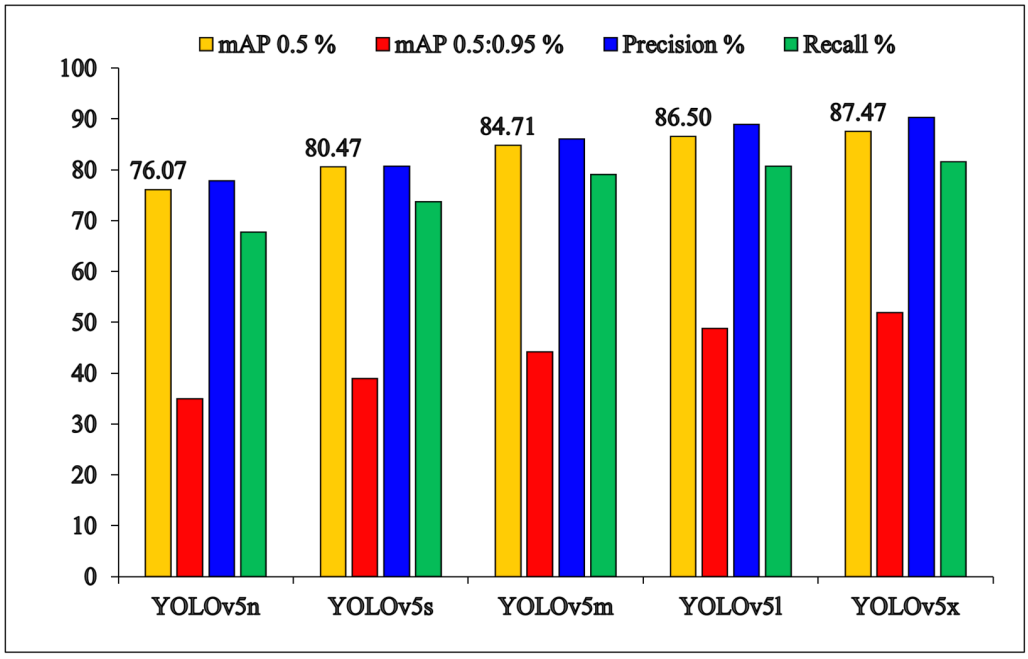

Figure 8. Quantitative results of the training process of five different architectures.

Table 2. Comparison of trained models with the YOLO (v5n, v5s, v5m, v5l, and v5x) architectures.

Model	mAP 0.5 (%)	mAP 0.5:0.95 (%)	Precision (%)	Recall	F1 Score	GFLOPS	Inference Time Millisecond (ms)
YOLOv5n	76.07	34.87	77.81	67.66	70.62	4.2	3.3
YOLOv5s	80.47	38.88	80.69	73.65	77.00	16.0	4.8
YOLOv5m	84.70	44.16	86.02	78.97	82.34	448.2	8.9
YOLOv5l	86.50	48.74	88.90	80.68	84.59	108.2	14.7
YOLOv5x	87.46	51.90	90.28	81.47	85.64	204.4	23.0

4.3. Detection Results

Figure 9 shows the result of using a model trained in real time to detect droplets generated on the surface of a camera lens. The experimental results show that a significant number of droplets can be successfully detected from backgrounds of varying hue, saturation, and brightness. This demonstrates the ability of the trained model with YOLOv5x architecture to effectively identify droplets even under complex and diverse environmental conditions. Furthermore, these results demonstrate the application of the real-time droplet detector in the diverse real world where environmental conditions are

variable and unpredictable. The trained model is open source at https://github.com/TransparentHeaterYKKIM/Droplet-detector.git (accessed on 29 January 2024).

Figure 9. Detection results of YOLOv5x.

4.4. Integrating Contamination Detection Models with Cleaning Systems

Finally, an effective contamination detection and cleaning system was implemented by integrating the proposed deep learning-based detector with a previously developed contamination cleaning system using a transparent heater [24]. The system consists of a camera, a transparent heater for droplet removal, a server for image analysis, and a monitor for real-time contamination detection and cleaning verification (Figure 10a). The transparent heater consists of optical glass, mesh copper electrodes (electrode width: 10 μm, gap: 350 μm), and insulating film (Cytop, thickness 1 μm). The manufactured transparent heater is attached to the outside of the camera lens and is directly exposed to external contamination. To simulate lens contamination, multiple droplets were sprayed. When droplets adhere to the transparent heater, the digital image input to the image sensor through the camera lens is distorted. These digital images are transmitted in real time from the camera to the analysis server, which uses a pre-trained YOLOv5 model to determine whether the image is contaminated (Figure 10(b1)). When contamination is detected, the analysis server transmits a command signal to drive the transparent heater to the Arduino switch. Direct current voltage is applied to the transparent heater through this switch, and the transparent heater generates heat by resistance heat. Through this process, several droplets evaporate and are removed from the lens surface (Figure 10(b2)). This experiment successfully demonstrated that when the image is distorted due to contamination on the lens surface, the contamination detection model can be used to identify the contamination, and the cleaning system can be used to effectively restore the image. This result is significant in that it established an efficient and automated lens contamination cleaning system by combining a lens cleaning system and a deep learning-based detector. On development hardware, the YOLO v5x model has an inference time of 23 ms, which corresponds to a processing power of 43 fps. This performance exceeds the traditional real-time threshold of 30 fps, which can provide smooth video in a real-time monitoring environment. However, as the performance of the model is related to the operating environment, many factors need to be considered comprehensively to design an efficient system for real-world applications. Lightweight deep-learning models (ex. YOLO v5n) optimized for less powerful hardware may be a suitable alternative for detecting contamination in environments where computing resources are limited.

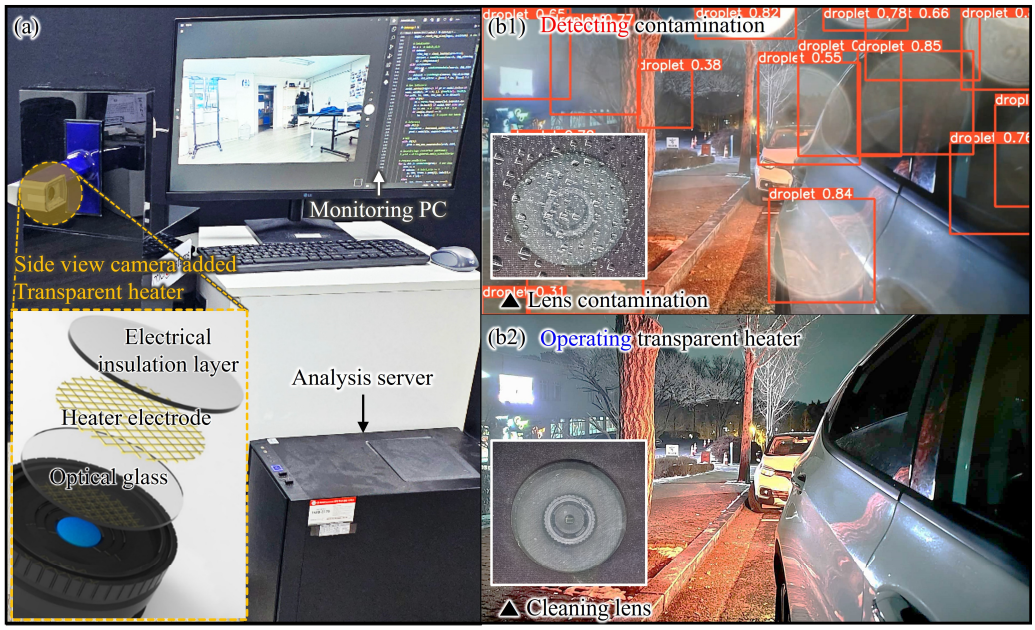

Figure 10. Demonstration of the proposed system integrating lens contamination detection and transparent heater cleaning: (**a**) experimental setup of the proposed system; (**b1**,**b2**) sequential snapshot of detecting lens contamination using the deep learning-based detection system and the removal of the contamination by the transparent heater-based cleaning system.

5. Discussion

Our proposed system, trained on a diverse range of environmental data, shows great promise in accurately recognizing and categorizing contamination under a broad spectrum of conditions. This stands in contrast to traditional contamination detection methods, which operate effectively only in limited environments. The application of deep learning technology enables more precise detection, even in variable and uncontrolled environments, thus expanding the scope of real-world applications. This research represents the first application of AI in the field of sensor cleaning technology, demonstrating practical viability through a series of demo experiments. Our future endeavors include enriching the existing single-class dataset with additional images to enhance the accuracy of the contamination detection model. Given that deep learning models perform optimally with larger datasets, we plan to collect data across a wider array of environments and implement data augmentation techniques such as cropping, brightening, and blurring. These methods will enable the model to adapt more effectively to complex conditions, thus enhancing its practicality and reliability for real-world applications. Initially focused on droplet contamination commonly found on external camera lenses, our future objective is to expand the detectable range of contamination. We aim to construct a multi-class dataset that includes not only droplets but also dust, mud, and insects. This expansion will enable the model to detect a wider variety of contamination types more efficiently.

6. Conclusions

We introduced an innovative lens contamination detection system designed to address the accuracy challenges in contamination detection for vehicle vision sensors. Our approach leverages various object detection architectures, including YOLO (versions v5n, v5s, v5m, v5l, and v5x), enabling the real-time detection of contaminants in digital images. To actualize this system, we compiled a dataset comprising 4000 images, each depicting droplet contamination under diverse conditions. To implement the proposed approach, we constructed a dataset consisting of 4000 images contaminated by droplets under various conditions. Model training, validation, and testing were performed on datasets generated using a computer equipped with a 12th Generation Intel® Core™ i7-12700 processor, an

NVIDIA RTX4080 graphics card, and 32 GB RAM. The results show that the model trained with the YOLOv5x architecture was the most successful and achieved significant results. The model achieved an average precision (mAP@0.5) of 87.46%, (mAP@0.5:0.95) of 51.90%, accuracy of 90.28%, recall of 81.47%, and F1 score of 85.64%, with an inference time of 23.0 ms. Moreover, we successfully integrated this deep learning-based detection system with a heater-based cleaning mechanism on a vehicle camera, showcasing its ability to detect and eliminate contaminants in real time.

We anticipate that the proposed system will find utility not only in vehicle vision sensors but also in a multitude of practical applications such as autonomous driving, surveillance cameras, and drone imaging systems. Furthermore, by adapting to complex and diverse environments, this system holds the potential for significant roles in industrial robots, smart agriculture, disaster response, and rescue operations, contributing to creating safer and more efficient environments through real-time pollution detection.

Author Contributions: Conceptualization, Y.K., W.K., S.C. and D.K.; methodology, Y.K., W.K. and J.Y.; software, Y.K., W.K. and J.Y.; validation, Y.K., W.K, S.C. and D.K.; formal analysis, Y.K., W.K. and J.Y.; investigation, Y.K., W.K., J.Y. and D.K.; resources, S.C.; data curation, W.K. and J.Y.; writing—original draft preparation, Y.K., W.K., J.Y. and D.K.; writing—review and editing, S.C. and D.K.; visualization, Y.K. and W.K.; supervision, S.C. and D.K.; project administration, S.C. and D.K.; funding acquisition, S.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by 2024 Research Fund of Myongji University.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: Author Daegeun Kim was employed by the company Microsystems, Inc. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Road Traffic Deaths, Global Health Observatory Data Repository by World Health Organization. Available online: https://www.who.int/data/gho/data/themes/topics/topic-details/GHO/road-traffic-mortality (accessed on 22 November 2023).
2. Bellis, E.; Page, J. *National Motor Vehicle Crash Causation Survey (NMVCCS) SAS Analytical Users Manual*; Calspan Corp.: Buffalo, NY, USA, 2008; pp. 154–196.
3. Rolison, J.J.; Regev, S.; Moutari, S.; Feeney, A. What Are the Factors That Contribute to Road Accidents? An Assessment of Law Enforcement Views, Ordinary Drivers' Opinions, and Road Accident Records. *Accid. Anal. Prev.* **2018**, *115*, 11–24. [CrossRef]
4. Dabral, S.; Kamath, S.; Appia, V.; Mody, M.; Zhang, B.; Batur, U. Trends in Camera Based Automotive Driver Assistance Systems (ADAS). In Proceedings of the 2014 IEEE 57th International Midwest Symposium on Circuits and Systems (MWSCAS), College Station, TX, USA, 3–6 August 2014; ISBN 9781479941322.
5. Yeong, D.J.; Velasco-Hernandez, G.; Barry, J.; Walsh, J. Sensor and Sensor Fusion Technology in Autonomous Vehicles: A Review. *Sensors* **2021**, *21*, 2140. [CrossRef] [PubMed]
6. Ziebinski, A.; Cupek, R.; Erdogan, H.; Waechter, S. A Survey of ADAS Technologies for the Future Perspective of Sensor Fusion. In Proceedings of the Computational Collective Intelligence: 8th International Conference, Halkidiki, Greece, 28–30 September 2016; Springer International Publishing: Cham, Switzerland, 2016.
7. Zang, S.; Ding, M.; Smith, D.; Tyler, P.; Rakotoarivelo, T.; Kaafar, M.A. The Impact of Adverse Weather Conditions on Autonomous Vehicles: How Rain, Snow, Fog, and Hail Affect the Performance of a Self-Driving Car. *IEEE Veh. Technol. Mag.* **2019**, *14*, 103–111. [CrossRef]
8. Mannetje, T.D.J.C.M.; Murade, C.U.; Van Den Ende, D.; Mugele, F. Electrically assisted drop sliding on inclined planes. *Appl. Phys. Lett.* **2011**, *98*, 118–121. [CrossRef]
9. Hong, J.; Lee, S.J.; Koo, B.C.; Suh, Y.K.; Kang, K.H. Size-selective sliding of sessile drops on a slightly inclined plane using low frequency AC electrowetting. *Langmuir* **2012**, *28*, 6307–6312. [CrossRef]
10. Tan, M.K.; Friend, J.R.; Yeo, L.Y. Microparticle collection and concentration via a miniature surface acoustic wave device. *Lab Chip* **2007**, *7*, 618–625. [CrossRef] [PubMed]
11. Alagoz, S.; Apak, Y. Removal of spoiling materials from solar panel surfaces by applying surface acoustic waves. *J. Clean. Prod.* **2020**, *253*, 119992. [CrossRef]

12. Lee, S.; Kim, D.I.; Kim, Y.Y.; Park, S.-E.; Choi, G.; Kim, Y.; Kim, H.J. Droplet evaporation characteristics on transparent heaters with different wettabilities. *RSC Adv.* **2017**, *7*, 45274–45279. [CrossRef]
13. Kim, H.J.; Kim, J.; Kim, Y. Afluoropolymer-coated nanometer-thick Cu Mesh film for a robust and hydrophobic transparent heater. *ACS Appl. Nano Mater.* **2020**, *3*, 8672–8678. [CrossRef]
14. Yong Lee, K.; Hong, J.; Chung, S.K. Smart self-cleaning lens cover for miniature cameras of automobiles. *Sens. Actuators B Chem.* **2017**, *239*, 754–758. [CrossRef]
15. Song, H.; Jang, D.; Lee, J.; Lee, K.Y.; Chung, S.K. SAW-driven self-cleaning drop free glass for automotive sensors. *J. Micromech. Microeng.* **2021**, *31*, 12. [CrossRef]
16. Park, J.; Lee, S.; Kim, D.I.; Kim, Y.Y.; Kim, S.; Kim, H.J.; Kim, Y. Evaporation-rate control of water droplets on flexible transparent heater for sensor application. *Sensors* **2019**, *19*, 4918. [CrossRef]
17. Robins, M.N.; Bean, H.N. Camera Lens Contamination Detection and Indication System and Method. U.S. Patent US6940554B2, 6 September 2005.
18. Zhang, Y. Self-Detection of Optical Contamination or Occlusion in Vehicle Vision Systems. *Opt. Eng.* **2008**, *47*, 067006. [CrossRef]
19. Lai, C.C.; Li, C.H.G. Video-Based Windshield Rain Detection and Wiper Control Using Holistic-View Deep Learning. In Proceedings of the IEEE International Conference on Automation Science and Engineering (CASE), Vancouver, BC, Canada, 22–26 August 2019; Section III. pp. 1060–1065. [CrossRef]
20. Tao, H.; Duan, Q.; Lu, M.; Hu, Z. Learning Discriminative Feature Representation with Pixel-level Supervision for Forest Smoke Recognition. *Pattern Recognit.* **2023**, *143*, 109761. [CrossRef]
21. Cao, Y.; Li, C.; Peng, Y.; Ru, H. MCS-YOLO: A Multiscale Object Detection Method for Autonomous Driving Road Environment Recognition. *IEEE Access* **2023**, *11*, 22342–22354. [CrossRef]
22. Bengio, Y.; LeCun, Y. Scaling Learning Algorithms towards AI. In *Large-Scale Kernel Machines*; MIT Press: Cambridge, MA, USA, 2007; Volume 34.
23. Hassan, M.; Wang, Y.; Wang, D.; Li, D.; Liang, Y.; Zhou, Y.; Xu, D. Deep Learning Analysis and Age Prediction from Shoeprints. *Forensic Sci. Int.* **2021**, *327*, 110987. [CrossRef] [PubMed]
24. Kim, Y.; Lee, J.; Chung, S.K. Heat-Driven Self-Cleaning Glass Based on Fast Thermal Response for Automotive Sensors. *Phys. Scr.* **2023**, *98*, 085932. [CrossRef]
25. Kamilaris, A.; Prenafeta-Boldú, F.X. Deep learning in agriculture: A survey. *Comput. Electron. Agric.* **2018**, *147*, 70–90. [CrossRef]
26. Agatonovic-Kustrin, S.; Beresford, R. Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research. *J. Pharm. Biomed. Anal.* **2000**, *22*, 717–727. [CrossRef]
27. Huang, Y.; Sun, S.; Duan, X.; Chen, Z. A study on Deep Neural Networks framework. In Proceedings of the 2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), Xi'an, China, 3–5 October 2016. [CrossRef]
28. Samek, W.; Montavon, G.; Lapuschkin, S.; Anders, C.J.; Müller, K.-R. Explaining Deep Neural Networks and Beyond: A review of Methods and Applications. *Proc. IEEE* **2021**, *109*, 247–278. [CrossRef]
29. Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-Based Learning Applied to Document Recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
30. Alzubaidi, L.; Zhang, J.; Humaidi, A.J.; Al-Dujaili, A.; Duan, Y.; Al-Shamma, O.; Santamaría, J.; Fadhel, M.A.; Al-Amidie, M.; Farhan, L. Review of Deep Learning: Concepts, CNN Architectures, Challenges, Applications, Future Directions. *J. Big Data* **2021**, *8*, 53. [CrossRef] [PubMed]
31. Kattenborn, T.; Leitloff, J.; Schiefer, F.; Hinz, S. Review on Convolutional Neural Networks (CNN) in Vegetation Remote Sensing. *ISPRS J. Photogramm. Remote Sens.* **2021**, *173*, 24–49. [CrossRef]
32. Sarvamangala, D.R.; Kulkarni, R.V. Convolutional neural networks in medical image understanding: A survey. *Evol. Intell.* **2022**, *15*, 1–22. [CrossRef] [PubMed]
33. Guo, Z.; Wang, C.; Yang, G.; Huang, Z.; Li, G. MSFT-YOLO: Improved YOLOv5 Based on Transformer for Detecting Defects of Steel Surface. *Sensors* **2022**, *22*, 3467. [CrossRef] [PubMed]
34. Yan, B.; Fan, P.; Lei, X.; Liu, Z.; Yang, F. A Real-Time Apple Targets Detection Method for Picking Robot Based on Improved YOLOv5. *Remote Sens.* **2021**, *13*, 1619. [CrossRef]
35. Xue, Z.; Lin, H.; Wang, F. A Small Target Forest Fire Detection Model Based on YOLOv5 Improvement. *Forests* **2022**, *13*, 1332. [CrossRef]
36. Wang, Z.; Wu, L.; Li, T.; Shi, P. A Smoke Detection Model Based on Improved YOLOv5. *Mathematics* **2022**, *10*, 1190. [CrossRef]
37. Wang, K.; Liew, J.H.; Zou, Y.; Zhou, D.; Feng, J. PANet: Few-Shot Image Semantic Segmentation with Prototype Alignment. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019.
38. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
39. Xu, R.; Lin, H.; Lu, K.; Cao, L.; Liu, Y. A Forest Fire Detection System Based on Ensemble Learning. *Forests* **2021**, *12*, 217. [CrossRef]
40. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path Aggregation Network for Instance Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.

41. Nepal, U.; Eslamiat, H. Comparing YOLOv3, YOLOv4 and YOLOv5 for Autonomous Landing Spot Detection in Faulty UAVs. *Sensors* **2022**, *22*, 464. [CrossRef]
42. Sama, A.K.; Sharma, A. Simulated Uav Dataset for Object Detection. *ITM Web Conf.* **2023**, *54*, 02006. [CrossRef]
43. Bayer, H.; Aziz, A. Object Detection of Fire Safety Equipment in Images and Videos Using Yolov5 Neural Network. In Proceedings of the 33rd Forum Bauinformatik, München, Germany, 7–9 September 2022. [CrossRef]
44. Jocher, G.; Stoken, A.; Borovec, J.; NanoCode012, C.; Changyu, L.; Laughing, H. ultralytics/yolov5: v3.0. 2020. Available online: https://github.com/ultralytics/yolov5 (accessed on 20 December 2020).
45. Liu, P.; Zhang, G.; Wang, B.; Xu, H.; Liang, X.; Jiang, Y.; Li, Z. Loss Function Discovery for Object Detection via Convergence-Simulation Driven Search. *arXiv* **2021**, arXiv:2102.04700. [CrossRef]
46. Liu, W.; Wang, Z.; Zhou, B.; Yang, S.; Gong, Z. Real-time Signal Light Detection based on Yolov5 for Railway. *IOP Conf. Ser. Earth Environ. Sci.* **2021**, *769*, 042069. [CrossRef]
47. Stevens, E.; Antiga, L.; Viehmann, T. *Deep Learning with PyTorch*; Manning Publications: Shelter Island, NY, USA, 2020.
48. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019.
49. Jia, W.; Xu, S.; Liang, Z.; Zhao, Y.; Min, H.; Li, S.; Yu, Y. Real-time Automatic Helmet Detection of Motorcyclists in Urban Traffic Using Improved YOLOv5 Detector. *IET Image Process.* **2021**, *15*, 3623–3637. [CrossRef]
50. Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.