Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.044374

ARTICLE

Zero-DCE++ Inspired Object Detection in Less Illuminated Environment Using Improved YOLOv5

Ananthakrishnan Balasundaram[1,*], Anshuman Mohanty[2], Ayesha Shaik[1], Krishnadoss Pradeep[2], Kedalu Poornachary Vijayakumar[2] and Muthu Subash Kavitha[3]

[1]Centre for Cyber Physical Systems, Vellore Institute of Technology (VIT), Chennai, Tamil Nadu, 600127, India

[2]School of Computer Science and Engineering, Vellore Institute of Technology (VIT), Chennai, Tamil Nadu, 600127, India

[3]School of Information and Data Sciences, Nagasaki University, Nagasaki, 8528521, Japan

*Corresponding Author: Ananthakrishnan Balasundaram. Email: balasundaram.a@vit.ac.in

Received: 28 July 2023 Accepted: 07 November 2023 Published: 26 December 2023

ABSTRACT

Automated object detection has received the most attention over the years. Use cases ranging from autonomous driving applications to military surveillance systems, require robust detection of objects in different illumination conditions. State-of-the-art object detectors tend to fare well in object detection during daytime conditions. However, their performance is severely hampered in night light conditions due to poor illumination. To address this challenge, the manuscript proposes an improved YOLOv5-based object detection framework for effective detection in unevenly illuminated nighttime conditions. Firstly, the preprocessing strategies involve using the Zero-DCE++ approach to enhance lowlight images. It is followed by optimizing the existing YOLOv5 architecture by integrating the Convolutional Block Attention Module (CBAM) in the backbone network to boost model learning capability and Depthwise Convolutional module (DWConv) in the neck network for efficient compression of network parameters. The Night Object Detection (NOD) and Exclusively Dark (ExDARK) dataset has been used for this work. The proposed framework detects classes like humans, bicycles, and cars. Experiments demonstrate that the proposed architecture achieved a higher Mean Average Precision (mAP) along with a reduction in model size and total parameters, respectively. The proposed model is lighter by 11.24% in terms of model size and 12.38% in terms of parameters when compared to baseline YOLOv5.

KEYWORDS

Object detection; deep learning; nighttime road scenes; YOLOv5; DWConv; Zero-DCE++; CBAM

1 Introduction

Rapid advancements in Deep Neural Networks have contributed to tremendous breakthroughs in deep learning (DL). Among various DL use cases, computer vision approaches have fueled the emergence of various networks for object detection and classification [1]. Object detectors aim to detect and classify several class instances, such as pedestrians, etc., given an image/video [2,3]. Over the years, researchers have strived to formulate advanced object detection solutions to achieve better detection results with faster inference speeds in the fields of medicine [4], agriculture [5], etc. Vast research scope has led to the development of several benchmark datasets like Pascal VOC [6,7], COCO [8], ImageNet [9], etc. Over the years, DL has gained the attention of various researchers for its significance in various real-time applications. Domains like disaster management have extensively used DL to perform studies on assessing structural conditions [10], predicting natural disasters like earthquakes [11] and floods [12], detecting victims under debris post disasters [13], etc. Similarly, object detection has been revolutionized by the introduction of DL object detectors. Various use cases like anomaly detection for surveillance [14] and healthcare [15] are some of the most common fields which use DL.

During its initial phase, the core idea of object detectors was based on using handcrafted features such as the Deformable part-based model [16], Histogram of Oriented Gradient (HOG) [17], Viola et al. [18], etc. However, the rising popularity of convolutional neural networks (CNN) led to the introduction of various DL detectors. They were broadly categorized into single-stage and two-stage detectors. Single-stage detectors like Single Shot Multibox Detector [19], You Only Look Once [20], Retinanet [21], etc., interpret the detection task as a regression problem by considering the input and learning the co-ordinates of bounding box and class probabilities. These frameworks do not involve the generation of a region proposal in its initial phase and hence, are faster than the two-stage object detectors but have lower detection accuracy [22]. Two-stage detectors like Feature Pyramid Networks (FPN) [23], Fast RCNN [24], Faster RCNN [25], Mask RCNN [26], etc., use Region Proposal Networks to generate Region of Interest as the first step, followed by bounding box regression and object classification using the generated region proposals. They have a higher detection accuracy than one-stage detectors but are typically slower.

Different use cases of object detection have varying challenges. Domains like defect detection in buildings [27], and roads [28] generally fail in low-light conditions due to the presence of shadows or artifacts around the area of interest. Furthermore, use cases like the detection of underwater objects are challenging due to attenuation and scattering of light along with the suspended impurities in water which lead to noise [29]. Therefore, it is essential to devise certain preprocessing strategies to boost the performance of the detectors. In autonomous driving systems, it is crucial to efficiently detect objects in road scenes to ensure the safety of pedestrians and vehicle drivers. Over the last few years, the rise in the volume of vehicles on the road has inevitably exacerbated traffic issues and led to numerous accidents. Over 100 thousand fatalities are associated with car accidents in China yearly, with an estimated 1.5 times higher accident rate during the night than daytime [30]. Earlier developments in automated driving used sensors to gather information about the surrounding environment. However, there were challenges associated with using sensors in varying illuminated conditions while locating the object's coordinates.

Owing to the tremendous development in hardware, especially Graphical Processing Units (GPUs) and multi-core processors, a significant amount of data processing is made possible for the efficient working of DL algorithms. Besides, traditional handcrafted feature-based object detectors have been outperformed by DL frameworks both in terms of detection accuracy and inference speed [31]. Although contemporary detection strategies have achieved good results in normal conditions, several challenges, such as noise, uneven brightness, color, and low contrast in night-time conditions, significantly hamper the detection results [32]. Apart from the detection accuracy, obtaining high detection speed with fewer computation resources is also an essential factor in self-driving systems. It necessitates the requirement to have a robust detection framework that performs accurate detection at a faster rate in low-light environments.

In this study, the image enhancement ability of the ZeroDCE++ is integrated with a lightweight YOLOv5 to accurately detect objects in poorly lit road scenes at night. The proposed approach achieves a boost in detection accuracy and a reduction in the model complexity. This paper uses the given sequence. Section 2 elaborates on several works performed in detecting objects in insufficient light. Section 3 presents a detailed working of different frameworks in the proposed methodology. Section 4 offers a comprehensive stepwise explanation of the proposed work, followed by experimental analysis and results of all considered models in Section 5. Finally, the last section presents the paper summary.

2 Related Works

Efficient real-time detection of objects in outdoor nighttime environments is crucial for autonomous driving applications, surveillance systems, wildlife detection, etc. Researchers have recently proposed several novel detection strategies that have achieved excellent results [33]. This section highlights a detailed analysis of the evolution of various research works carried out in this direction.

Traditional feature learning-based object detectors such as HOG, Scale Invariant Feature Transform (SIFT) [34], etc., work based on manual feature extraction. Extracted features are passed to the classifier to identify the instances of a given class. Munian et al. [35] presented a HOG-based approach to detect wild animals at night. Using thermal images, HOG was used to extract animal features by normalizing the radiometric images and estimating the pixels' magnitude and gradient. A 1D CNN was fed with the input features, and the object's existence was evaluated using binary cross entropy. Their framework achieved a detection accuracy of 91%. A similar analysis was performed by Baek et al. [36] to detect pedestrians at night. Their work proposed a novel framework combining thermal-position-intensity-histogram (TPIHOG) and additive kernel support vector machine. TPIHOG was observed to have a more robust discriminative power than HOG due to its gradient cell location for each orientation channel. Their approach was performed on the KAIST pedestrian dataset and showed a higher detection accuracy and speed than other models.

Further, the focus shifting towards leveraging the power of CNNs led to the formulation of numerous two-stage and single-stage object detectors. Several research works were carried out using two-stage networks. Ho Kwan Leung et al. suggested an optimized Faster-RCNN-based approach to detect vehicles in the dark. Suitable hyperparameters were chosen to optimize the training process. Their approach achieved a mAP of approximately 0.85. Besides, their study also highlighted solutions for collecting and labeling data. In another work, Galarza-Bravo et al. [37] implemented an improved Faster-RCNN to detect pedestrians from far infrared images efficiently. Their approach involved Faster-RCNN-based multi-scale detection using two region proposal networks. They observed an improvement in mAP of approximately 85%. However, the proposed approach's computation time was significantly higher than the considered baselines. Further, Dai et al. [38] presented a multi-task Faster RCNN framework to detect pedestrians and estimate distances using near-infrared images. The proposed methodology achieved a detection accuracy of nearly 80% and an inference speed of 7 fps. Further, less than 5% error was recorded while calculating the total average absolute distance estimation error.

Along similar lines, scholars have developed one-stage detectors due to their lower computation time [39]. Huang et al. [40] suggested a vehicle detection approach by integrating the Mobilenetv2 and YOLOv3 models. Their detection model involved using Mobilenetv2 as the backbone network. Besides, they have used the K-means algorithm to obtain anchor boxes, optimizing the model using the EIoU loss function. Their results achieved an AP of 94.96% with a rate of 10 fps which performed better than baseline YOLOv3. Devi et al. [41] adopted a YOLOv5-PANet architecture to detect pedestrians at night. Their proposed approach relied on using a transformer module and attention mechanism to improve the feature extraction capabilities. Their methodology observed an improved detection rate over existing approaches, and they have used network pruning to ensure its suitability for lower memory requirements.

Various enhancement techniques have been used in conjunction with object detectors to boost the overall detection performance. Murugan et al. [42] suggested a YOLOv4-based approach. They have used low-pass and unsharp filters as preprocessing steps to reduce noise and enhance image sharpness. The proposed framework achieved a detection rate of 0.95 mAP at 79 fps. Shao et al. [43] provided a feature enhancement network based on CycleGAN to enhance the vehicle's features and improve vehicle detection. They combined it with Faster R-CNN to achieve a high detection rate of 97.8%. In the domain of disaster management, Kao et al. [44] formulated a YOLOv4-based approach for identifying cracks in bridges with complex backgrounds and poor lighting. Their framework achieved a detection accuracy of 92%. In another study, Prabhu et al. [45] proposed RescueNet, a YOLO-based model for detecting and counting objects in flood-affected areas. Their approach was able to achieve a mAP of 98% and an F1-score of 94%.

Most approaches faced challenges while improving the detection rate with lower memory requirements. This study suggests an optimized Zero-DCE++-based YOLOv5 framework to perform detection in extremely low-light environments. The proposed approach applied in the domain of road object detection at night with varying degrees of illumination conditions achieved a significant improvement in detection accuracy and in the lowering of the computation requirements.

3 Proposed Methodology

A detailed overview of all the steps for efficient detection is presented in Fig. 1. The work utilizes the YOLOv5s architecture to perform detection. Once the images are processed through the Zero-DCE++ enhancement network, the improved YOLOV5s framework is used for the detection task.

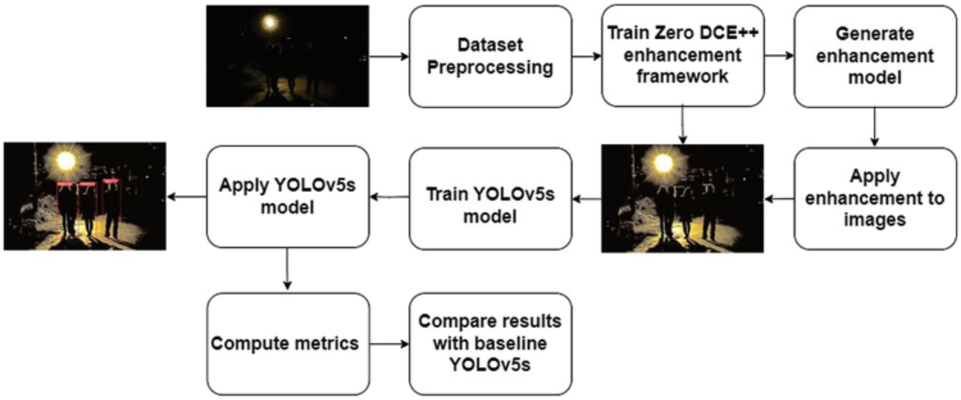
Dataset Preprocessing
Train Zero DCE++ enhancement framework
Generate enhancement model
Apply enhancement to images
Train YOLOv5s model
Apply YOLOv5s model
Compute metrics
Compare results with baseline YOLOv5s

Figure 1: Workflow of proposed methodology

3.1 Data Collection

This work utilizes the NOD and ExDARK [46,47] dataset. For NOD, high-quality street scenes were captured in the evening to night hours using 2 DSLR cameras: Nikon D750 and Sony RX100 VII. The dataset includes a total of 7.2 k annotated images. The ExDARK dataset consists of 7.3 k annotated images taken in 10 different lighting conditions and has 12 object classes. Fig. 2 displays the different images from both datasets with varying degrees of illumination. For this study, three class labels, namely: Bicycle, Person, and Car are considered.

(a) (b)

Figure 2: Sample dataset images in (a) NOD dataset (b) Exdark dataset

3.2 Dataset Preprocessing

Initial preprocessing involves resizing the images. Further, the bounding box annotations are converted to the YOLO bounding box format for processing in YOLOv5. Besides, to train the Zero-DCE++ enhancement framework and YOLOv5 architecture, the dataset is split into two parts, each consisting of validation and training sets for the enhancement and detection framework for both datasets.

3.3 Zero-DCE++ Enhancement Framework

To boost detection efficacy, the work integrates the Zero-DCE++ enhancement approach with YOLOv5. The purpose of choosing Zero-DCE++ over GAN and CNN-based approaches is the advantage of zero-reference, as proposed by Chongyi et al. [48], i.e., the image dataset need not contain paired or unpaired image data to supplement its training process. Additionally, DCE-net makes it computationally efficient due to the lower number of parameters, which makes it an ideal candidate for utilization in real-time environments [49]. Fig. 3 displays the architecture of Zero-DCE++. The approach considers a dimly lit input image and learns the mapping curve to provide a brightly illuminated image via a Deep Curve Estimation network (DCE-Net). Further, the mapping curve is utilized for adjusting the dynamic pixels' range of the original image Red, Green, and Blue (RGB) channels iteratively to obtain an enhanced final version of the image [50]. The improved image's dynamic range and the surrounding pixels' contrast are preserved while best-fitting curves are approximated (curve parameter maps).

3.3.1 Light Enhancement Curves

In this approach, the parameters of the Light Enhancement (LE) curve depend entirely upon the input image to understand the mapping between the poorly lit image and its enhanced version. Every individual pixel of the original image will receive its corresponding enhancement curve. Eq. (1) highlights the quadratic curve expression to achieve the designed image enhancement.

$$\boldsymbol{LE_{iter(n)}(s) = LE_{iter(n-1)}(s) + P(s)\, LE_{iter(n-1)}(s)(1 - LE_{iter(n-1)}(s))} \tag{1}$$

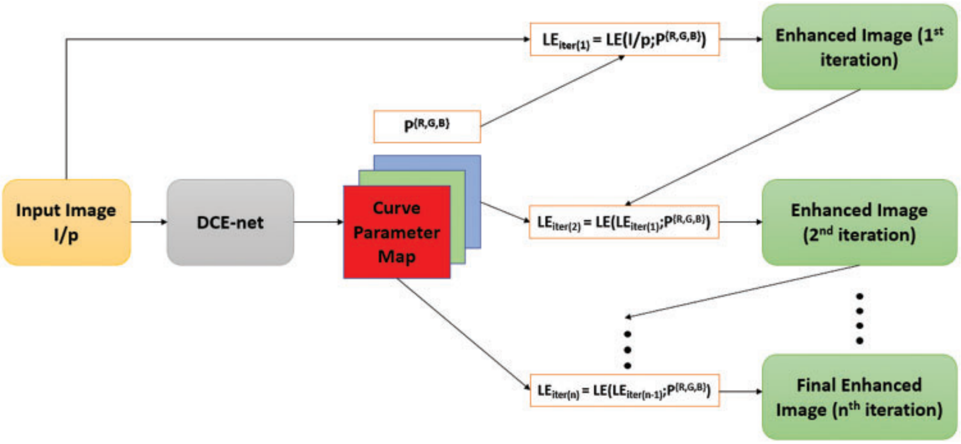
Input Image
I/p
DCE-net
Curve
Parameter
Map
p(R,G,B)
LEiter(1) = LE(I/p;P(R,G,B))
LEiter(2) = LE(LEiter(1);P(R,G,B))
LEiter(n) = LE(LEiter(n-1);P(R,G,B))
Enhanced Image (1st iteration)
Enhanced Image (2nd iteration)
Final Enhanced Image (nth iteration)

Figure 3: Diagrammatic representation of Zero-DCE++ enhancement framework

3.3.2 DCE-Net

DCE-Net is used in the Zero-DCE++ framework to understand the association between the input and its corresponding best-fitting parameter maps. The low light input image is downsampled by factor 12 in Zero-DCE++, which is presented to DCE-Net as input. Besides, Zero-DCE++ uses depthwise separable convolution [51] instead of the traditional CNN layers (as in Zero-DCE) to reduce network parameters. DCE-Net comprises seven DWConv layers. For the first six layers, 32 convolutional layers are present in each layer with ReLU activation, and the seventh layer uses tanh activation. Further, the kernel size of the DSConv layer is 3 × 3 and stride 1, and the kernel size of 1 × 1 and stride of 1 for pointwise convolution. Finally, the curve parameter maps are resized back to the original input size.

3.3.3 Non-Reference Loss Functions

DCE-Net is trained based on four custom-defined differentiable loss functions to analyze the enhanced image quality. Exposure Control loss controls exposure levels and prevents over/under-exposed regions in the image. Eq. (2) represents the exposure control loss $L_{exposure}$ as:

$$L_{exposure} = \frac{1}{M}\sum_{k=1}^{M} |Y_K - E_{level}|, \tag{2}$$

where E_{level} is the optimal exposure level, M is the number of non-overlapping local areas of size 16 × 16, and Y denotes the local regions' average intensity value in the improved image.

The spatial consistency loss $L_{spatial}$ preserves the difference between the improved image and between the surrounding areas of the given input image to retain the enhanced image's spatial consistency.

$$L_{spatial} = \frac{1}{T}\sum_{x=1}^{M}\sum_{y\in\Omega(x)} (|E_x - E_y| - |O_x - O_y|)^2, \tag{3}$$

where $\Omega\,(x)$ is the adjacent regions in all four directions and T is the number of local regions. Further, E and O are the improved and low-lit input images, and the value of the local region's size is fixed as 4 × 4.

Further, the monotonous values between the nearby pixels are retained using illumination smoothness loss $L_{illumination}$, expressed as:

$$L_{illumination} = \frac{1}{N}\sum_{i=1}^{N}\sum_{c\in\epsilon}\left(\left|\nabla_x P_n^c\right| + \left|\nabla_y P_n^c\right|\right)^2, \epsilon = \{R, G, B\}, \tag{4}$$

where ∇_y and ∇_x are the vertical and horizontal gradient operations and N is the number of iterations.

The reduction of deviations in the color of enhanced images is denoted as color constancy loss L_{color}.

$$L_{color} = \sum_{\forall(p,q)\in\epsilon}^{N} \left(T^p - T^q\right)^2, \epsilon = \{(R, G), (G, B)\ (R, B)\}, \tag{5}$$

where the pixel average value is T^p for the p channel in the improved image, and the pair of channels is (p, q).

To sum up, the total value of loss function L_{total} can be expressed as:

$$L_{total} = L_{exposure} + L_{spatial} + W_{illumination}L_{illumination} + W_{color}L_{color}, \tag{6}$$

where the values of $W_{illumination}$ & W_{color} denote the weights for the losses.

3.4 YOLOv5 Model

YOLOv5 was proposed by Glenn Jocher in 2020 [52]. Different versions of the YOLOv5 models were proposed with varying network depth and width. They are denoted with letters n, s, m, l, and x represented in an increasing order of model parameters [53]. The task of detection in YOLOv5 is carried out as a regression problem by using a single neural network for making predictions of bounding boxes and their subsequent classes. Initially, basic preprocessing is applied to the images, such as mosaic augmentation to accurately detect small objects with adaptive anchor boxes and adaptive image scaling.

YOLOv5 comprises three sections. Starting with the backbone, it has a Spatial Pyramid Pooling Fast (SPPF) module, C3, and Conv modules. The conv modules perform the basic 2D convolution, regularization, and activation operations on the input, which supplements the feature extraction process in the C3 module [54]. C3 derives its concept from the CSPNet structure [55]. It includes two distinct branches the first branch includes series-wise connection of n Bottleneck components, whereas the second includes a Conv layer. The feature extraction capability is improved by splicing two branches. Furthermore, SPPF efficiently extracts global information about the target. It improves feature extraction ability and achieves a fusion of features with varied scales by integrating a set of fixed block pooling operations.

The second part of the YOLOv5 architecture is the Neck module which includes a combination of Path Aggregation Network (PANet) [56] and FPN. The FPN facilitates the generation of feature pyramids to detect objects with varying sizes and improves the generalization of the model. The PANet structure enables the shifting of stronger localization features from feature maps of the bottom to the top layers.

The head layer in the YOLOv5 detection framework applies anchors on the features and produces the final output vectors with objectness scores, class-based probabilities, and bounding boxes. It comprises three different detection layers of differing feature map sizes to aid the detection of varying object sizes.

3.5 Improved YOLOv5 Model

Fig. 4 depicts the changes made to the existing YOLOv5 network. Firstly, all C3 structures in the backbone are replaced by CBAM modules (C3CBAM) to enhance the model's ability to detect the features. However, the addition of CBAM modules contributes to increasing complexity by using more parameters to make the model lightweight, the Conv modules in the neck of YOLOv5 are substituted with DWConv modules which optimize the model parameters while still maintaining the feature extraction ability and thus, the required detection accuracy.

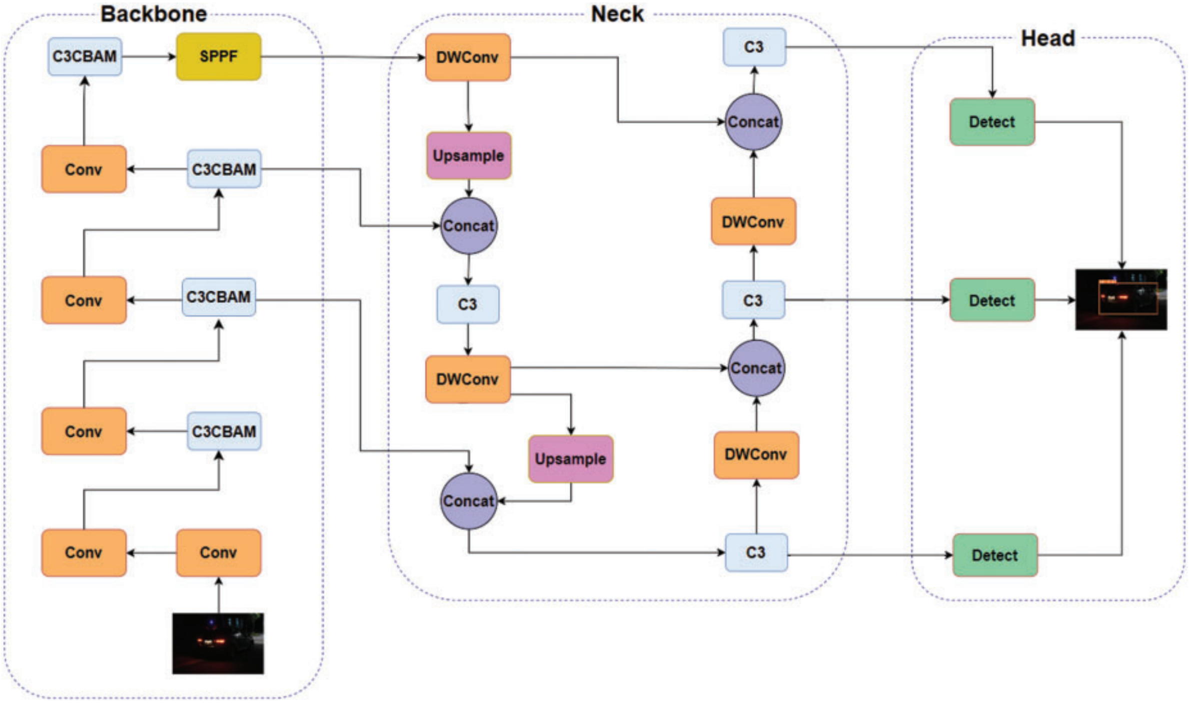

Figure 4: Architecture of improved YOLOv5 network

3.5.1 Convolutional Block Attention Module (CBAM)

Detection of smaller targets in the dark is challenging as they occupy fewer pixels, and their feature information is effectively lost, leading to false or missed detections [57]. CBAM [58] combines Channel Attention and Spatial Attention modules. The attention module uses a feedforward convolutional neural network. Fig. 5 explains the detailed architecture of CBAM. It could be observed that a single-dimensional channel attention map M_c of size C × 1 × 1 and two-dimensional spatial attention map M_s of size 1 × H × W is determined sequentially by CBAM, which could be arranged in parallel or sequence [59].

$$F\_Map' = M_C\left(F\_Map\right) \otimes F\_Map, \tag{7}$$

$$F\_Map'' = M_S\left(F_{Map}\right) \otimes F\_Map', \tag{8}$$

where F_Map denotes the feature map, F_Map'' is the final refinement output and ⊗ signifies element multiplication.

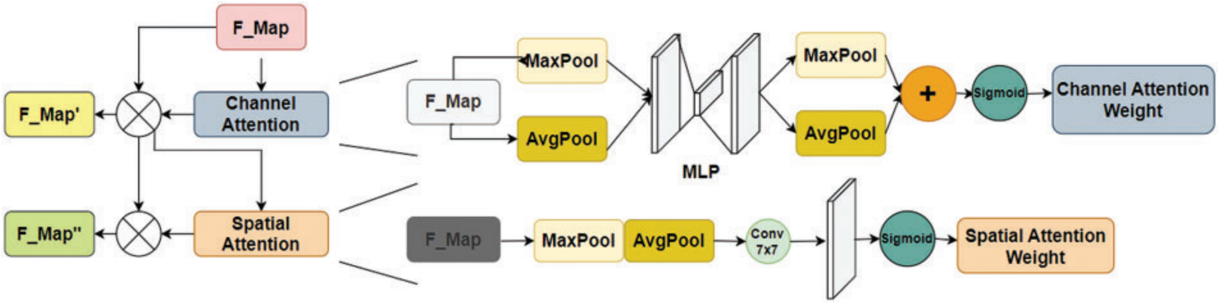

Figure 5: Network structure of CBAM

The channel attention module boosts the ability of the detector to extract channel information using max pooling, which accumulates details on the unique features of an object, and average, pooling to gather spatial information. Subsequently, the extracted information is filtered, activated, and normalized.

$$M_C\left(F\_Map\right) = \sigma\left(MLP\left(MaxPool\left(F\_Map\right)\right) + MLP\left(AvgPool\left(F\_Map\right)\right)\right) \tag{9}$$

where σ is the sigmoid function. $MaxPool\left(F_{Map}\right)$ and $AvgPool\left(F_{Map}\right)$ are the max pooled and average pooled features, respectively.

Channel attention map $M_C\left(F\_Map\right)$ is produced by forwarding both the descriptors to a shared network which has a multi-layer perceptron with a hidden layer. Both the output feature vectors are combined after applying the shared network to each descriptor using the element-wise addition operation. F_Map' is obtained from the element-wise multiplication of $M_C\left(F\_Map\right)$ and F_Map.

The spatial attention module concentrates on local information. It uses max pooling and average pooling operations to pool the channel axis. Further, it is concatenated to generate an effective feature descriptor. Spatial attention map $M_S\left(F\_Map\right)$ is generated by applying a convolutional layer, that encodes the positions thatare required to be suppressed or attended.

$$M_S\left(F\_Map\right) = \sigma\left(f^{7\times7}\left(\left[MaxPool\left(F\_Map\right); AvgPool\left(F\_Map\right)\right]\right)\right) \tag{10}$$

where $f^{7\times7}$ denotes the convolution operation with the filter of size 7 × 7.

3.5.2 Depthwise Separable Convolution Module (DWConv)

DWConv [60] was introduced in 2017 for integration into embedded devices and mobile applications. It reduced the total number of parameters, which enables higher efficiency during the operation of convolution by doing a split in the standard convolution in the channel and spatial dimensions. The key feature of the module is that it is lightweight and thus included as a core unit in the MobileNet structure. As seen in Fig. 6, the architecture of DWConv is segregated into pointwise and depthwise convolution. Only an individual convolution kernel is responsible for a single channel in the case of depthwise convolution, i.e., each convolution kernel could only access a single channel [61]. Besides, the pointwise convolution process uses a kernel of size 1 × 1 is weighed in a single direction associated with the preceding map depth to develop a newer feature map. Eqs. (11) and (12) highlight the computational complexities of normal convolution $C_{Depthwiseconvolution}$ and $C_{Convolution}$, respectively. The ratio between the computational cost of $C_{DepthwiseConvolution}$ and computational cost of $C_{convolution}$ is presented in the Eq. (13). Experimental results demonstrate the amount of computation of the DWConv is lower than the normal convolution operation by 8–9 time while considering the number of convolutions in the DWConv module as 3 × 3.

$$C_{DepthwiseConvolution} = D_{out1}.D_{out2}.D_{k1}.D_{k2}.C_{in} + D_{out1}.D_{out2}.C_{out}.C_{in}, \tag{11}$$

$$C_{convolution} = D_{out1}.D_{out2}.D_{k1}.D_{k2}.C_{out}.C_{in}, \tag{12}$$

$$\frac{C_{DepthwiseConvolution}}{C_{convolution}} = \frac{D_{out1}.D_{out2}.D_{k1}.D_{k2}.C_{in} + D_{out1}.D_{outt}.C_{out}.C_{in}}{D_{outl}.D_{out2}.D_{k1}.D_{k2}.C_{out}.C_{in}}, \tag{13}$$

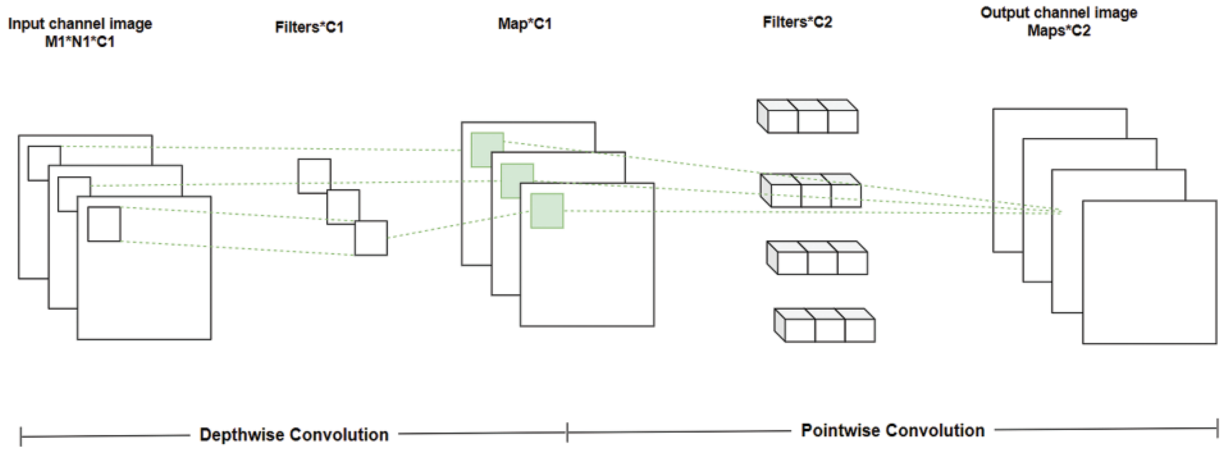

Figure 6: Network structure of DWConv

4 Experimental Setup

The proposed work was implemented using Python 3.9.16, Pytorch 1.13.1, and Cuda 11.6 on Google Colab. The Colab environment was allocated with a Tesla T4 GPU and 15102 MiB memory. The parameter settings used in the proposed Zero-DCE++ approach are like the ones provided in the original paper. The images are resized to 512 × 512 × 3 during the training phase. However, the inference is performed on the images of size 720 × 480 × 3. Furthermore, the parameter settings include the following: the batch size of 16, the optimizer as Adam, the learning rate of 1e^{-4}, and epochs are set to 100. The standard deviation Gaussian function is set to 0.02, along with the standard zero means for the filter weights of every layer. Besides, the values of the weights $W_{illumination}$ & W_{color} are set to 20 and 0.5, respectively. The YOLOv5 network parameters settings are initialized to run the YOLOv5s model. The image size parameter is fixed as 720. Other training parameters include the following: the batch size of 16, SGD optimizer, learning rate of 0.01, and epochs of 100. The dataset distribution is presented in Table 1.

Table 1: Dataset split for Zero-DCE++ and YOLOv5

Dataset	Models	Total	Training	Validation
NOD	Zero-DCE++ and YOLOv5	3000	2400	600
ExDark	Zero-DCE++ and YOLOv5	900	720	180

5 Experimental Results and Discussion

The proposed implementation incorporates Zero-DCE++ over the Zero-DCE framework as the number of parameters is lowered further. Fig. 7 shows the benefit of using the enhancement of Zero-DCE++. Fig. 8 shows the target objects of variable sizes which are detected in different scenarios of lighting. It could be observed that objects which have less visibility are also detected accurately.

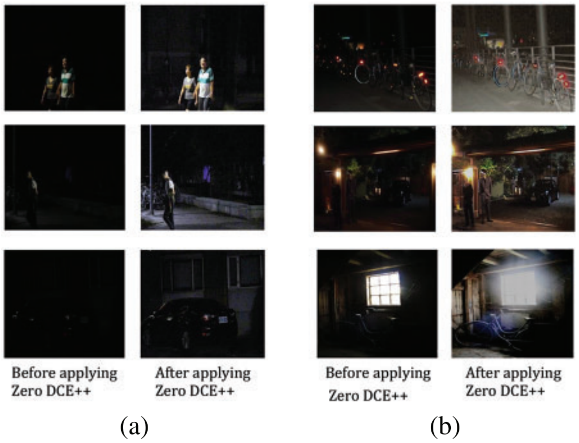
Before applying Zero DCE++
After applying Zero DCE++
(a)
Before applying Zero DCE++
After applying Zero DCE++
(b)

Figure 7: Images before and after applying Zero-DCE++ framework in (a) NOD (b) Exdark dataset

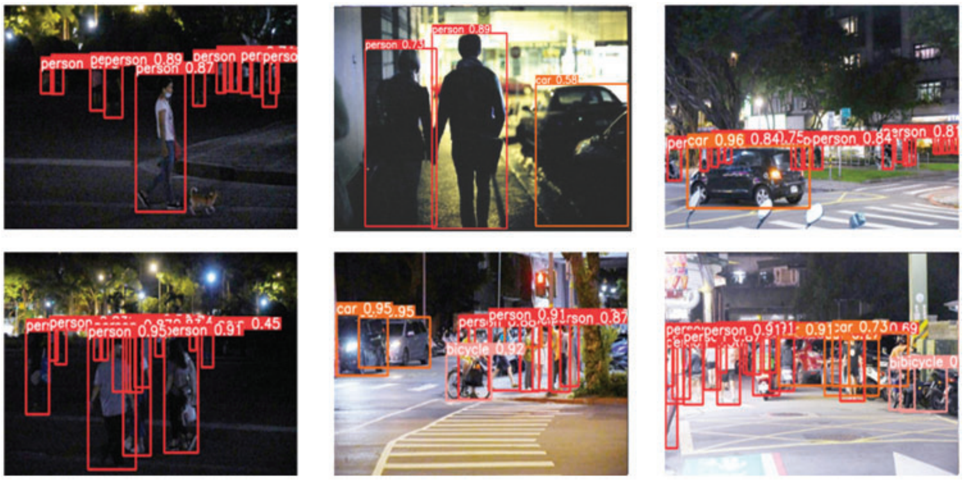

Figure 8: Results for identification of target objects

5.1 Evaluation Metrics

This work leverages various metrics to evaluate the efficacy of the proposed approach for performing efficient detection of objects in challenging nighttime lighting conditions. The metrics include $Precision$, $Recall$, $F1-score$, AP and mAP scores. The value of $Precision$ refers to the fraction of the accurately classified positive instances to the total positively classified instances. The formula for precision is:

$$Precision = \frac{TP}{TP + FP}, \tag{14}$$

where TP refers to the True Positives and false positives are denoted by FP.

The value *Recall* denotes the fraction of the accurately classified positive instances to the actual number of objects present in each image. Eq. (15) highlights the formula for recall:

$$Recall = \frac{TP}{TP + FN}, \tag{15}$$

where FN refers to the False Negatives. The weighted average of recall and precision gives the F1 score. Eq. (16) denotes the formula for F1-score.

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall}, \tag{16}$$

The average precision (AP) metric is calculated for all the classes in this work. It measures the precision of all the values of recall that lie between 0 to 1. The formula can be observed in Eq. (17). On similar lines, the mean average precision (mAP) metric is calculated by the mean of the AP values for all the class labels. Eq. (18) signifies the formula for mAP:

$$AP = \int_0^1 Precision\,(i)\,di, \tag{17}$$

$$mAP = \frac{\sum_{i=1}^{C} AP(i)}{C}, \tag{18}$$

where C denotes the total number of class labels. Furthermore, the model complexity is estimated by the following metrics: *model size* and *total parameter count*.

5.2 Ablation Analysis

To justify the efficacy of the proposed approach over the baseline YOLOv5 detection approach, an ablation analysis is conducted. Table 2 highlights the results of the addition of every individual module in the study.

Table 2: Model ablation analysis

YOLOv5	Zero-DCE++	CBAM	DWConv	mAP$_{\text{NOD}}$	mAP$_{\text{ExDark}}$	Params(M)
✓				0.626	0.655	7.027
✓	✓			0.662	0.678	7.038
✓	✓	✓		0.683	0.732	7.054
✓	✓	✓	✓	0.679	0.724	6.157

In the ablation results, it could be observed that there is an increase in mAP from the baseline YOLOv5 to Zero-DCE++ + YOLOv5. An additional increase in mAP is observed while adding the CBAM attention module. However, this resulted in increased model parameters. Thus, the last model involves the usage of the DWConv module, which reduces parameters greatly without compromising much on the overall mAP, thereby providing an optimized detection approach.

5.3 Result and Discussion

The detection framework's accuracy is depicted in the form of loss functions and mAP curves. Fig. 9 displays the curves for the loss functions in YOLOv5. The figure shows that the loss curves of the model converge to smaller values of loss when trained on 100 epochs. It shows that mAP of 0.679 and obtained the highest mAP of 0.68 at the 96th epoch during the training phase. Fig. 10 demonstrates the Precision-Recall (PR) curve of the suggested model. From the figure, it could be observed that the curve for the classes 'person' & 'car' is higher than that of the class 'bicycle', which is suggestive of the better performance of the former as compared to the latter.

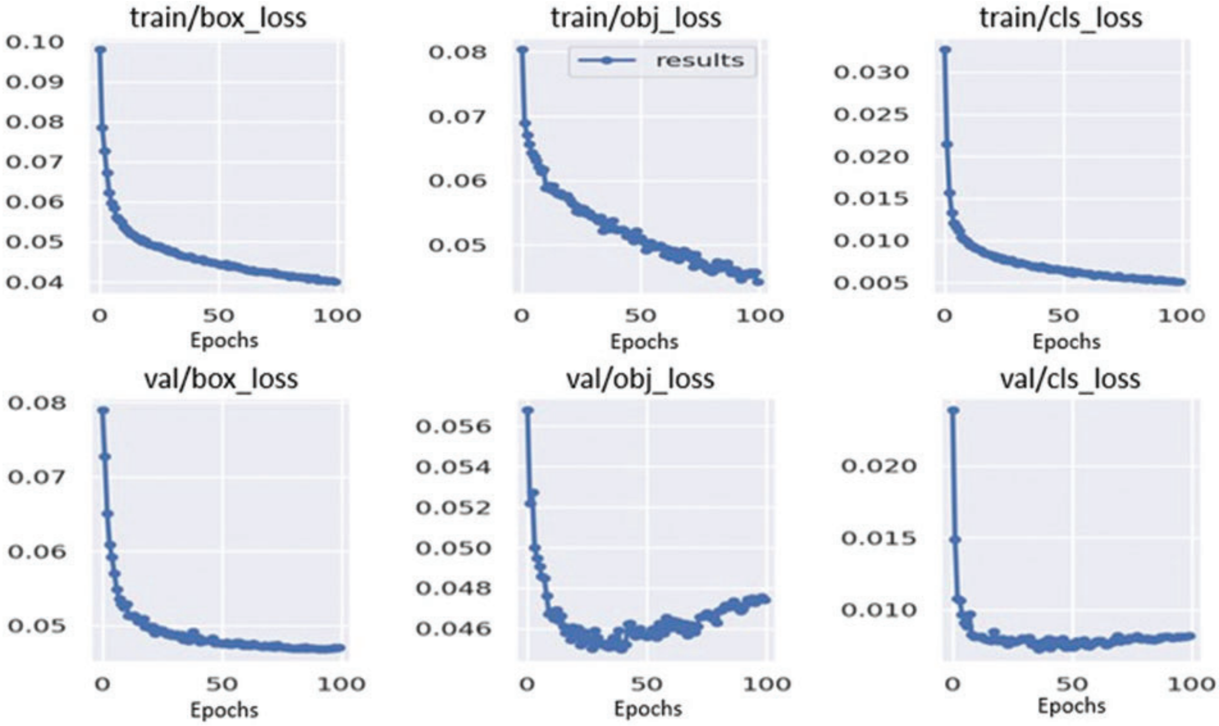

Figure 9: Curves for training and validation loss for the NOD dataset

Table 3 provides a detailed description of different object detection-related metrics, and Table 4 describes the average precision values of all the class labels. As inferred from the table below, the proposed framework had the highest precision value among all the potential models. Besides, the proposed model had the second-best mAP, F1, and recall values after the third model (YOLOv5 + Zero-DCE++ + CBAM). In general, it is observed that the AP of the classes car and person is observed to possess the best and the next highest mAP values, which points towards better detection accuracy of these classes.

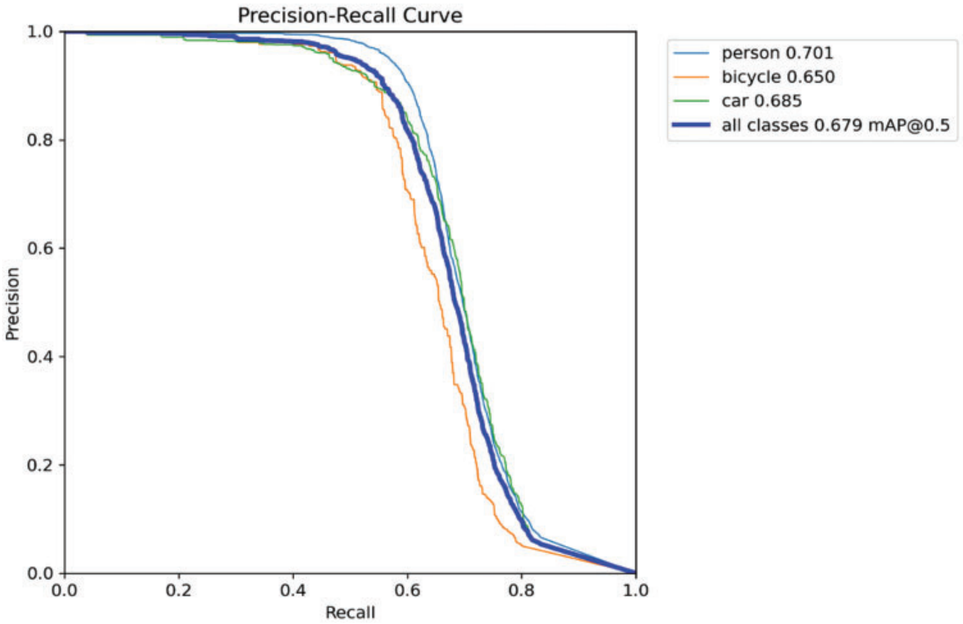

Figure 10: PR curve for the proposed approach for the NOD dataset

Table 3: Values of detection metrics

Model	Precision$_{NOD}$	Recall$_{NOD}$	F1$_{NOD}$	Precision$_{ExDARK}$	Recall$_{ExDARK}$	F1$_{ExDARK}$
YOLOv5	0.813	0.554	0.659	0.824	0.642	0.721
YOLOv5 + Zero-DCE++	0.841	0.577	0.684	0.836	0.653	0.733
YOLOv5 + Zero-DCE++ + CBAM	0.863	0.599	0.707	0.882	0.693	0.776
YOLOv5 + Zero-DCE++ + CBAM + DWConv	0.868	0.586	0.699	0.891	0.679	0.771

Table 4: Values of average precision for all the individual classes

Model	Person$_{NOD}$	Bicycle$_{NOD}$	Car$_{NOD}$	Person$_{ExDARK}$	Bicycle$_{ExDARK}$	Car$_{ExDARK}$
YOLOv5	0.645	0.607	0.638	0.684	0.647	0.641
YOLOv5 + Zero-DCE	0.689	0.648	0.672	0.702	0.664	0.683
YOLOv5 + Zero-DCE++ + CBAM	0.708	0.647	0.694	0.764	0.692	0.711
YOLOv5 + Zero-DCE++ + CBAM + DWConv	0.701	0.65	0.685	0.731	0.693	0.719

The computational complexity is presented in Table 5. Table 6 performs a comparison of mAP values with the other recent work [62]. Regardless of the high mAP observed in the third model, the proposed approach is observed to have the least number of parameters among all the considered models. Besides, the model size is also smaller than all other models, and a detection rate of 59 fps is observed, indicating the approach's feasibility for several applications with low memory and high-speed requirements.

Table 5: Model complexity of all the models

Model	Params (M)	Model Size (MB)
YOLOv5	7.027	13.7
YOLOv5 + Zero-DCE++	7.038	13.76
YOLOv5 + Zero-DCE++ + CBAM	7.054	13.86
YOLOv5 + Zero-DCE++ + CBAM + DWConv	6.157	12.16

Table 6: Comparison of mAP with contemporary work over ExDark dataset

Model	mAP
[62]	71.9
YOLOv5 + Zero-DCE++ + CBAM + DWConv	72.4

6 Conclusion and Future Work

The context of detection of objects in varying lighting scenarios is quite significant for a multitude of scenarios like surveillance applications and autonomous driving applications. Several existing approaches perform efficient detection for daytime conditions. However, detecting objects in night-time conditions is still an open challenge. Besides, most of the approaches tend to have higher model parameters which, in turn, contribute towards the model complexity and make the framework memory intensive. In this work, an optimized YOLOv5 structure that integrates the low light image enhancement capabilities of Zero-DCE++ architecture with the YOLOv5 detection framework to achieve a higher detection accuracy compared to baseline YOLOv5 has been implemented. This work makes use of YOLOv5s architecture. Moreover, the architecture is further optimized by modifying the existing YOLOv5 framework. Firstly, the existing C3 modules in the backbone network were replaced with CBAM blocks, further increasing the mAP. However, adding the Zero-DCE++ framework and CBAM modules also increased the total model parameters. Thus, to optimize the model complexity, the Conv module in the neck network was substituted with the DWConv module. Finally, the proposed architecture was evaluated against the baseline YOLOv5 architecture. It achieved an mAP of approximately 67.9% on NOD dataset and 72.4% on ExDARK dataset. Thus, the results indicate the benefit of the proposed approach by observing an increase in detection accuracy and the lowering of the model complexity.

Future work would involve testing several image-denoising mechanisms to improve the model performance. It was observed that Zero-DCE++ successfully enhanced the exceptionally poorly lit sections of an image quite effectively. However, during the enhancement phase, it also added some noise to the images, which could have influenced the mAP values. Therefore, the effect of noise on the detection performance will be considered. Additionally, the focus would be on building a custom low-light image dataset with more classes and using the proposed approach to fulfill the necessities of different use cases requiring nighttime object detection.

Acknowledgement: The authors wish to express their thanks to VIT management for their extensive support during this work.

Funding Statement: The authors received no specific funding for this work.

Author Contributions: The authors confirm their contribution to the paper as follows: study conception and design: Ananthakrishnan Balasundaram and Anshuman Mohanty; data collection: Ayesha Shaik; analysis and interpretation of results: Ananthakrishnan Balasundaram, Anshuman Mohanty, Ayesha Shaik, Krishnadoss Pradeep, Kedalu Poornachary Vijayakumar and Muthu Subash Kavitha; draft manuscript preparation: Ananthakrishnan Balasundaram, Anshuman Mohanty, Ayesha Shaik, Krishnadoss Pradeep, Kedalu Poornachary Vijayakumar and Muthu Subash Kavitha. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Data openly available in a public repository. NOD Dataset is available at https://opendatalab.com/NOD and ExDark dataset is available at https://www.kaggle.com/datasets/washingtongold/exdark-dataset.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] H. K. Leung, X. Z. Chen, C. W. Yu, Liang, J. Y. Wu *et al.,* "A deep-learning-based vehicle detection approach for insufficient and nighttime illumination conditions," *Applied Sciences*, vol. 9, no. 22, pp. 4769, 2019.

[2] Z. Zou, K. Chen, Z. Shi, Y. Guo and J. Ye, "Object detection in 20 years: A survey," *Proceedings of the IEEE*, vol. 111, no. 3, pp. 257–276, 2023.

[3] Y. Amit, P. Felzenszwalb and R. Girshick, "Object detection, a reference guide," *Computer Vision*, vol. 3, no. 1, pp. 1–9, 2020.

[4] J. Latif, C. Xiao, A. Imran and S. Tu, "Medical imaging using machine learning and deep learning algorithms: A review," in *Proc. of 2nd Int. Conf. on Computing, Mathematics and Engineering Technologies (ICOMET)*, Sukkur, Pakistan, pp. 1–5, 2019.

[5] Q. Zhang, Y. Liu, C. Gong, Y. Chen and H. Yu, "Applications of deep learning for dense scenes analysis in agriculture: A review," *Sensors*, vol. 20, no. 5, pp. 1520, 2020.

[6] K. K. Santhosh, D. P. Dogra and P. P. Roy, "Anomaly detection in road traffic using visual surveillance: A survey," *ACM Computing Surveys*, vol. 53, no. 6, pp. 1–26, 2020.

[7] M. Everingham, L. Van Gool, C. K. Williams, J. Winn and A. Zisserman, "The pascal visual object classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, no. 1, pp. 303–308, 2009.

[8] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona *et al.,* "Microsoft COCO: Common objects in context," in *Proc. of 13th European Conf. on Computer Vision (ECCV2014)*, Zurich, Switzerland, pp. 6–12, 2014.

[9] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh *et al.,* "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 1, pp. 211–252, 2015.

[10] C. Galera-Zarco and G. Floros, "A deep learning approach to improve built asset operations and disaster management in critical events: An integrative simulation model for quicker decision making," *Annals of Operations Research*, vol. 1, no. 1, pp. 1–40, 2023.

[11] G. Gursoy, A. Varol and A. Nasab, “Importance of machine learning and deep learning algorithms in earthquake prediction: A review,” in *Proc. of 11th Int. Symp. on Digital Forensics and Security (ISDFS)*, Chattanooga, TN, USA, pp. 1–6, 2023.

[12] M. A. Islam, S. I. Rashid, N. U. I. Hossain, R. Fleming and A. Sokolov, “An integrated convolutional neural network and sorting algorithm for image classification for efficient flood disaster management,” *Decision Analytics Journal*, vol. 7, no. 1, pp. 1–13, 2023.

[13] G. Seeja, A. Selvakumar and V. B. Hency, “A novel approach for disaster victim detection under debris environments using decision tree algorithms with deep learning features,” *IEEE Access*, vol. 11, no. 1, pp. 54760–54772, 2023.

[14] A. Berroukham, K. Housni, M. Lahraichi and I. Boulfrifi, “Deep learning-based methods for anomaly detection in video surveillance: A review,” *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 1, pp. 314–327, 2023.

[15] M. A. Morid, O. R. L. Sheng and J. Dunbar, “Time series prediction using deep learning methods in healthcare,” *ACM Transactions on Management Information Systems*, vol. 14, no. 1, pp. 1–29, 2023.

[16] P. Felzenszwalb, D. McAllester and D. Ramanan, “A discriminatively trained, multiscale, deformable part model,” in *Proc. of Computer Vision and Pattern Recognition (CVPR2008)*, Anchorage, AK, USA, pp. 1–8, 2008.

[17] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Proc. of Computer Vision and Pattern Recognition (CVPR2005)*, San Diego, USA, pp. 886–893, 2005.

[18] P. Viola and M. J. Jones, “Robust real-time face detection,” *International Journal of Computer Vision*, vol. 57, pp. 137–154, 2004.

[19] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed *et al.,* “SSD: Single shot multibox detector,” in *Proc. of European Conf. on Computer Vision (ECCV2016)*, Amsterdam, The Netherlands, pp. 11–14, 2016.

[20] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proc. of Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, USA, pp. 779–788, 2016.

[21] T. Y. Lin, P. Goyal, R. Girshick, K. He and P. Dollar, “Focal loss for dense object detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 2980–2988, 2017.

[22] P. Soviany and R. T. Ionescu, “Optimizing the trade-off between single-stage and two-stage deep object detectors using image difficulty prediction,” in *Proc. of Int. Symp. on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, Timisoara, Romania, pp. 209–214, 2018.

[23] T. Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan *et al.,* “Feature pyramid networks for object detection,” in *Proc. of Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, pp. 2117–2125, 2017.

[24] R. Girshick, “Fast R-CNN,” in *Proc. of ICCV*, Santiago, Chile, pp. 1440–1448, 2015.

[25] S. Ren, K. He, R. Girshick and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016.

[26] K. He, G. Gkioxari, P. Dollár and R. Girshick, “Mask R-CNN,” in *Proc. of Computer Vision and Pattern Recognition (CVPR2018)*, Salt Lake City, Utah, pp. 2961–2969, 2018.

[27] P. Kumar, S. Batchu and S. R. Kota, “Real-time concrete damage detection using deep learning for high rise structures,” *IEEE Access*, vol. 9, pp. 112312–112331, 2021.

[28] R. Vishwakarma and R. Vennelakanti, “CNN model & tuning for global road damage detection,” in *Proc. of Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, pp. 5609–5615, 2020.

[29] X. Wang, J. Ouyang, D. Li and G. Zhang, “Underwater object recognition based on deep encoding-decoding network,” *Journal of Ocean University of China*, vol. 18, no. 1, pp. 376–382, 2019.

[30] Y. Miao, F. Liu, T. Hou, L. Liu and Y. Liu, “A nighttime vehicle detection method based on YOLO v3,” in *Proc. of China Int. Agrochemical and Crop Protection (CAC)*, Shanghai, China, pp. 6617–6662, 2020.

[31] M. Haris and A. Glowacz, “Road object detection: A comparative study of deep learning-based algorithms,” *Electronics*, vol. 10, no. 16, pp. 1932, 2021.

[32] Y. Xiao, A. Jiang, J. Ye and M. W. Wang, “Making of night vision: Object detection under low-illumination,” *IEEE Access*, vol. 8, pp. 123075–123086, 2020.

[33] Y. Zhang, Z. Guo, J. Wu, Y. Tian, H. Tang *et al.,* “Real-time vehicle detection based on improved YOLO v5,” *Sustainability*, vol. 14, no. 19, pp. 12274, 2022.

[34] G. Lowe, “Sift-the scale invariant feature transform,” *International Journal of Computer Vision*, vol. 2, no. 2, pp. 91–110, 2004.

[35] Y. Munian, Y. A. Martinez-Molina and M. Alamaniotis, “Intelligent system for detection of wild animals using HOG and CNN in automobile applications,” in *Proc. of Int. Conf. on Information, Intelligence, Systems and Applications (IISA)*, Piraeus, Greece, pp. 1–8, 2020.

[36] J. Baek, S. Hong, J. Kim and E. Kim, “Efficient pedestrian detection at nighttime using a thermal camera,” *Sensors*, vol. 17, no. 8, pp. 1850, 2017.

[37] M. A. Galarza-Bravo and M. J. Flores-Calero, “Pedestrian detection at night based on faster R-CNN and far infrared images,” in *Proc. of Int. Conf. on Intelligent Robotics and Applications (ICIRA)*, Newcastle, NSW, Australia, pp. 335–345, 2018.

[38] X. Dai, J. Hu, H. Zhang, A. Shitu, C. Luo *et al.,* “Multi-task faster R-CNN for nighttime pedestrian detection and distance estimation,” *Infrared Physics & Technology*, vol. 115, no. 1, pp. 103694, 2021.

[39] A. Lohia, K. D. Kadam, R. R. Joshi and A. M. Bongale, “Bibliometric analysis of one-stage and two-stage object detection,” *Library Philosophy and Practice*, vol. 7, no. 3, pp. 10–21, 2021.

[40] S. Huang, Y. He and X. A. Chen, “M-YOLO: A nighttime vehicle detection method combining mobilenet v2 and YOLO v3,” in *Proc. of 2nd Int. Conf. on Computer Information and Big Data Applications (ICCIBA)*, Wuhan, China, pp. 012094, 2021.

[41] S. Devi, K. Thopalli, P. Malarvezhi and J. J. Thiagarajan, “Improving single-stage object detectors for nighttime pedestrian detection,” *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 36, no. 9, pp. 2250034, 2022.

[42] R. A. Murugan and B. Sathyabama, “Object detection for night surveillance using ssan dataset based modified YOLO algorithm in wireless communication,” *Wireless Personal Communications*, vol. 128, no. 3, pp. 1813–1826, 2023.

[43] X. Shao, C. Wei, Y. Shen and Z. Wang, “Feature enhancement based on CycleGAN for nighttime vehicle detection,” *IEEE Access*, vol. 9, no. 1, pp. 849–859, 2020.

[44] S. P. Kao, Y. C. Chang and F. L. Wang, “Combining the YOLOv4 deep learning model with UAV imagery processing technology in the extraction and quantization of cracks in bridges,” *Sensors*, vol. 23, no. 5, pp. 2572, 2022.

[45] B. B. Prabhu, R. Lakshmi, R. Ankitha, M. S. Prateeksha and N. C. Priya, “RescueNet: YOLO-based object detection model for detection and counting of flood survivors,” *Modeling Earth Systems and Environment*, vol. 8, no. 4, pp. 4509–4516, 2022.

[46] I. Morawski, Y. A. Chen, Y. S. Lin and W. H. Hsu, “Nod: Taking a closer look at detection under extreme low-light conditions with night object detection dataset,” arXiv preprint arXiv:2110.10364, 2021.

[47] Y. P. Loh and C. S. Chan, “Getting to know low-light images with the exclusively dark dataset,” *Computer Vision and Image Understanding*, vol. 178, no. 1, pp. 30–42, 2019.

[48] C. Li, C. Guo and C. C. Loy, “Learning to enhance low-light image via zero-reference deep curve estimation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 8, pp. 4225–4238, 2021.

[49] A. S. Parihar, S. Singhal, S. Nanduri and Y. Raghav, “A comparative analysis of deep learning based approaches for low-light image enhancement,” in *Proc. of Int. Conf. on Recent Advances and Innovations in Engineering (ICRAIE)*, Jaipur, India, pp. 1–5, 2020.

[50] W. Mu, H. Liu, W. Chen and Y. Wang, “A more effective zero-DCE variant: Zero-DCE tiny,” *Electronics*, vol. 11, no. 17, pp. 2750, 2022.

[51] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” in *Proc. of Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, pp. 1251–1258, 2017.

[52] S. Lyu, R. Li, Y. Zhao, Z. Li, R. Fan *et al.,* "Green citrus detection and counting in orchards based on YOLOv5-CS and AI edge system," *Sensors*, vol. 22, no. 2, pp. 576, 2022.

[53] Z. Xue, Z. R. Xu, D. Bai and H. Lin, "YOLO-Tea: A tea disease detection model improved by YOLOv5," *Forests*, vol. 14, no. 2, pp. 415, 2023.

[54] Z. Sun, P. Li, Q. Meng, Y. Sun and Y. Bi, "An improved YOLOv5 method to detect tailings ponds from high-resolution remote sensing images," *Remote Sensing*, vol. 15, no. 7, pp. 1796, 2023.

[55] C. Y. Wang, H. Y. M. Liao, Y. H. Wu, P. Y. Chen, J. W. Hsieh *et al.,* "CSPNet: A new backbone that can enhance learning capability of CNN," in *Proc. of Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, pp. 390–391, 2020.

[56] S. Liu, L. Qi, H. Qin, J. Shi and J. Jia, "Path aggregation network for instance segmentation," in *Proc. of Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, pp. 18–23, 2018.

[57] Y. Guo, S. E. Aggrey, X. Yang, A. Oladeinde, Y. Qiao *et al.,* "Detecting broiler chickens on litter floor with the YOLOv5-CBAM deep learning model," *Artificial Intelligence in Agriculture*, vol. 9, no. 1, pp. 36–45, 2023.

[58] S. Woo, J. Park, J. Y. Lee and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. of European Conf. on Computer Vision (ECCV)*, Munich, Germany, pp. 3–19, 2018.

[59] L. Zhu, X. Geng, Z. Li and C. Liu, "Improving YOLOv5 with attention mechanism for detecting boulders from planetary images," *Remote Sensing*, vol. 13, no. 18, pp. 3776, 2021.

[60] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang *et al.,* "MobileNets: Efficient convolutional neural networks for mobile vision applications," in *Proc. of Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, pp. 7–13, 2017.

[61] J. Tang, S. Liu, D. Zhao, L. Tang, W. Zou *et al.,* "An algorithm for real-time aluminum profile surface defects detection based on lightweight network structure," *Metals*, vol. 13, no. 3, pp. 507, 2023.

[62] J. Wang, P. Yang, Y. Liu, D. Shang, X. Hui *et al.,* "Research on improved YOLOv5 for low-light environment object detection," *Electronics*, vol. 12, no. 1, pp. 3089–3111, 2023.