

Article

Traffic Management: Multi-Scale Vehicle Detection in Varying Weather Conditions Using YOLOv4 and Spatial Pyramid Pooling Network

Mamoona Humayun ^{1,*}, Farzeen Ashfaq ², Noor Zaman Jhanjhi ² and Marwah Khalid Alsadun ³

¹ Department of Information Systems, College of Computer and Information Sciences, Jouf University, Sakakah 72311, Saudi Arabia

² School of Computer Science (SCS), Taylor's University, Subang Jaya 47500, Malaysia

³ Department of Computer Science, College of Computer and Information Sciences, Jouf University, Sakakah 72311, Saudi Arabia

* Correspondence: mahumayun@ju.edu.sa

Abstract: Detecting and counting on road vehicles is a key task in intelligent transport management and surveillance systems. The applicability lies both in urban and highway traffic monitoring and control, particularly in difficult weather and traffic conditions. In the past, the task has been performed through data acquired from sensors and conventional image processing toolbox. However, with the advent of emerging deep learning based smart computer vision systems the task has become computationally efficient and reliable. The data acquired from road mounted surveillance cameras can be used to train models which can detect and track on road vehicles for smart traffic analysis and handling problems such as traffic congestion particularly in harsh weather conditions where there are poor visibility issues because of low illumination and blurring. Different vehicle detection algorithms focusing the same issue deal only with on or two specific conditions. In this research, we address detecting vehicles in a scene in multiple weather scenarios including haze, dust and sandstorms, snowy and rainy weather both in day and nighttime. The proposed architecture uses CSPDarknet53 as baseline architecture modified with spatial pyramid pooling (SPP-NET) layer and reduced Batch Normalization layers. We also augment the DAWN Dataset with different techniques including Hue, Saturation, Exposure, Brightness, Darkness, Blur and Noise. This not only increases the size of the dataset but also make the detection more challenging. The model obtained mean average precision of 81% during training and detected smallest vehicle present in the image

Keywords: traffic surveillance; intelligent traffic monitoring; urban and highway traffic analysis; artificial intelligence; deep learning; vehicle detection



Citation: Humayun, M.; Ashfaq, F.; Jhanjhi, N.Z.; Alsadun, M.K. Traffic Management: Multi-Scale Vehicle Detection in Varying Weather Conditions Using YOLOv4 and Spatial Pyramid Pooling Network. *Electronics* **2022**, *11*, 2748. <https://doi.org/10.3390/electronics11172748>

Academic Editors: Lei Chen, Lingtao Wu, Yajie Zou and Yue Zhang

Received: 3 August 2022

Accepted: 25 August 2022

Published: 1 September 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Over the last decade, deep neural networks have greatly influenced our way of working and processing information. Particularly, their ability to perceive, understand and analyze the visual information accurately and quickly has made them an important part of computer vision applications such as image classification [1–7], semantic segmentation [8–13], object detection [14–19] and object segmentation [20–26].

The task of object detection aims to detect instances of different objects for example humans [27–30], animals [31–33], trees [34–37], vehicles [38–41] or other items present in digital media such as images and video frames. To automate this task machine and deep learning involves developing computational models which are trained by learning class-specific features using huge amount of annotated data and then localizing and generating bounding boxes around each instance and label them. This makes object detection a super task of many other smaller tasks such as image captioning [42–45], object tracking [46–49], instance segmentation [50–52] and instance counting [53–56].

2. Approaches to Object Detection

Object detection methods can be mainly classified into classical methods which relies on manual feature extraction algorithms with a machine learning classifier and the recent more robust methods which depends on deep learning models for automatic feature extraction.

2.1. Object Detection Using Machine Learning

Classical methods of detecting objects within a media involves feature extraction algorithms such as HOG [57–63], HAAR [64–74] and SIFT [75–81] to manually craft features. Secondly, using a classifier such as support vector machine [8,82–86] and K Nearest Neighbor [84,87,88] to classify multiple objects within an image. The general processing pipeline is shown in Figure 1.

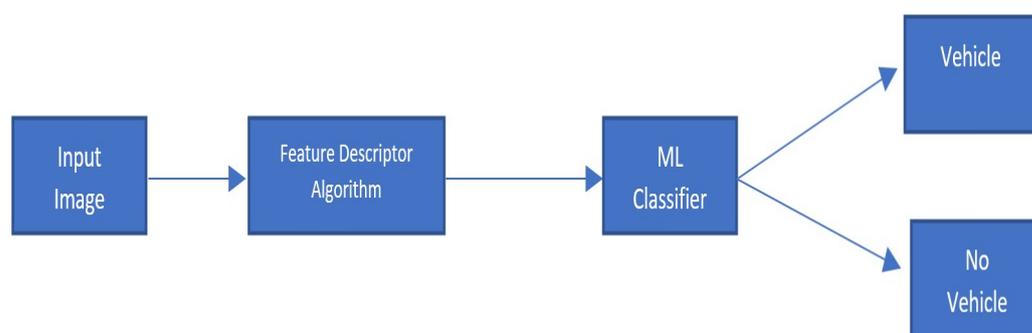


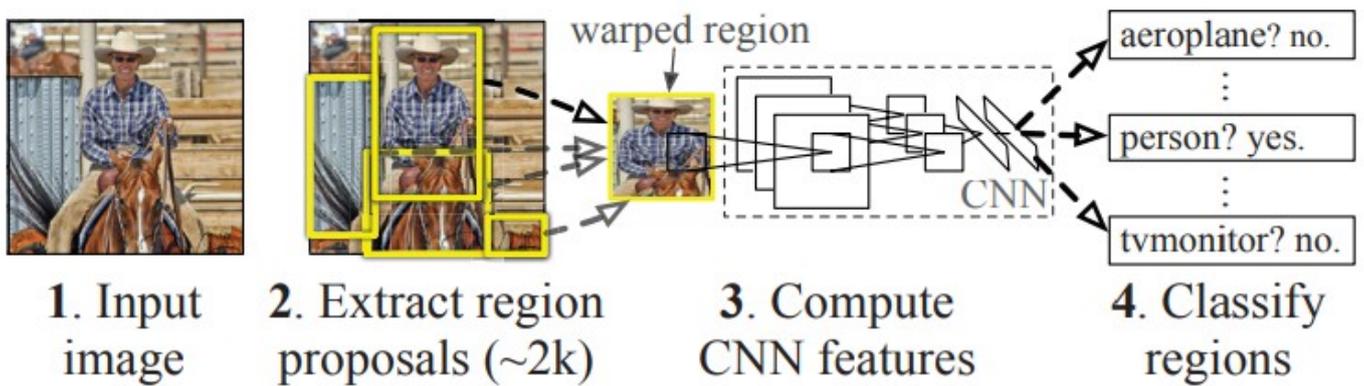
Figure 1. General processing pipeline of object detection with machine learning classifier.

2.2. Object Detection Using Deep Learning

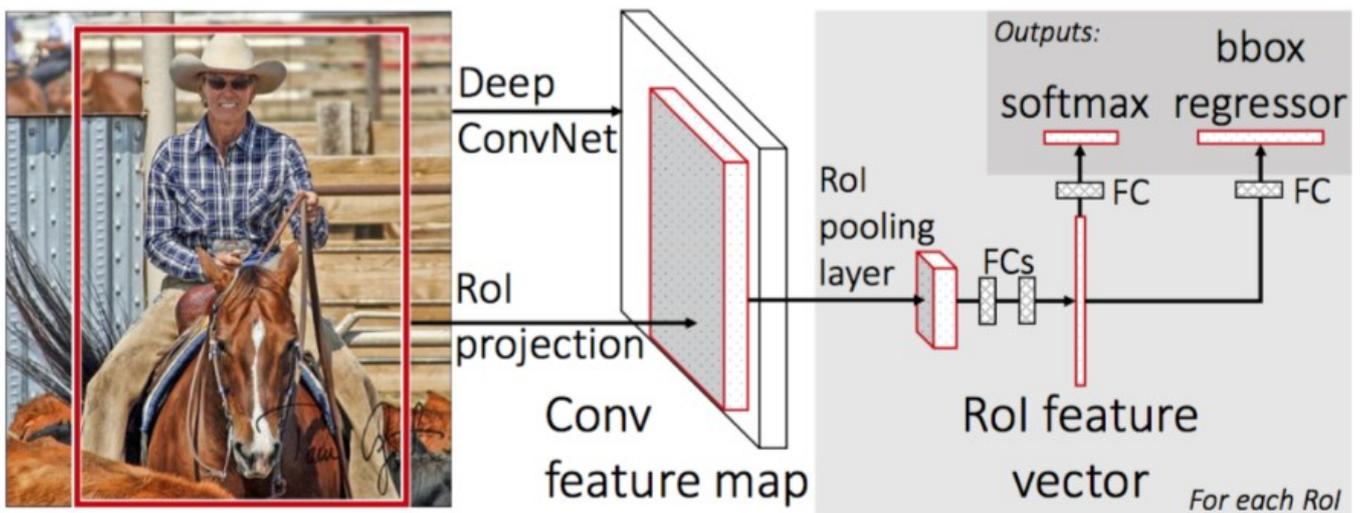
Deep Learning based algorithms provide a more robust and accurate solution to the problem of object detection. There is no need for a separate feature description part, rather the model is trained using images with bounding boxes and class labels. Thus the model learns automatically the visual attributes present in the underlying image. In the past few years, the task of object detection using deep learning is performed by a number of algorithms including RCNN, Fast RCNN, Faster RCNN, YOLO, YOLOv2, YOLOv3, YOLOv4 and RetinaNet.

2.2.1. RCNN Family

The algorithms belonging to this family are called “Classification Based Algorithms”. The first variant based on AlexNet as CNN backbone, known as Region based Convolution Neural Network or R-CNN was proposed by [89] in 2015. The network works by first identifying the regions with the probability of having objects. The rectangular boundaries are drawn using a selective search algorithm containing many ROI (Region of Interest). All these candidate bounding boxes are passed into the layers of convolution neural network. The CNN extract individual features from each bounding box. The output vector from the CNN representing the feature map is passed on to the classifier which then classifies each one of them into object and not object. The Fast RCNN [90] is inspired by Spatial Pyramid Pooling Network [91], The limitation of multi staged pipeline architecture which was the reason for slow training was overcome by the same author by applying CNN over whole image only once. The architecture of RCNN and Fast RCNN from the papers are shown in Figure 2a,b.



(a) RCNN



(b) Fast RCNN

Figure 2. Architecture of RCNN and fast RCNN.

The more refined and improved algorithms proposed under this family includes [92–95].

2.2.2. YOLO Family

YOLO is an abbreviation of “You Only Look Once” which is a regression based algorithm. It is a class of Convolution Neural Network which is meant for fast and accurate object detection in a single go. Soon after it was released in 2015 by [96], it gained world wide attention and many papers have been published on the topic from 2015 till now including [93,97–101]. The basic architecture of YOLO contains DarkNet as a backbone. DarkNet is an open source framework to implement neural networks written in C language and CUDA. It supports CPU and GPU computing, is quick, and is simple to install. The YOLO comprises of CNN layers, where vanilla YOLO by [96] contains 24 conv layers and 2 fully connected layers, whereas Fast Yolo [102] contains 9 convolutional layers.

The algorithm works by dividing an input image into grids of size $S \times S$. Each grid is responsible for predicting a number of bounding boxes for an object. Each bounding box is represented by five parameters as represented in Equation (1). Out of many bounding boxes the one with highest confidence score is selected. This is conducted by applying Non Maximum suppression (NMS) over all bounding boxes.

$$Bounding_Box = \{P_{confidence}, x, y, w, h\} \tag{1}$$

Here, x and y represents the centre coordinates of the bounding box, w is the Width of the Bounding Box, h is the Height of the Bounding Box and, $P_confidence$ represents the Probability in terms of Score from 0 to 1 that the Bounding Box contains an object. In addition, all YOLO models share the same architecture, which is broken down into the following components.

1. **BackBone:** It consists of a convolutional neural network that gathers and generates visual features of various sizes and shapes. The feature extractors employed are classification models such as ResNet, VGG, and EfficientNet.
2. **Neck:** A group of layers that combine and mix traits before sending them to the prediction layer.
3. **Head:** Combines the predictions from the bounding box with features from the neck. The features and bounding box coordinates are subjected to classification and regression in order to complete the detection procedure. normally produces 4 values, including *width*, *height*, and x, y coordinates.

The other versions of YOLO include;

- **YOLOv2:** [96] overcomes the limitations in the previous version by replacing the backbone architecture with DarkNet19 which results in faster detection with mAP of 78.6% on 544×544 resolution images from PASCAL VOC 2007 dataset. Moreover, out of the some improvements in this version compared to the predecessor, the first one is the addition of batch normalization layer for enhanced stability in the network. Secondly, the replacement of grid boxes with anchor box which allows the prediction of multiple objects. The ratio of overlap over union *IoU* between the predetermined anchor box and the anticipated bounding box is calculated. The *IoU* value is a threshold used to determine whether or not the likelihood of the identified object is high enough to make a forecast.

$$IOU = \frac{area_of_overlap}{area_of_union} \quad (2)$$

- **YOLOv3:** The same authors published their updated algorithm [103] in 2018. The Darknet-53 architecture served as its foundation. Newer version introduced the idea of Feature Pyramid Network (FPN) for multiple feature extraction for an image.
- **YOLOv4:** The bag of freebies and bag of specials concepts were introduced in YOLOv4 [104] as methods for improving model performance without raising the cost of inference. However, the author of YOLOv4 did experiment on three different backbone architectures including, CSPDarknet533, CSPResNext50 and EfficientNet-B3. The bag of specials allows the selection of additional modules in the neck such as Special Attention Module (SAM), Special Pyramid Pooling (SPP) and Path Aggregation Network (PAN).

2.2.3. RetinaNet

Introduced by [105], this is one of the best one-stage object detection models and has demonstrated success with dense and small-scale objects.

3. Literature Review

The task of vehicle detection and tracking systems is addressed by many researchers. The research can be classified based on three different criteria as illustrated in Figure 3.

1. Classification based on type of input:
 - (a) surveillance cameras acquired images and videos;
 - (b) UAV acquired images and videos;
 - (c) sensors acquired data (e.g., Lidar and Radar).
2. Classification based on object detection techniques:
 - (a) detection using Image processing with machine learning;
 - (b) deep learning-based systems.

- i. Two Stage Detector;
 - ii. One Stage Detector;
 - iii. Hybrid Models.
3. Classification based on purpose
 - (a) Autonomous Vehicles and Self driving vehicles
 - (b) Traffic Surveillance

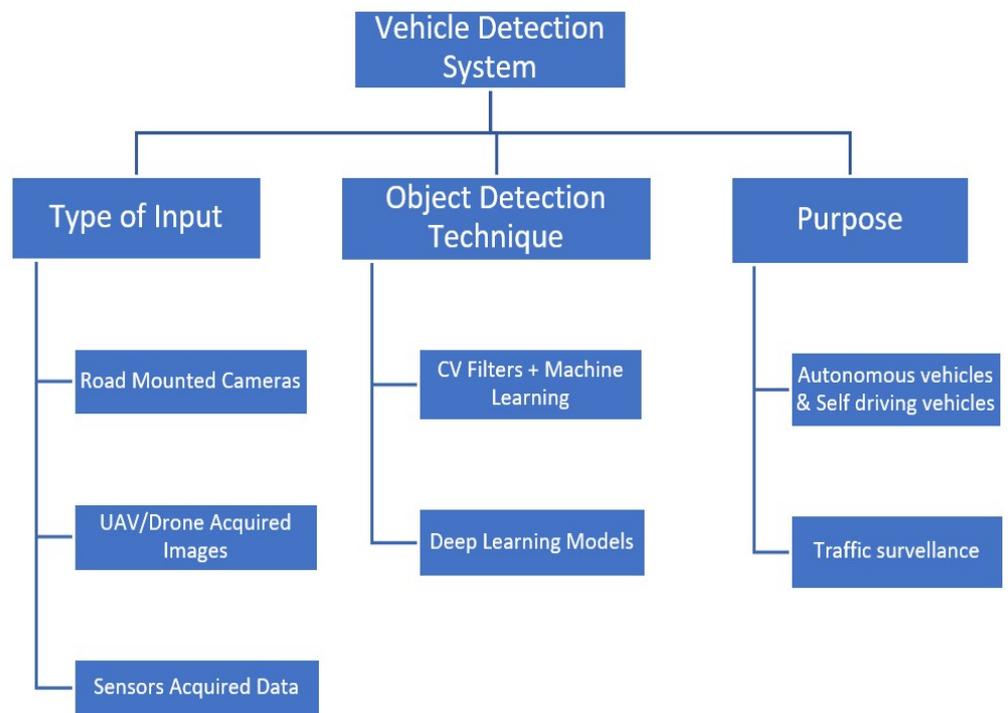


Figure 3. Classification of vehicle detection systems.

The process of recognising moving cars on the road can be conducted by using motion based or appearance based methods and is used for vehicle monitoring [106], counts, the average speed of each vehicle, movement analysis, and vehicle classification purposes. One of the potential application is self driving cars and autonomous vehicles [107]. Because of our page limit we restrict our focus only to research relating to images acquired from road surveillance or UAV mounted cameras. In addition, we will more focus on deep learning-based techniques conducted for the purpose of traffic surveillance monitoring and control.

We start our discussion with simple machine learning based vehicle detection systems. Ref. [108] examined three different classifiers for vehicle recognition including KNN, Neural Network, and Decision Tree. After extracting features using Canny Edge Detector.

Ref. [109] presented a cascade of boosted classifiers based on the characteristics of the vehicle images for detecting vehicles in on-road scene images. In their work, the classifier for vehicle detection was constructed utilising Haar-like features and an AdaBoost technique. The closest neighbourhood of the Euclidean distance was then used for the final classification.

Ref. [110] proposed a new vehicle identification model based on YOLOv2 with enhanced speed, accuracy, and predictive ability. To make the predictions over vehicles of varying sizes in the same image the loss function is normalized. In addition, to reduce the model's complexity the repeated convolutional layers in the final part of the model were removed as they normally deal with multi class in same image, whereas, the vehicle detection task only involves single class.

The idea of hybridizing manual feature extraction with deep learning framework is presented in a work by [111]. In this work, foreground feature detector is used along with morphological segmentation operations as pre-processing step. Kalman filter is used to hand craft position vector. In the last stage R-CNN is used to label the vehicles in the images. In another research conducted by [112] a three layer architecture is used to detect vehicles in video frames. The first layer draws feature maps from the input video frames. After that, a Region Proposed Network (RPN) is placed which slides over the feature maps and makes bounding boxes containing vehicles. At the final stage a R-CNN based detection network is used to assign class label to each ROI (Region of Interest) [113] supplies input images to an auto encoder layer before passing it to the deep learning framework. The AE neural network rather works as a pre-processing step and extracts more fine-grained robust features. In addition, they add different noise and vary illumination and contrast levels to make it suitable for training under varying weather conditions. After AE, fast RCNN and YOLO v3 was used for training and results comparison. Some other works where filters are applied as preprocessing the images before they are passed to deep learning model include dehazing, masking and thresholding [114], (automatic white balance combined with Laplacian pyramids) AWBLP [115,116]. Motivated by the idea of saving processing through transfer learning [117] applied frozen weights of SqueezeNet [118], ResNet50 [93] and EfficientNet [14] to train on DAWN dataset to detect vehicles under six different weather conditions. Ref. [119] uses YOLOv3 pretrained model on custom acquired images taken at nighttime. To enhance the visibility of the images, Multi Scale Retinex algorithm was applied before the convolution operation. This increases the contrast and removes the uneven brightness values in the images. The method obtained 93.66% average precision value which is higher than the single shot detector and Faster RCNN on the same image dataset. Ref. [120] presented a method to detect on road vehicle by modifying the faster RCNN architecture with multiple Region Proposed Networks (RPN). The reason for this is to capture the features of varying sizes vehicles from smaller to very larger, by setting each RPN with different aspect ratio and scale. The average precision of the proposed method with four RPNs is 89.48% on DAWN and 98.20% on CDNet 2014.

None of the above studies attempt to fully solve the issue of detecting different scales of vehicles appearing in a single image under a variety of weather conditions. However, to some extent the task of multi-scale vehicle detection was addressed by [121] using YOLOv3 as backbone architecture. The work suggest an encoding and decoding-based feature pyramid module. Through a straightforward U-shaped structure, this module may produce a high-order multi-scale feature MAP.

Keeping in mind all the above discussed studies, we propose a smart vehicle detection system using YOLOv4 to detect vehicles of varying sizes in images under varying weather conditions. The main contributions of our work in the domain of smart and intelligent vehicle detection are as follows:

1. We address the problem of vehicle detection in varying and harsh weather condition where the problem of low visibility is addressed by using a image restoration module.
2. We use YOLOv4 as backbone architecture for the detection of vehicles in six difficult weather conditions including haze, sandstorm, snowstorm, fog, dust, rain both in day and night.
3. We address the problem of detection far away and small size vehicles through the addition of double Spatial Pyramid Pooling Network after the last convolution layer [122].
4. For capturing vehicle's fine grained information we add an extra attention module before the convolution layer in the backbone architecture.

4. Materials and Methods

In this section, we discuss in detail the main key processes involved in our research. Figure 4 shows the overall flow of our proposed methodology.

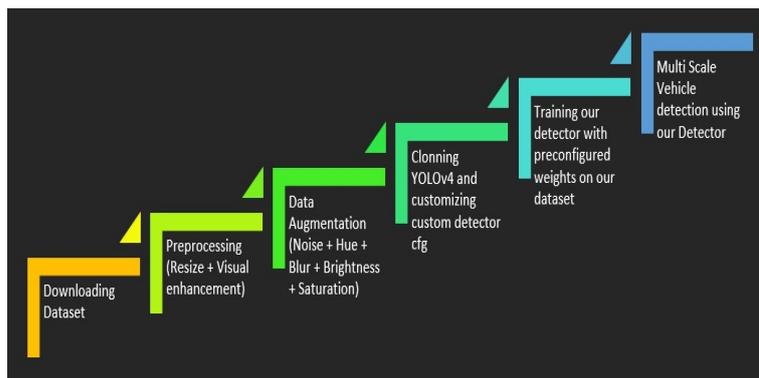


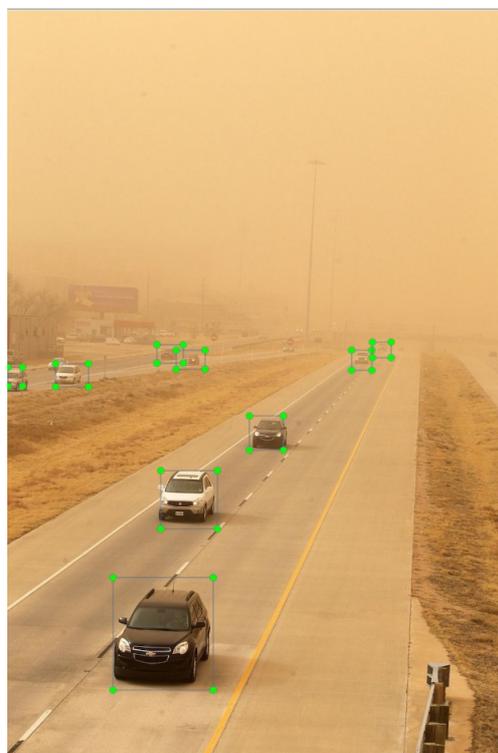
Figure 4. Flowchart of our proposed methodology.

4.1. Dataset

For the research, we chose to train our model with DAWN dataset which contains a collection of 1000 photos from actual traffic scenarios including highways and urban road traffic in varying weather conditions.

4.2. Data Annotation

For the task of object detection, the labeling of images was conducted in YOLO_Darknet format using a free and open source tool ‘LabelImg’ [123]. The labeling was conducted using polygon bounding box. After labeling a text file is placed in the same folder where images are located. The text file contains the $\{class_label, x, y, w, h\}$ for each image. Here, x and y represents the coordinates of the bounding box (anchor point) and w and h represents the width and height of the bounding box. In our research, we are only detecting cars, so class label is ‘car’. A set of image with assigned bounding boxes and associated text file is shown in Figure 5a,b.



(a)

Figure 5. Cont.

```

car 0.315657 0.836364 0.198990 0.150842
car 0.367677 0.656566 0.113131 0.078114
car 0.521717 0.567003 0.067677 0.045791
car 0.020707 0.491582 0.029293 0.025589
car 0.133838 0.489562 0.065657 0.030976
car 0.370202 0.468687 0.057576 0.025589
car 0.711111 0.469360 0.040404 0.026936
car 0.751010 0.455556 0.037374 0.020875
car 0.329293 0.460943 0.052525 0.024916

```

(b)

Figure 5. An example image annotated in LabelImg software. (a) Shows sample image with bounding boxes on vehicle instances; (b) Shows text file with $\{classlabel, x, y, w, h\}$ for each vehicle instance.

4.3. Data Augmentation

One of the challenging issue in vehicle detection research is the scarcity of available road side data, particularly if we are interested in detecting them in harsh and severe weather situations. Although, the DAWN dataset contains 1000 images from actual traffic scenarios under four weather categories, i.e., fog, sand, rain and snow. However, as the neural network's model receives increasingly more input, it's performance can continue to advance. One strategy to overcome this challenge is *data augmentation*, that allows researchers to greatly broaden the variety of environmental conditions on the road. The dataset was augmented with following techniques.

4.3.1. Hue

This augmentation techniques changes the colour levels in an image randomly so as the model will take into account different colour schemes for objects and scenes in input images. For our dataset images the hue level was chosen between -8° to $+82^\circ$.

4.3.2. Saturation

Similar to hue augmentation, saturation augmentation modifies how vivid and vibrant the image looks. we augmented the dataset with 50% saturation.

4.3.3. Brightness

To help your model be more resistant to variations in lighting and camera settings, we added variability to the image brightness up to 75%.

4.3.4. Exposure

While generally speaking, the brightness and exposure options both act the same. However, exposure favours highlight tones more than brightness, which has no bias and influences all tones equally. We add exposure in the range -30 to $+30$.

4.3.5. Blur

To make our model more camera focus-resistant, we added random Gaussian blur upto 10 px. With this the image appears to be completely out of focus due.

4.3.6. Noise

All these augmentation techniques were added to extend the number of training examples set under conditions so as to give an impression of difficult and varying weather conditions such as noise gives the effect of heavy rain and blur gives an impression of haze or fog in the environment. All these augmentation techniques along with original image are shown in Figure 6a–h.



(a) Original



(b) Hue

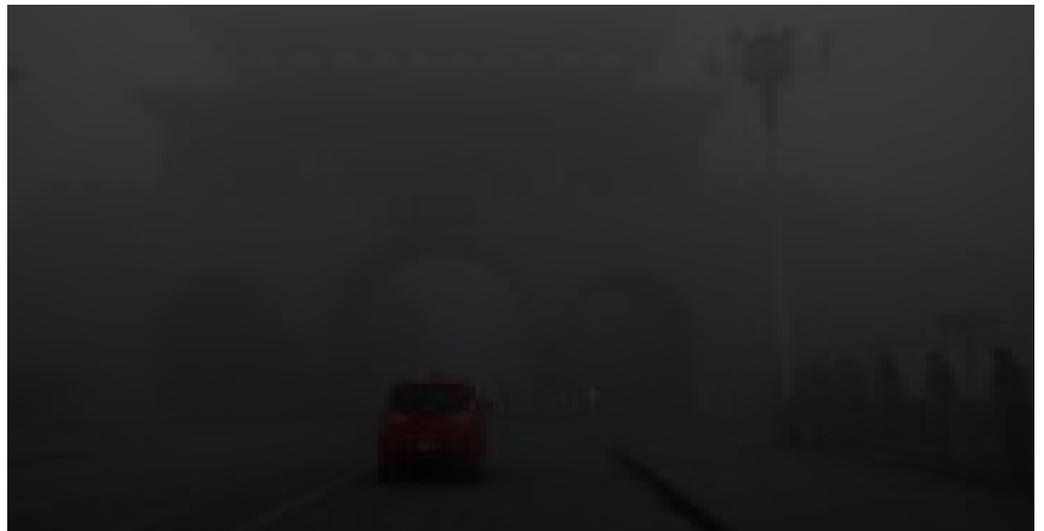


(c) Saturation

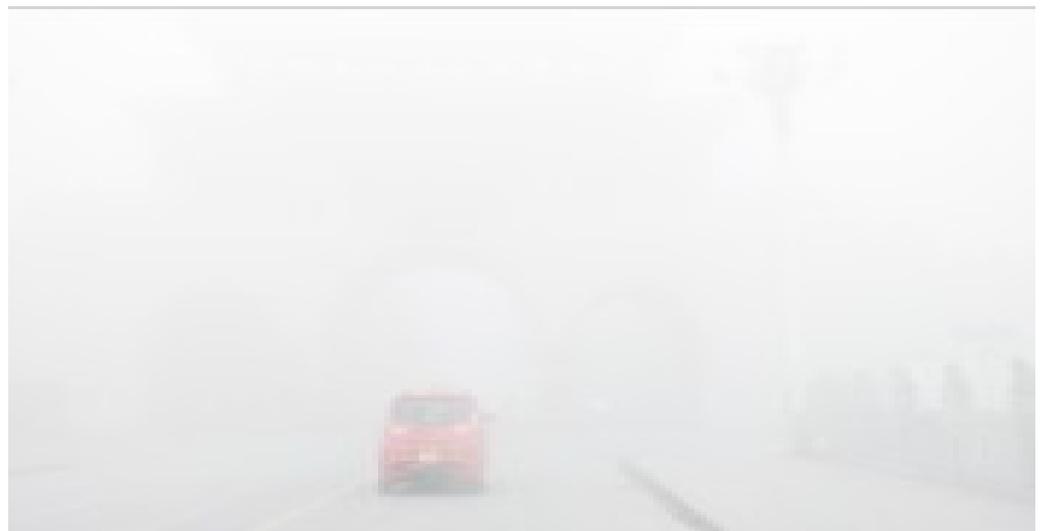
Figure 6. Cont.



(d) Brightness

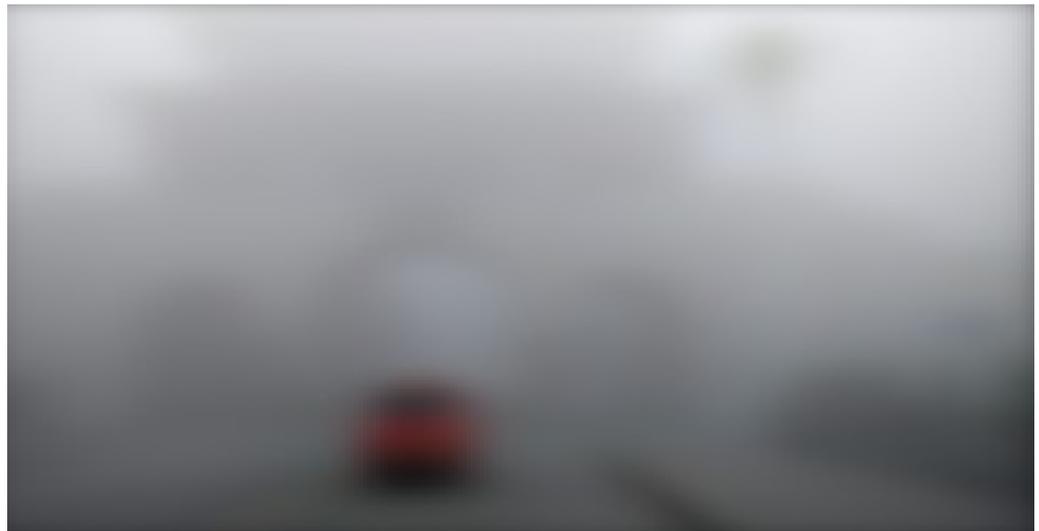


(e) Darkness



(f) Exposure

Figure 6. Cont.



(g) Blur



(h) Noise

Figure 6. Image augmentation techniques applied.

The augmentation techniques are related with the corresponding weather conditions in Table 1.

Table 1. Relation between augmentation techniques and weather conditions.

Hue	Saturation	Brightness	Exposure	Blur	Noise
Foggy	- low saturation to create artificial fog			- high blur to show extreme foggy day	
Sunny		- high brightness to show more day time light	- high gamma exposure to create clear visibility		
Windy				- adjust gaussian blur to show wind storm	

Table 1. Cont.

	Hue	Saturation	Brightness	Exposure	Blur	Noise
Rainy					- adding Blur to show low visibility during rain	- adding line noise to show falling rain effect
Snowy			- very high brightness to create synthetic snow on road			-adding dot noise to show snow flakes or rain drops
Dusty	- adjust color changes to show dust storm	- low saturation to show poor visibility in dust storm by changing color more vivid				
Shady/ Dark			- low brightness to show shade, e.g., underpass, another vehicle shadow	- low gamma exposure to show less visibility at night		
Cloudy			-low brightness to show cloudy day			
color change	- sunrise - sunset - road illumination and reflection		- lowest values to show night time			

4.4. Customized Yolov4 Detector

In the section below, we outline our approach which uses YOLOv4 as our baseline architecture. The reason being, high processing frame rate and state-of-the-art accuracy offered by the model as depicted in Figure 7. For the MS COCO, it obtains an accuracy of 43.5% AP (65.7% AP50) with an approximate inference speed of 65 FPS on the Tesla V100 [104].

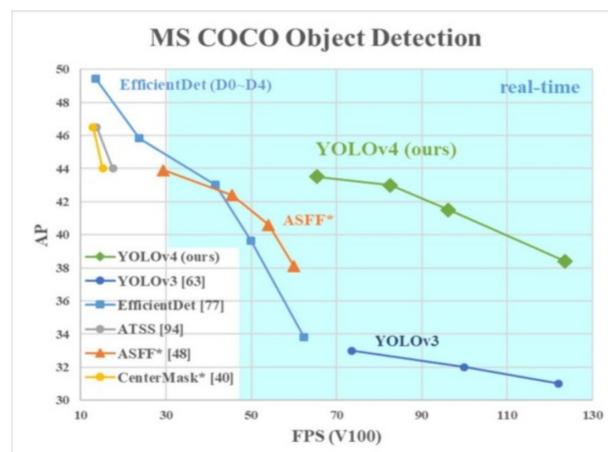


Figure 7. The suggested YOLOv4 is compared to various cutting-edge object detectors. YOLOv4 performs equally well as EfficientDet while operating twice as quickly, increases the AP and FPS of YOLOv3 by 10% and 12%, respectively.

4.4.1. Overall Model Design

There are a number of parts that make up our model architecture, but in general they are as follows: The network will be fed a set of training images, which makes up the input, which comes first. These photos are processed by the GPU in concurrent batches. Following the Neck and Backbone, which carry out feature extraction and aggregation, respectively. Our model contains eight convolution layers with 16, 32, 64, 128, 256 and three 512 filters. The object detector is the collective name for the detection head and detection neck. Finally, the head is responsible for detection and prediction. The primary responsibility for detection rests with the head (both localization and classification).

4.4.2. CSPDarknet53 as Backbone Network

To detect multi scale vehicle requires a focused and increased recognition accuracy of a detector. Therefore, we selected CSPDarknet53 as backbone network. The YOLO series deep learning models' most notable benefit is their ability to swiftly and accurately recognise and classify targets. Figure 8 shows the complete structure of YOLOv4 CSPDarknet.

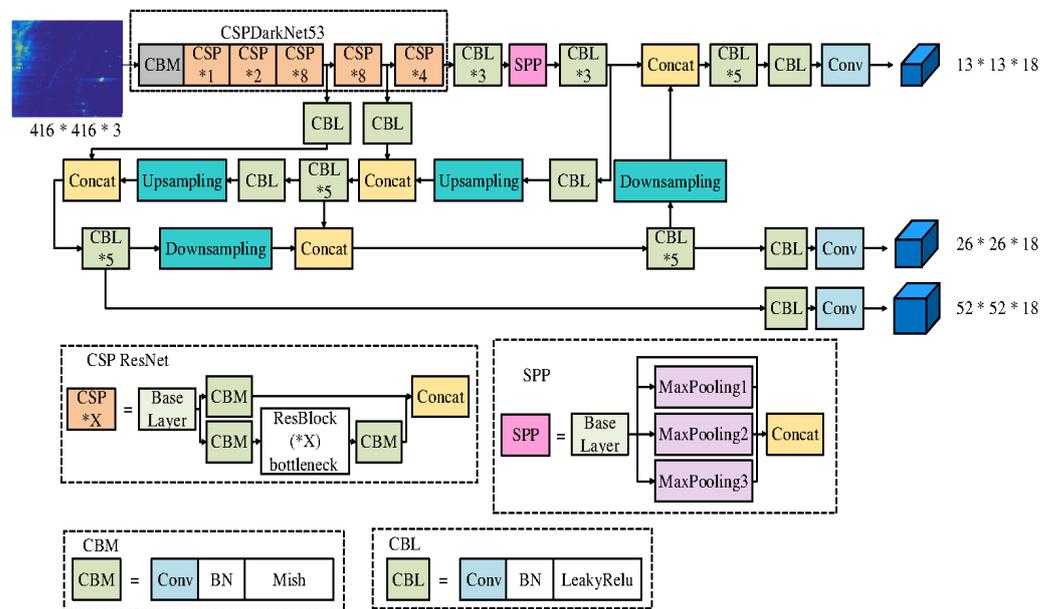


Figure 8. The architecture of CSPDarknet53 from Baseline paper.

Figure 9 shows our modified architecture of the network with layered approach.

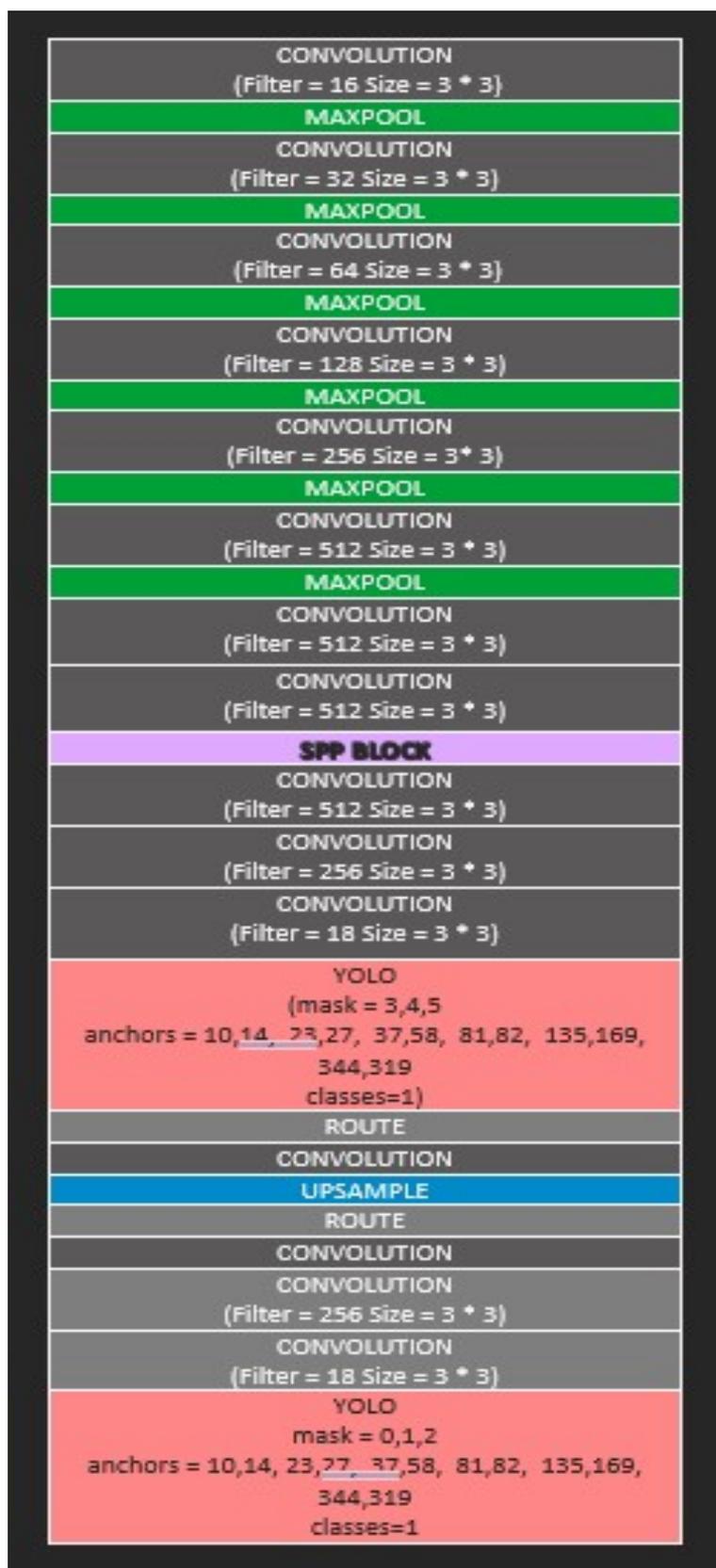


Figure 9. Architecture of our custom YOLOv4-SPP detector.

4.4.3. Adding Spatial Pyramid Pooling Block

From the Bag of specials we add double special pyramid pooling blocks. A pooling layer called spatial pyramid pooling (SPP) frees the network's fixed-size constraint, meaning a CNN no longer needs a fixed-size input image. In spatial pyramid pooling fine grained features are pooled from any arbitrary sections (sub-images) to create fixed-length representations for training the detectors by computing the feature maps from the complete image just once. By doing this the spatial information in an image is preserved. Hence, the SPP module may significantly expand the receptive field, enhance local and global vehicle recognition accuracy in a complicated environment, and hence enhance detection effectiveness. Figure 10 illustrates the placement of SPP net layer in the overall network.

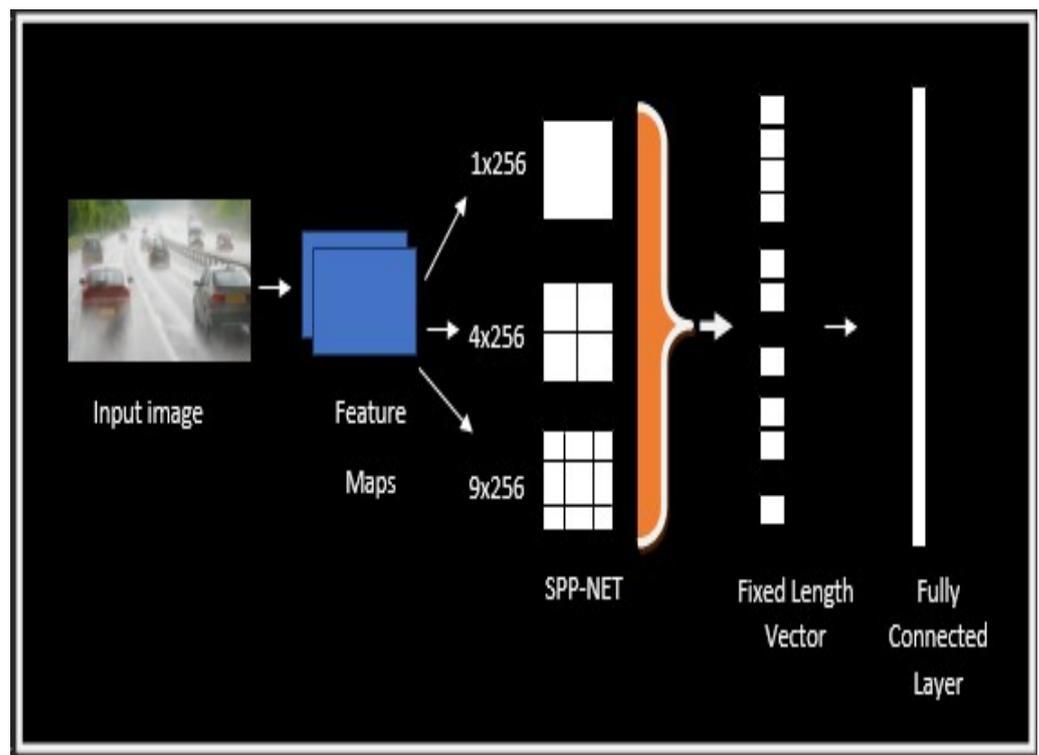


Figure 10. The role of spatial pyramid pooling network: 256 represents the last convolution layer filter size.

4.4.4. Removing Batch Normalization

The camera is frequently set up in a specific spot for traffic surveillance purposes. As a result, since the background of photographs is fixed, little preliminary processing and normalisation is required. Hence, by removing the batch normalization layers we are able to reduce model complexity by minimizing the number of required model parameters.

5. Experimental Analysis

5.1. Training Setup

The training of the vehicle detector model was conducted on Google Colab using TensorFlow framework. The reason for this adoption was free availability of GPU back-end. Additionally, adopting a Python code here is simple as most of the required packages are built in.

5.2. Evaluation of the Detector

To evaluate the accuracy of our object detector we took mAP (mean Average Precision) as our main measurement metric. To fully understand mAP we must first comprehend

certain fundamental terms such as confidence score, IoU (Intersection over union), precision, recall with respect to our study.

- Confidence Score: It displays the likelihood that bounding box holds a vehicle. It is predicted by the last layer of the detector, i.e., a classifier.
- Intersection over Union (IoU): This metric counts the area encompassed by Actual bounding box (B_a) and predicted bounding box (B_p), We used a predefined threshold value of 0.5 for single class to classify if the prediction is false or true positive.

$$IoU = \frac{\text{area_of_intersection}}{\text{area_of_union}} \quad (3)$$

$$= \frac{\text{area_of_}B_a \cap B_p}{\text{area_of_}B_a \cup B_p} \quad (4)$$

- Precision: is the ratio of total true positive and sum of true positive and false positive.

$$\text{precision} = \frac{T_p}{T_p + F_p} \quad (5)$$

- Recall: is the ratio of true positives and sum of true positive and false negative.

$$\text{recall} = \frac{T_p}{T_p + F_n} \quad (6)$$

- Minimum Average Precision (mAP@0.5):

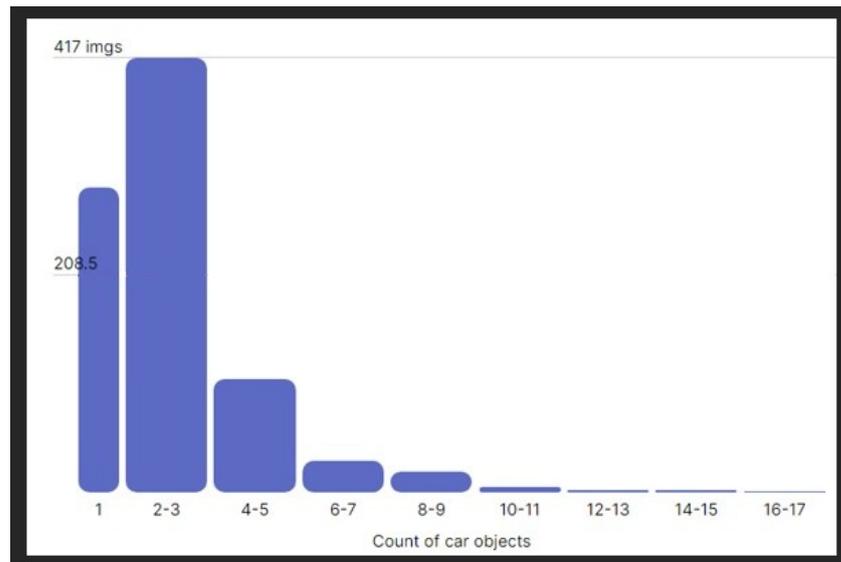
5.2.1. Results

Table 2 shows the split of images into training, validation and testing set. The split was conducted using the ratio 70% for training, 15% for validation set and 15% for testing.

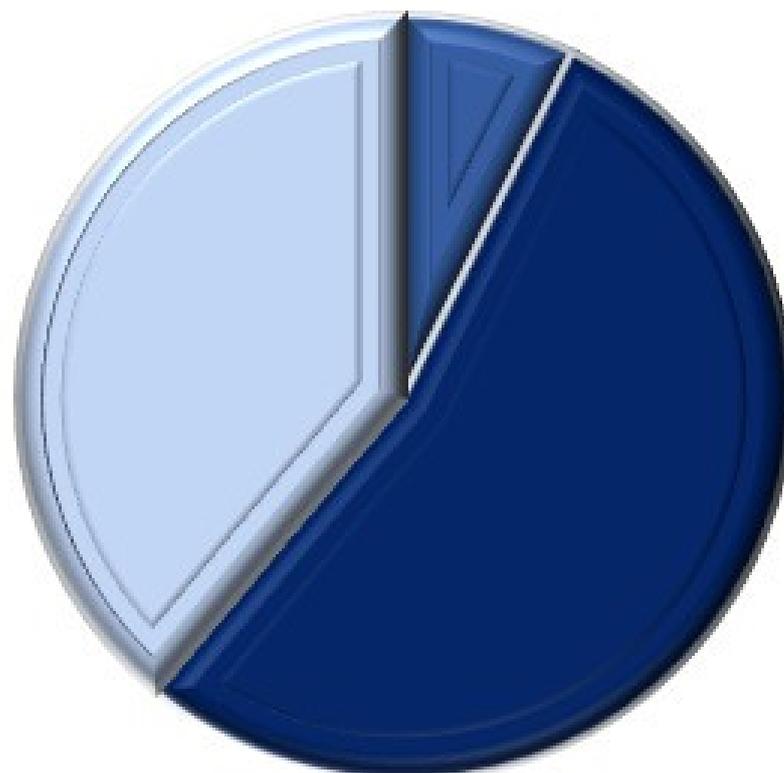
Table 2. Number of images and car instances in our Augmented DAWN Dataset.

	Training Set	Validation Set	Testing Set
Number of images	1500	424	213
Percentage	70	20	10
Number of car instances	2048	235	101

One of the main objective of this research was to detect not only clear, large and visible vehicles but also to detect unclear, small size or occluded vehicles. The images in data set contains vehicles of variable sizes. Figure 11a represents the distribution of vehicles with respect to size in the complete dataset using a pie chart. In addition, Figure 11b shows the histogram of vehicle instances per image. As seen in the bar graph, most of the images contains two to three vehicles per image. Around 2000 images contains single vehicle. There were very less images which contains more than six or seven vehicles.



(a) Histogram of Vehicle Count By Image



Small Medium Large

(b) Distribution of images with respect to size

Figure 11. The vehicle distribution and count of vehicles per image.

Figure 12 displays the training loss and mean average precision in 2000 epochs with our augmented DAWN Dataset.

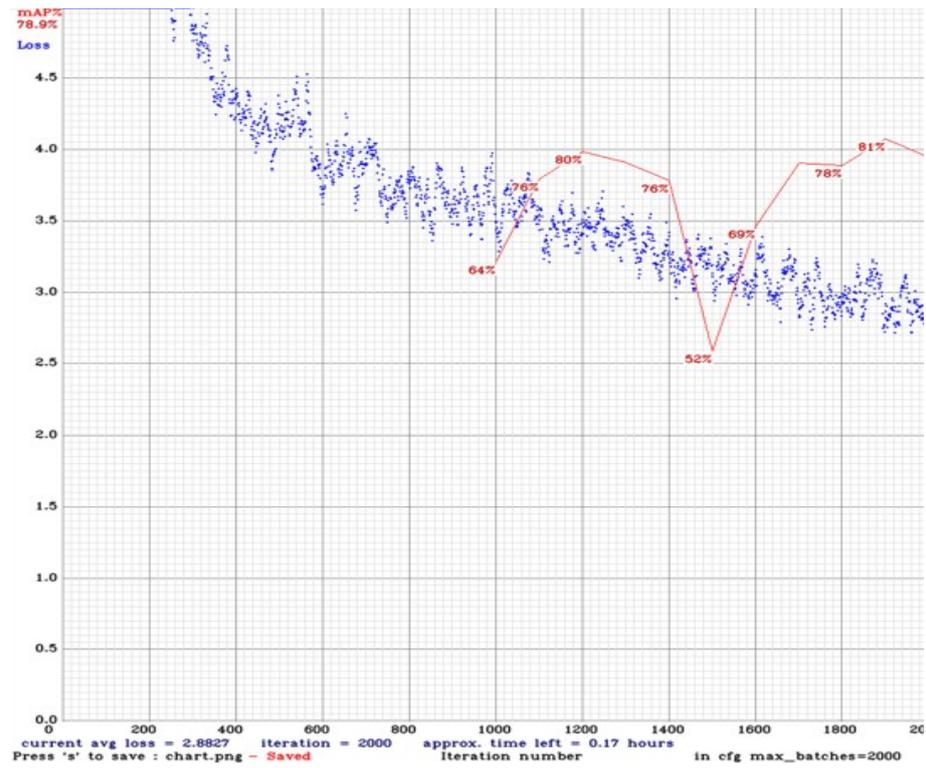


Figure 12. mAP and training Loss graph of first 2000 epochs.

Figure 13 shows the comparison of training and validation loss. During experiment it was observed that the training loss of un-augmented DAWN data set was quite high. However, the loss curve was decreased by the data set size through augmentation and fine tuning the hyper = parameter values.

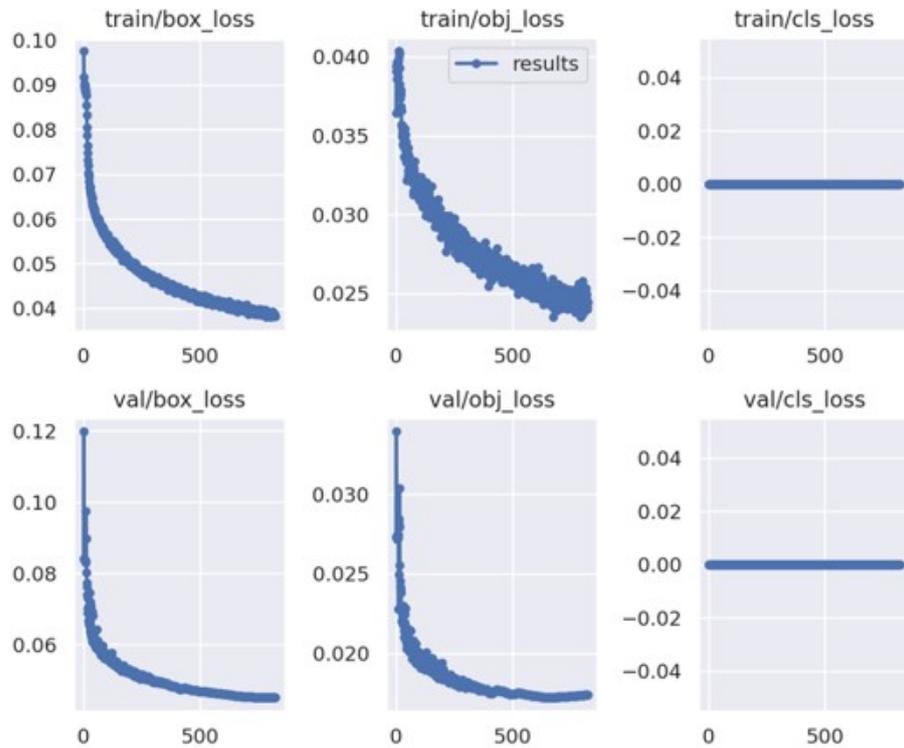


Figure 13. Comparison training vs. validation Loss with and without augmentation.

Table 3 shows the hyper parameter values of our model used during the training.

Table 3. Model Hyper parameters set during training.

Name	Value
batch	64
subdivisions	24
momentum = 0.949	0.949
decay = 0.0005	0.005
learning_rate = 0.001	0.001
max_batches = 2000	2000
steps	1600.0, 1800.0

Figure 14 shows the precision and recall curve of training.

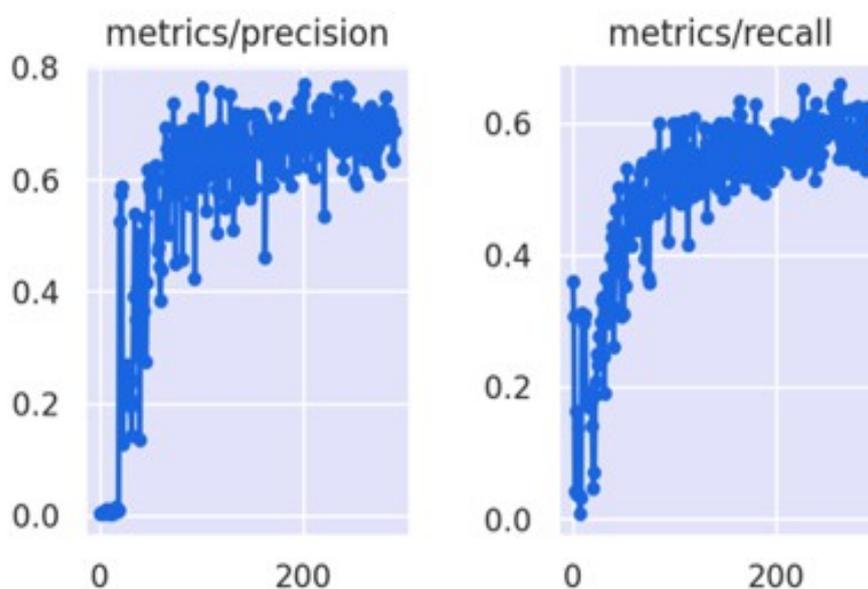


Figure 14. Precision and recall curves.

IoU is calculated for each prediction with regard to each ground truth box in the image. Table 4 shows the average intersection over union values calculated after taking 10 random sample images from each four categories, i.e., snow, fog, dust and rain and then taking their mean. The results show that almost for each weather condition the average IoU is much greater than >0.5 which is considered as excellent prediction.

Table 4. Average IoU values calculated by taking 10 random images from each weather condition.

	Fog	Snow	Rain	Sand
Average IoU	0.892	0.914	0.966	0.972

5.2.2. Result Comparison and Discussion

The total training time with 2000 images using YOLOv2 detector was 6 h. The Tiny Yolo v3 on the same data set took around 1 and a half hour in training which was comparatively faster than any other detector. The training time with our customized Yolov4 detector was 2.5 h, which was half less than the baseline Yolov4 detector without Spatial pyramid pooling layer. Table 5 shows the comparison of the training time in hours. and also the mAp (mean average precision) of our model compared with three other variants belonging to same family.

Table 5. Comparison of mAP of our customized Yolo v4 with three other variants of same family including Yolov2, Tiny Yolov3, baseline Yolov4 and customized Yolo v4.

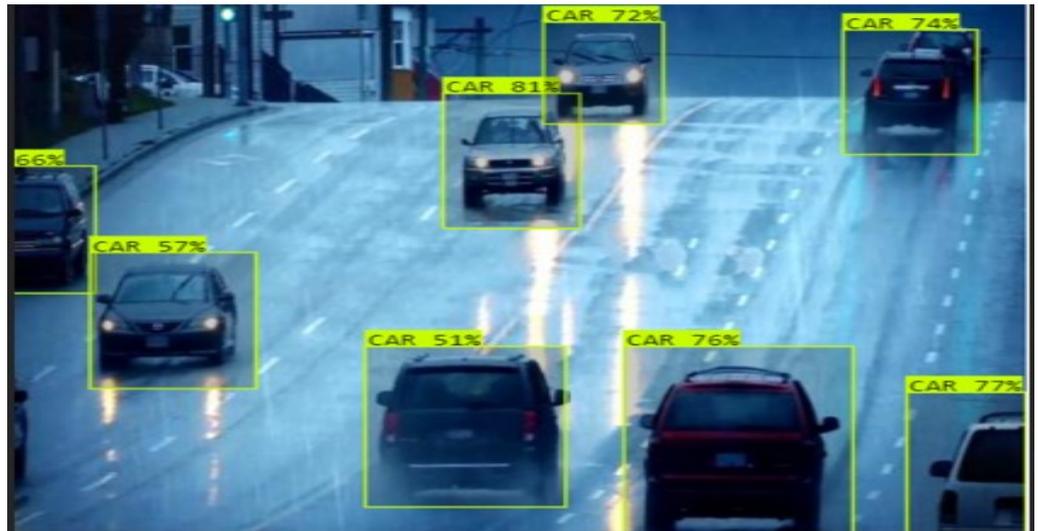
	Yolo v2 Hour	Tiny Yolov3	Yolov4	Customized Yolo v4
Training Time hours	6	1.5	4	2.5
mAP@0.5 (%)	79	64	89	82

It is evident from the table that our model has optimized training time with good mean average precision of the detector. The Table 5 shows the progression of average precision of our model compared with four other baseline variants. The Figure 15 shows the associated line chart. It is clear that our model reaches the average precision of 82% on the same DAWN Dataset, the YOLOv2, YOLOv3, Tiny YOLOv3 and basic YOLOv4 achieves 57%, 74%, 66% and 80% respectively. Table 6 shows the average precision score of vehicle detection under the four weather conditions. It can be inferred from the values that large size vehicles have a high precision score, whereas the smallest vehicles have comparatively low precision values still, although these scores are still good.

Table 6. Detection of vehicles under different weather conditions with their average precision values.

	Small Vehicles	Medium Size Vehicles	Large Size Vehicles
Fog	78.43	76.45	78.32
Sand	76.35	77.62	79.81
Rain	75.88	76.61	77.34
Snow	77.51	77.23	78.32

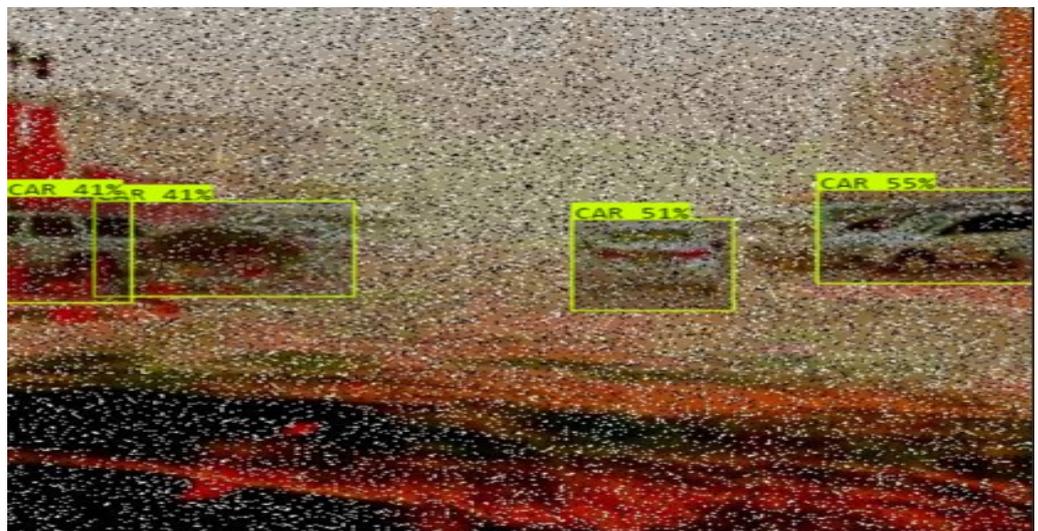
When we see the testing results as shown in Figure 15a–f, we observe that the prediction scores are quite good and high. Figure 15a shows the far away vehicles in rainy condition have confidence score of 74–81% and a nearby vehicle but occluded with fog have score of 51%. The same fog and low illumination testing accuracy can be observed in Figure 15b, where the bounding box has accuracy of 81%. Figure 15c shows our augmented image with high added noise to make visibility very challenging. However, it can be observed that almost all the vehicles present in the image have been detected with prediction score above 50%. Figure 15d–f shows nighttime images with low illumination, fog and glare respectively. All the vehicles in the images were detected with 60% to 70% prediction score. The last two test images show detection in snow-storm and dust-storm conditions with prediction score of almost 70 to 74%.



(a) Rainy

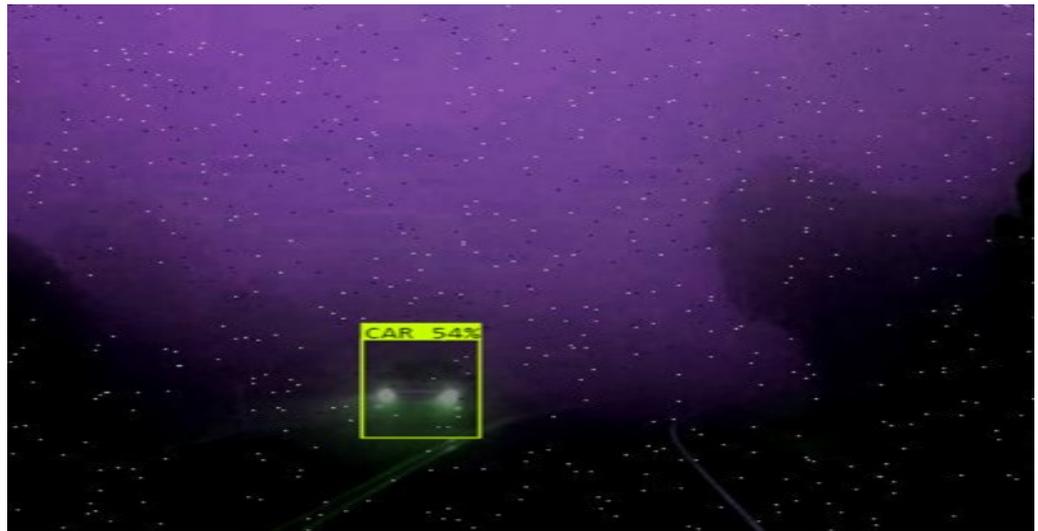


(b) Fog

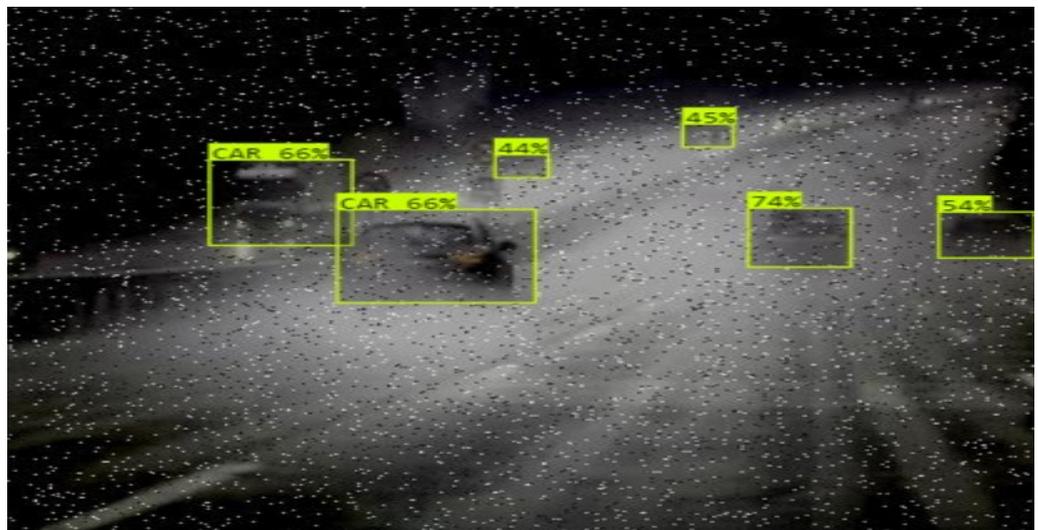


(c) Poor Visibility

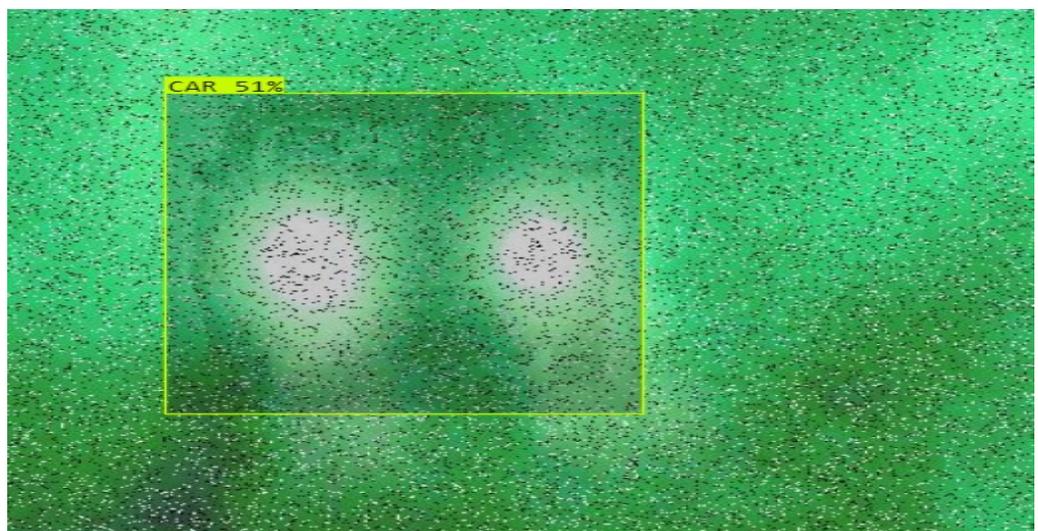
Figure 15. Cont.



(d) Low Illumination



(e) Fog and Darkness

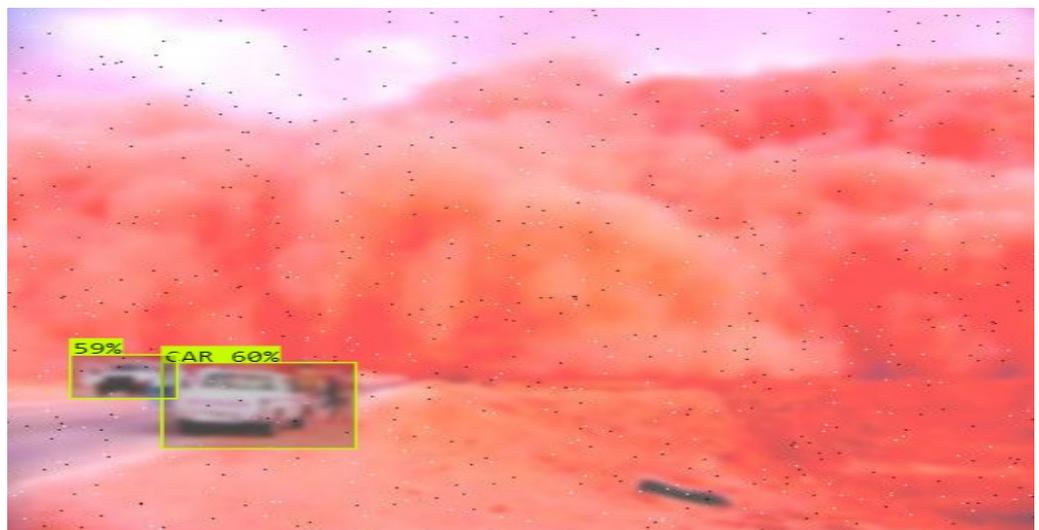


(f) Night and Poor Visibility

Figure 15. Cont.



(g) Snow and poor visibility



(h) Dust storm

Figure 15. Sample images tested with confidence score.

6. Conclusions

Based on Yolov4 CSPDarknet53 with SPP-NET, we propose a modified vehicle detector architecture which can detect vehicles in extremely challenging situations. We augmented DAWN Dataset with added noise, blur and varying hue, saturation, brightness and darkness to make the situation more worse. However, it is evident that our detector is able to predict the vehicles which are almost non detectable with the other baseline detectors. Our model achieve the mAP of 81%, which is higher than four other variants of YOLO family including YOLOv2, Yolov3, Tiny Yolo and Yolov4. In addition, the findings indicate that practically every meteorological condition has an average IoU that is significantly greater than >0.5 , which is regarded as an excellent forecast. Lastly, we optimized the training time by removing the batch normalization layers since our model was using static background and we added a spatial pyramid pooling layer between the eighth and ninth convolution layer before the first YOLO detector layer. In future, we plan to use generative neural networks for data augmentation and create more realistic synthetic scenes representing challenging on road scenarios and then public the data set. Moreover, we decide to extend the vehicle detection algorithm for vehicle trajectory analysis where using

vehicle location, speed, acceleration and jerk, as time functions, normal and abnormal vehicle motion patterns can be detected using video dataset.

Author Contributions: Conceptualization, M.H. and N.Z.J., software, M.H., N.Z.J., F.A. and M.K.A.; validation, M.H., N.Z.J., F.A. and M.K.A.; formal analysis, M.H., N.Z.J., F.A. and M.K.A.; investigation, M.H., N.Z.J., F.A. and M.K.A.; resources, M.H., N.Z.J., F.A. and M.K.A.; data curation, M.H., N.Z.J., F.A. and M.K.A.; writing—original draft preparation, M.H., N.Z.J., F.A. and M.K.A.; visualization, M.H., N.Z.J., F.A. and M.K.A.; supervision, M.H. and N.Z.J., project administration, M.H., N.Z.J., F.A. and M.K.A.; funding acquisition, M.H., N.Z.J., F.A. and M.K.A. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the Deanship of Scientific Research at Jouf University under grant No (DSR-2021-02-0324).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data will be available on request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chen, L.; Li, S.; Bai, Q.; Yang, J.; Jiang, S.; Miao, Y. Review of image classification algorithms based on convolutional neural networks. *Remote. Sens.* **2021**, *13*, 4712. [\[CrossRef\]](#)
2. Khairandish, M.; Sharma, M.; Jain, V.; Chatterjee, J.; Jhanjhi, N. A hybrid CNN-SVM threshold segmentation approach for tumor detection and classification of MRI brain images. *IRBM* **2021**, *43*, 290–299. [\[CrossRef\]](#)
3. Humayun, M.; Sujatha, R.; Almuayqil, S.N.; Jhanjhi, N. A Transfer Learning Approach with a Convolutional Neural Network for the Classification of Lung Carcinoma. *Healthcare* **2022**, *10*, 1058. [\[CrossRef\]](#)
4. He, T.; Zhang, Z.; Zhang, H.; Zhang, Z.; Xie, J.; Li, M. Bag of tricks for image classification with convolutional neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2019; pp. 558–567.
5. Mahmood, A.; Bennamoun, M.; An, S.; Sohel, F.A.; Boussaid, F.; Hovey, R.; Kendrick, G.A.; Fisher, R.B. Deep image representations for coral image classification. *IEEE J. Ocean. Eng.* **2018**, *44*, 121–131. [\[CrossRef\]](#)
6. Yap, J.; Yolland, W.; Tschandl, P. Multimodal skin lesion classification using deep learning. *Exp. Dermatol.* **2018**, *27*, 1261–1267. [\[CrossRef\]](#) [\[PubMed\]](#)
7. Sharma, R.; Singh, A.; Kavita; Jhanjhi, N.; Masud, M.; Jaha, E.S.; Verma, S. Plant Disease Diagnosis and Image Classification Using Deep Learning. *CMC Comput. Mater. Contin.* **2022**, *71*, 2125–2140. [\[CrossRef\]](#)
8. Khalid, N.; Shahrol, N.A. Evaluation the accuracy of oil palm tree detection using deep learning and support vector machine classifiers. In Proceedings of the IOP Conference Series: Earth and Environmental Science, Tashkent, Uzbekistan, 18–20 May 2022; IOP Publishing: Bristol, UK, 2022; Volume 1051, p. 12028.
9. Saeed, S.; Jhanjhi, N.Z.; Naqvi, M.; Humyun, M.; Ahmad, M.; Gaur, L. Optimized Breast Cancer Premature Detection Method with Computational Segmentation: A Systematic Review Mapping. *Approaches Appl. Deep. Learn. Virtual Med. Care* **2022**, 24–51. [\[CrossRef\]](#)
10. Khalil, M.I.; Humayun, M.; Jhanjhi, N.; Talib, M.; Tabbakh, T.A. Multi-class segmentation of organ at risk from abdominal CT images: A deep learning approach. In *Intelligent Computing and Innovation on Data Science*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 425–434.
11. Dash, S.; Verma, S.; Kavita; Jhanjhi, N.; Masud, M.; Baz, M. Curvelet Transform Based on Edge Preserving Filter for Retinal Blood Vessel Segmentation. *CMC Comput. Mater. Contin.* **2022**, *71*, 2459–2476. [\[CrossRef\]](#)
12. Deng, J.; Zhong, Z.; Huang, H.; Lan, Y.; Han, Y.; Zhang, Y. Lightweight semantic segmentation network for real-time weed mapping using unmanned aerial vehicles. *Appl. Sci.* **2020**, *10*, 7132. [\[CrossRef\]](#)
13. Hofmarcher, M.; Unterthiner, T.; Arjona-Medina, J.; Klambauer, G.; Hochreiter, S.; Nessler, B. Visual scene understanding for autonomous driving using semantic segmentation. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 285–296.
14. Tan, M.; Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In Proceedings of the International Conference on Machine Learning, Nanchang, China, 21–23 June 2019; pp. 6105–6114.
15. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 213–229.

16. Gouda, W.; Sama, N.U.; Al-Waakid, G.; Humayun, M.; Jhanjhi, N.Z. Detection of Skin Cancer Based on Skin Lesion Images Using Deep Learning. *Healthcare* **2022**, *10*, 1183. [[CrossRef](#)]
17. Ramamoorthy, M.; Qamar, S.; Manikandan, R.; Jhanjhi, N.Z.; Masud, M.; AlZain, M.A. Earlier Detection of Brain Tumor by Pre-Processing Based on Histogram Equalization with Neural Network. *Healthcare* **2022**, *10*, 1218. [[CrossRef](#)] [[PubMed](#)]
18. Saeed, S.; Abdullah, A.; Jhanjhi, N.; Naqvi, M.; Nayyar, A. New techniques for efficiently k-NN algorithm for brain tumor detection. *Multimed. Tools Appl.* **2022**, *81*, 18595–18616. [[CrossRef](#)]
19. Attaullah, M.; Ali, M.; Almufareh, M.F.; Ahmad, M.; Hussain, L.; Jhanjhi, N.; Humayun, M. Initial Stage COVID-19 Detection System Based on Patients' Symptoms and Chest X-Ray Images. *Appl. Artif. Intell.* **2022**, *36*, 1–20. [[CrossRef](#)]
20. Oh, S.W.; Lee, J.Y.; Xu, N.; Kim, S.J. Video object segmentation using space-time memory networks. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 9226–9235.
21. Sagues-Tanco, R.; Benages-Pardo, L.; López-Nicolás, G.; Llorente, S. Fast synthetic dataset for kitchen object segmentation in deep learning. *IEEE Access* **2020**, *8*, 220496–220506. [[CrossRef](#)]
22. Wu, B.; Wan, A.; Yue, X.; Keutzer, K. Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018; pp. 1887–1893.
23. Liu, X.; Dai, B.; He, H. Real-time object segmentation for visual object detection in dynamic scenes. In Proceedings of the 2011 International Conference of Soft Computing and Pattern Recognition (SoCPaR), Dalian, China, 14–16 October 2011; pp. 423–428.
24. Chen, Z.; Zhang, Z.; Bu, Y.; Dai, F.; Fan, T.; Wang, H. Underwater object segmentation based on optical features. *Sensors* **2018**, *18*, 196. [[CrossRef](#)]
25. Gu, J.; Bellone, M.; Sell, R.; Lind, A. Object segmentation for autonomous driving using iseAuto data. *Electronics* **2022**, *11*, 1119. [[CrossRef](#)]
26. Fukuda, M.; Okuno, T.; Yuki, S. Central Object Segmentation by Deep Learning to Continuously Monitor Fruit Growth through RGB Images. *Sensors* **2021**, *21*, 6999. [[CrossRef](#)]
27. Kim, B.; Lee, J.; Kang, J.; Kim, E.S.; Kim, H.J. Hotr: End-to-end human-object interaction detection with transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 74–83.
28. Yao, B.; Fei-Fei, L. Recognizing human-object interactions in still images by modeling the mutual context of objects and human poses. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 1691–1703.
29. Zheng, C.; Zhu, S.; Mendieta, M.; Yang, T.; Chen, C.; Ding, Z. 3d human pose estimation with spatial and temporal transformers. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Nashville, TN, USA, 20–25 June 2021; pp. 11656–11665.
30. Tompson, J.J.; Jain, A.; LeCun, Y.; Bregler, C. Joint training of a convolutional network and a graphical model for human pose estimation. *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 1799–1807.
31. Araújo, V.M.; Shukla, A.; Chion, C.; Gambis, S.; Michaud, R. Machine-Learning Approach for Automatic Detection of Wild Beluga Whales from Hand-Held Camera Pictures. *Sensors* **2022**, *22*, 4107. [[CrossRef](#)]
32. Sharma, N.; Scully-Power, P.; Blumenstein, M. Shark detection from aerial imagery using region-based CNN, a study. In Proceedings of the Australasian Joint Conference on Artificial Intelligence, Wellington, New Zealand, 11–14 December 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 224–236.
33. Tang, Y.; Wang, J.; Wang, X.; Gao, B.; Dellandréa, E.; Gaizauskas, R.; Chen, L. Visual and semantic knowledge transfer for large scale semi-supervised object detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 3045–3058. [[CrossRef](#)] [[PubMed](#)]
34. Sujatha, R.; Chatterjee, J.M.; Jhanjhi, N.; Brohi, S.N. Performance of deep learning vs. machine learning in plant leaf disease detection. *Microprocess. Microsyst.* **2021**, *80*, 103615. [[CrossRef](#)]
35. Asmar, D.C.; Zelek, J.S.; Abdallah, S.M. Seeing the trees before the forest [natural object detection]. In Proceedings of the 2nd Canadian Conference on Computer and Robot Vision (CRV'05), Victoria, BC, Canada, 9–11 May 2005; pp. 587–593.
36. Oh, S.; Chang, A.; Ashapure, A.; Jung, J.; Dube, N.; Maeda, M.; Gonzalez, D.; Landivar, J. Plant counting of cotton from UAS imagery using deep learning-based object detection framework. *Remote. Sens.* **2020**, *12*, 2981. [[CrossRef](#)]
37. Kusumo, B.S.; Heryana, A.; Mahendra, O.; Pardede, H.F. Machine learning-based for automatic detection of corn-plant diseases using image processing. In Proceedings of the 2018 International Conference on Computer, Control, Informatics and Its Applications (IC3INA), Tangerang, Indonesia, 1–2 November 2018; pp. 93–97.
38. Khan, N.A.; Jhanjhi, N.; Brohi, S.N.; Usmani, R.S.A.; Nayyar, A. Smart traffic monitoring system using unmanned aerial vehicles (UAVs). *Comput. Commun.* **2020**, *157*, 434–443. [[CrossRef](#)]
39. Alkinani, M.H.; Almazroi, A.A.; Jhanjhi, N.; Khan, N.A. 5G and IoT based reporting and accident detection (RAD) system to deliver first aid box using unmanned aerial vehicle. *Sensors* **2021**, *21*, 6905. [[CrossRef](#)]
40. Song, H.; Liang, H.; Li, H.; Dai, Z.; Yun, X. Vision-based vehicle detection and counting system using deep learning in highway scenes. *Eur. Transp. Res. Rev.* **2019**, *11*, 1–16. [[CrossRef](#)]
41. Arabi, S.; Haghghat, A.; Sharma, A. A deep-learning-based computer vision solution for construction vehicle detection. *Comput. Aided Civ. Infrastruct. Eng.* **2020**, *35*, 753–767. [[CrossRef](#)]
42. Wang, H.; Wang, H.; Xu, K. Evolutionary recurrent neural network for image captioning. *Neurocomputing* **2020**, *401*, 249–256. [[CrossRef](#)]

43. Alzubi, J.A.; Jain, R.; Nagrath, P.; Satapathy, S.; Taneja, S.; Gupta, P. Deep image captioning using an ensemble of CNN and LSTM based deep neural networks. *J. Intell. Fuzzy Syst.* **2021**, *40*, 5761–5769. [[CrossRef](#)]
44. Amirian, S.; Rasheed, K.; Taha, T.R.; Arabnia, H.R. Image captioning with generative adversarial network. In Proceedings of the 2019 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 5–7 December 2019; pp. 272–275.
45. Shi, H.; Li, P.; Wang, B.; Wang, Z. Image captioning based on deep reinforcement learning. In Proceedings of the 10th International Conference on Internet Multimedia Computing and Service, Nanjing, China, 17–19 August 2018; pp. 1–5.
46. Ullah, A.; Ishaq, N.; Azeem, M.; Ashraf, H.; Jhanjhi, N.; Humayun, M.; Tabbakh, T.A.; Almusaylim, Z.A. A survey on continuous object tracking and boundary detection schemes in IoT assisted wireless sensor networks. *IEEE Access* **2021**, *9*, 126324–126336. [[CrossRef](#)]
47. Wang, Q.; Zhang, L.; Bertinetto, L.; Hu, W.; Torr, P.H. Fast online object tracking and segmentation: A unifying approach. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2019; pp. 1328–1338.
48. Meinhardt, T.; Kirillov, A.; Leal-Taixe, L.; Feichtenhofer, C. Trackformer: Multi-object tracking with transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2022; pp. 8844–8854.
49. Kristan, M.; Leonardis, A.; Matas, J.; Felsberg, M.; Pflugfelder, R.; Kämäräinen, J.K.; Danelljan, M.; Zajc, L.Č.; Lukežič, A.; Drbohlav, O.; et al. The eighth visual object tracking VOT2020 challenge results. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 547–601.
50. Wang, Y.; Xu, Z.; Wang, X.; Shen, C.; Cheng, B.; Shen, H.; Xia, H. End-to-end video instance segmentation with transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 8741–8750.
51. Tian, Z.; Shen, C.; Chen, H. Conditional convolutions for instance segmentation. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 282–298.
52. Bolya, D.; Zhou, C.; Xiao, F.; Lee, Y.J. Yolact: Real-time instance segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 9157–9166.
53. Yang, Y.; Li, G.; Wu, Z.; Su, L.; Huang, Q.; Sebe, N. Reverse perspective network for perspective-aware object counting. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 4374–4383.
54. Rahutomo, R.; Perbangsa, A.S.; Lie, Y.; Cenggoro, T.W.; Pardamean, B. Artificial intelligence model implementation in web-based application for pineapple object counting. In Proceedings of the 2019 International Conference on Information Management and Technology (ICIMTech), Jakarta, Indonesia, 19–20 August 2019; Volume 1, pp. 525–530.
55. Zhang, S.; Li, H.; Kong, W. Object counting method based on dual attention network. *IET Image Process.* **2020**, *14*, 1621–1627. [[CrossRef](#)]
56. Dirir, A.; Ignatious, H.; Elsayed, H.; Khan, M.; Adib, M.; Mahmoud, A.; Al-Gunaid, M. An Advanced Deep Learning Approach for Multi-Object Counting in Urban Vehicular Environments. *Future Internet* **2021**, *13*, 306. [[CrossRef](#)]
57. Chayeb, A.; Ouadah, N.; Tobal, Z.; Lakrouf, M.; Azouaoui, O. HOG based multi-object detection for urban navigation. In Proceedings of the 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), Qingdao, China, 8–11 October 2014; pp. 2962–2967. [[CrossRef](#)]
58. Yamauchi, Y.; Matsushima, C.; Yamashita, T.; Fujiyoshi, H. Relational HOG feature with wild-card for object detection. In Proceedings of the 2011 IEEE International Conference on Computer Vision Workshop (ICCV Workshops), Venice, Italy, 22–29 October 2011; pp. 1785–1792. [[CrossRef](#)]
59. Dong, L.; Yu, X.; Li, L.; Hoe, J.K.E. HOG based multi-stage object detection and pose recognition for service robot. In Proceedings of the 2010 11th International Conference on Control Automation Robotics & Vision, Singapore, 7–10 December 2010; pp. 2495–2500.
60. Cheng, G.; Zhou, P.; Yao, X.; Yao, C.; Zhang, Y.; Han, J. Object detection in VHR optical remote sensing images via learning rotation-invariant HOG feature. In Proceedings of the 2016 4th International Workshop on Earth Observation and Remote Sensing Applications (EORSA), Guangzhou, China, 4–6 July 2016; pp. 433–436. [[CrossRef](#)]
61. Ren, H.; Li, Z.N. Object detection using edge histogram of oriented gradient. In Proceedings of the 2014 IEEE International Conference on Image Processing (ICIP), Paris, France, 27–30 October 2014; pp. 4057–4061.
62. Han, F.; Shan, Y.; Cekander, R.; Sawhney, H.S.; Kumar, R. A two-stage approach to people and vehicle detection with hog-based svm. In Proceedings of the Performance Metrics for Intelligent Systems 2006 Workshop, Washington, DC, USA, 11–13 October 2006; pp. 133–140.
63. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection, 2005. In Proceedings of the IEEE Computer Vision and Pattern Recognition, San Diego, CA, USA, 21–23 September 2005.
64. Viola, P.; Jones, M. Rapid object detection using a boosted cascade of simple features. In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2001, Kauai, HI, USA, 8–14 December 2001; Volume 1, p. 1.
65. Li, Y.L.; Wang, S. HAR-Net: Joint learning of hybrid attention for single-stage object detection. *arXiv* **2019**, arXiv:1904.11141.

66. Soo, S. *Object Detection Using Haar-Cascade Classifier*; Institute of Computer Science, University of Tartu: Tartu, Estonia, 2014; Volume 2, pp. 1–12.
67. Jalled, F.; Voronkov, I. Object detection using image processing. *arXiv* **2016**, arXiv:1611.07791.
68. Cuimei, L.; Zhiliang, Q.; Nan, J.; Jianhua, W. Human face detection algorithm via Haar cascade classifier combined with three additional classifiers. In Proceedings of the 2017 13th IEEE International Conference on Electronic Measurement & Instruments (ICEMI), Yangzhou, China, 20–23 October 2017; pp. 483–487.
69. Pawełczyk, M.; Wojtyra, M. Real world object detection dataset for quadcopter unmanned aerial vehicle detection. *IEEE Access* **2020**, *8*, 174394–174409. [[CrossRef](#)]
70. Whitehill, J.; Omlin, C.W. Haar features for FACS AU recognition. In Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition (FGR06), Southampton, UK, 10–12 April 2006; p. 5.
71. Haselhoff, A.; Kummert, A. A vehicle detection system based on haar and triangle features. In Proceedings of the 2009 IEEE Intelligent Vehicles Symposium, Xi'an, China, 3–5 June 2009; pp. 261–266.
72. Chen, D.S.; Liu, Z.K. Generalized Haar-like features for fast face detection. In Proceedings of the 2007 International Conference on Machine Learning and Cybernetics, Cincinnati, OH, USA, 13–15 December 2007; Volume 4, pp. 2131–2135.
73. Yun, L.; Peng, Z. An automatic hand gesture recognition system based on Viola-Jones method and SVMs. In Proceedings of the 2009 Second International Workshop on Computer Science and Engineering, Qingdao, China, 28–30 October 2009; Volume 2, pp. 72–76.
74. Sawas, J.; Petillot, Y.; Pailhas, Y. Cascade of boosted classifiers for rapid detection of underwater objects. In Proceedings of the European Conference on Underwater Acoustics, Istanbul, Turkey, 5 July 2010; Volume 164.
75. Nguyen, T.; Park, E.; Han, J.; Park, D.C.; Min, S.Y. Object detection using scale invariant feature transform. In *Genetic and Evolutionary Computing*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 65–72.
76. Geng, C.; Jiang, X. SIFT features for face recognition. In Proceedings of the 2009 2nd IEEE International Conference on Computer Science and Information Technology, Beijing, China, 8–11 August 2009; pp. 598–602.
77. Piccinini, P.; Prati, A.; Cucchiara, R. Real-time object detection and localization with SIFT-based clustering. *Image Vis. Comput.* **2012**, *30*, 573–587. [[CrossRef](#)]
78. Zhao, W.L.; Ngo, C.W. Flip-invariant SIFT for copy and object detection. *IEEE Trans. Image Process.* **2012**, *22*, 980–991. [[CrossRef](#)]
79. Najva, N.; Bijoy, K.E. SIFT and tensor based object detection and classification in videos using deep neural networks. *Procedia Comput. Sci.* **2016**, *93*, 351–358. [[CrossRef](#)]
80. Sun, S.W.; Wang, Y.C.F.; Huang, F.; Liao, H.Y.M. Moving foreground object detection via robust SIFT trajectories. *J. Vis. Commun. Image Represent.* **2013**, *24*, 232–243. [[CrossRef](#)]
81. Sakai, Y.; Oda, T.; Ikeda, M.; Barolli, L. An object tracking system based on sift and surf feature extraction methods. In Proceedings of the 2015 18th International Conference on Network-Based Information Systems, Washington, DC, USA, 2–4 September 2015; pp. 561–565.
82. Chen, Z.; Chen, K.; Chen, J. Vehicle and pedestrian detection using support vector machine and histogram of oriented gradients features. In Proceedings of the 2013 International Conference on Computer Sciences and Applications, Kunming, China, 21–23 September 2013; pp. 365–368.
83. Anwar, M.A.; Tahir, S.F.; Fahad, L.G.; Kifayat, K.K. Image Forgery Detection by Transforming Local Descriptors into Deep-Derived Features. Available online: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4134079 (accessed on 3 July 2022).
84. Patil, S.; Patil, Y. Face Expression Recognition Using SVM and KNN Classifier with HOG Features. In Proceedings of the International Conference on Computing in Engineering & Technology, Virtual, 12–13 February 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 416–424.
85. Gao, S.; Wei, Y.; Xiong, H. Pedestrian detection algorithm based on improved SLIC segmentation and SVM. In Proceedings of the 2022 IEEE 10th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), Chongqing, China, 17–19 June 2022; Volume 10, pp. 771–775.
86. Tseng, H.H.; Yang, M.D.; Saminathan, R.; Hsu, Y.C.; Yang, C.Y.; Wu, D.H. Rice Seedling Detection in UAV Images Using Transfer Learning and Machine Learning. *Remote. Sens.* **2022**, *14*, 2837. [[CrossRef](#)]
87. Yousef, N.; Parmar, C.; Sata, A. Intelligent inspection of surface defects in metal castings using machine learning. *Mater. Today Proc.* **2022**. [[CrossRef](#)]
88. Sharma, V.; Jain, M.; Jain, T.; Mishra, R. License plate detection and recognition using openCV–python. In *Recent Innovations in Computing*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 251–261.
89. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Region-based convolutional networks for accurate object detection and segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *38*, 142–158. [[CrossRef](#)] [[PubMed](#)]
90. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
91. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916. [[CrossRef](#)] [[PubMed](#)]
92. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*. [[CrossRef](#)] [[PubMed](#)]

93. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
94. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
95. Cai, Z.; Vasconcelos, N. Cascade r-cnn: Delving into high quality object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6154–6162.
96. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
97. Peng, Q.; Luo, W.; Hong, G.; Feng, M.; Xia, Y.; Yu, L.; Hao, X.; Wang, X.; Li, M. Pedestrian detection for transformer substation based on gaussian mixture model and YOLO. In Proceedings of the 2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), Zhejiang, China, 11–12 September 2016; Volume 2, pp. 562–565.
98. Liu, C.; Tao, Y.; Liang, J.; Li, K.; Chen, Y. Object detection based on YOLO network. In Proceedings of the 2018 IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, China, 14–16 December 2018; pp. 799–803.
99. Fang, W.; Wang, L.; Ren, P. Tinier-YOLO: A real-time object detection method for constrained environments. *IEEE Access* **2019**, *8*, 1935–1944. [[CrossRef](#)]
100. Tao, J.; Wang, H.; Zhang, X.; Li, X.; Yang, H. An object detection system based on YOLO in traffic scene. In Proceedings of the 2017 6th International Conference on Computer Science and Network Technology (ICCSNT), Dalian, China, 21–22 October 2017; pp. 315–319.
101. He, W.; Huang, Z.; Wei, Z.; Li, C.; Guo, B. TF-YOLO: An improved incremental network for real-time object detection. *Appl. Sci.* **2019**, *9*, 3225. [[CrossRef](#)]
102. Shafiee, M.J.; Chywl, B.; Li, F.; Wong, A. Fast YOLO: A fast you only look once system for real-time embedded object detection in video. *arXiv* **2017**, arXiv:1709.05943.
103. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
104. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
105. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
106. Liu, S.; Wang, J.; Wang, Z.; Yu, B.; Hu, W.; Liu, Y.; Tang, J.; Song, S.L.; Liu, C.; Hu, Y. Brief industry paper: The necessity of adaptive data fusion in infrastructure-augmented autonomous driving system. In Proceedings of the 2022 IEEE 28th Real-Time and Embedded Technology and Applications Symposium (RTAS), Milano, Italy, 4–6 May 2022; pp. 293–296.
107. Yang, X.; Zou, Y.; Chen, L. Operation analysis of freeway mixed traffic flow based on catch-up coordination platoon. *Accid. Anal. Prev.* **2022**, *175*, 106780. [[CrossRef](#)]
108. Munroe, D.T.; Madden, M.G. Multi-class and single-class classification approaches to vehicle model recognition from images. *Proc. AICS* **2005**, 1–11.
109. Tang, Y.; Zhang, C.; Gu, R.; Li, P.; Yang, B. Vehicle detection and recognition for intelligent traffic surveillance system. *Multimed. Tools Appl.* **2017**, *76*, 5817–5832. [[CrossRef](#)]
110. Sang, J.; Wu, Z.; Guo, P.; Hu, H.; Xiang, H.; Zhang, Q.; Cai, B. An improved YOLOv2 for vehicle detection. *Sensors* **2018**, *18*, 4272. [[CrossRef](#)]
111. Arora, N.; Kumar, Y.; Karkra, R.; Kumar, M. Automatic vehicle detection system in different environment conditions using fast R-CNN. *Multimed. Tools Appl.* **2022**, *81*, 18715–18735. [[CrossRef](#)]
112. Othmani, M. A vehicle detection and tracking method for traffic video based on faster R-CNN. *Multimed. Tools Appl.* **2022**, 1–19. [[CrossRef](#)]
113. Nguyen, V.; Tran, D.; Tran, M.; Nguyen, N.; Nguyen, V. Robust vehicle detection under adverse weather conditions using auto-encoder feature. *Int. J. Mach. Learn. Comput* **2020**, *10*, 549–555. [[CrossRef](#)]
114. Singh, A.; Kumar, D.P.; Shivaprasad, K.; Mohit, M.; Wadhawan, A. Vehicle detection and accident prediction in sand/dust storms. In Proceedings of the 2021 International Conference on Computing Sciences (ICCS), Phagwara, India, 4–5 December 2021; pp. 107–111.
115. Hassaballah, M.; Kenk, M.A.; Muhammad, K.; Minaee, S. Vehicle detection and tracking in adverse weather using a deep learning framework. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 4230–4242. [[CrossRef](#)]
116. Kenk, M.A.; Hassaballah, M.; Hameed, M.A.; Bekhet, S. Visibility enhancer: Adaptable for distorted traffic scenes by dusty weather. In Proceedings of the 2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES), Giza, Egypt, 24–26 October 2020; pp. 213–218.
117. Al-Haija, Q.A.; Gharabeh, M.; Odeh, A. Detection in Adverse Weather Conditions for Autonomous Vehicles via Deep Learning. *AI* **2022**, *3*, 303–317. [[CrossRef](#)]
118. Iandola, F.N.; Han, S.; Moskewicz, M.W.; Ashraf, K.; Dally, W.J.; Keutzer, K. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and < 0.5 MB model size. *arXiv* **2016**, arXiv:1602.07360.
119. Miao, Y.; Liu, F.; Hou, T.; Liu, L.; Liu, Y. A nighttime vehicle detection method based on YOLO v3. In Proceedings of the 2020 Chinese Automation Congress (CAC), Shanghai, China, 6–8 November 2020; pp. 6617–6621.

120. Ghosh, R. On-road vehicle detection in varying weather conditions using faster R-CNN with several region proposal networks. *Multimed. Tools Appl.* **2021**, *80*, 25985–25999. [[CrossRef](#)]
121. Hong, F.; Lu, C.H.; Liu, C.; Liu, R.R.; Wei, J. A traffic surveillance multi-scale vehicle detection object method base on encoder-decoder. *IEEE Access* **2020**, *8*, 47664–47674. [[CrossRef](#)]
122. Du, Z.; Yin, J.; Yang, J. Expanding receptive field yolo for small object detection. *J. Phys. Conf. Ser.* **2019**, *1314*, 12202. [[CrossRef](#)]
123. Tzatalin. *LabelImg*, Free Software, MIT License, 2015. Available online: <https://github.com/tzatalin/labelImg> (accessed on 3 July 2022).