PDE: A Real-Time Object Detection and Enhancing Model under Low Visibility Conditions

Zhiying Li[1], Shuyuan Lin[2]*, Zhongming Liang[3],
Yongjia Lei[4], Zefan Wang[5], Hao Chen[6]
Jinan University, Guangzhou, China[1,2,3,4,5]
The Hong Kong Polytechnic University, Hong Kong, China[6]

***Abstract*—Deep object detection models are important tools that can accurately detect objects and frame them for the user in real time. However, in low visibility conditions, such as fog or low light conditions, the captured images are underexposed and blurred, which negatively affects the recognition accuracy and is not well visible to humans. In addition, the image enhancement model is complex and time-consuming. Using the image enhancement model before the object recognition model cannot meet the real-time requirements. Therefore, we propose the Parallel Detection and Enhancement model (PDE), which detects objects and enhances poorly visible images in parallel and in real time. Specifically, we introduce the specially designed tiny prediction head along with coordinated attention and multi-stage concatenation modules to better detect underexposed and blurred objects. For the parallel image enhancement model, we adaptively develop improved weighting evaluation models for each "3D Lookup Table" module. As a result, PDE achieves better detection accuracy for poorly visible objects and more user-friendly reference in real time. Experimental results show that PDE has significantly better object recognition performance than the state-of-the-art on real foggy (8.9%) and low-light (20.6%) datasets.**

***Keywords*—Low-visibility condition; image enhance; object detection**

I. Introduction

Deep learning is used for many tasks, such as model fitting [1]–[3], object detection [4], [5], and so on. Recently, deep object detection models [6]–[8] have been widely used in daily life. These models provide accurate and 7×24 consistent object recognition, which facilitates people's work and helps them detect inconspicuous objects. However, in most places in the world, it is dark 42% of the day and there are one to three rainy or foggy days per week. These natural phenomena inevitably affect the performance of deep object detection models [9]–[11]. In addition, object detection models must filter out objects for user reference in real time (processing more than 30 frames per second). In this context, object detection in low visibility conditions has attracted much attention in both academia and industry to enable accurate and view-friendly object detection at all times of the day and under all climatic condition [12]–[14].

To mitigate the negative effects in low visibility conditions, current research can be divided into two classes: 1) Two-stage models: Two-stage models use image enhancement models to first enhance the images with poor visibility and then train the object detection model on the enhanced images [15], [16]. For example, [17] used "GridDehaze" to denoise foggy images, and [18] introduced a brightening step to lighten the images before object detection. 2) Joint Learning Models: Joint learning models jointly train an image brightening model and an object detection model to deal with poorly visible images [19], [20]. More specifically, two subnetworks [21] with common feature extraction layers are used to simultaneously detect the objects and brighten the images.

As shown in Fig. 1, current object detection models cannot detect and display objects well due to the following three challenges: 1) Out-of-focus and low-contrast objects. In low visibility conditions, objects are out of focus and low contrast, so their detectable areas are smaller and blurrier than in high quality images. This negatively affects the accuracy of conventional object detection methods originally developed for high-quality images. 2) Unfavorable representation in poor visibility conditions. Images captured in poor visibility conditions are not well visible to the user. It is necessary to enhance the images and frame the objects clearly for the user. 3) Real-time processing requirements. To meet real-time requirements, the entire process of image enhancement and object detection should be performed at more than 30 frames per second. In the two-stage models, the image enhancement step and the object recognition step are processed serially, which further limits the processing time of each step and results in unsatisfactory performance of both steps. In the joint learning models, the image enhancement model and the object detection model have completely different optimization objectives. The joint optimization of these two models may result in a wobble phenomenon, leading to better performance in image enhancement or recognition, but negatively affecting the other objective.

To this end, we study the problem of how to achieve better enhancement and detection while meeting real-time requirements. Specifically, we investigate the following three research questions. 1) How can the enhancement and detection steps be decoupled to improve both together? 2) How can objects with blurred edges be accurately detected? 3) How can images with poor visibility be enhanced to better present recognition results to users? By exploring the above questions, our work makes the following three contributions.

- We propose a novel parallel framework called Parallel Detecting and Enhancing models (PDE) that can solve the wobble problem while improving detection and enhancement performance.
- PDE introduces a tailored model for detecting objects

* Corresponding authors.

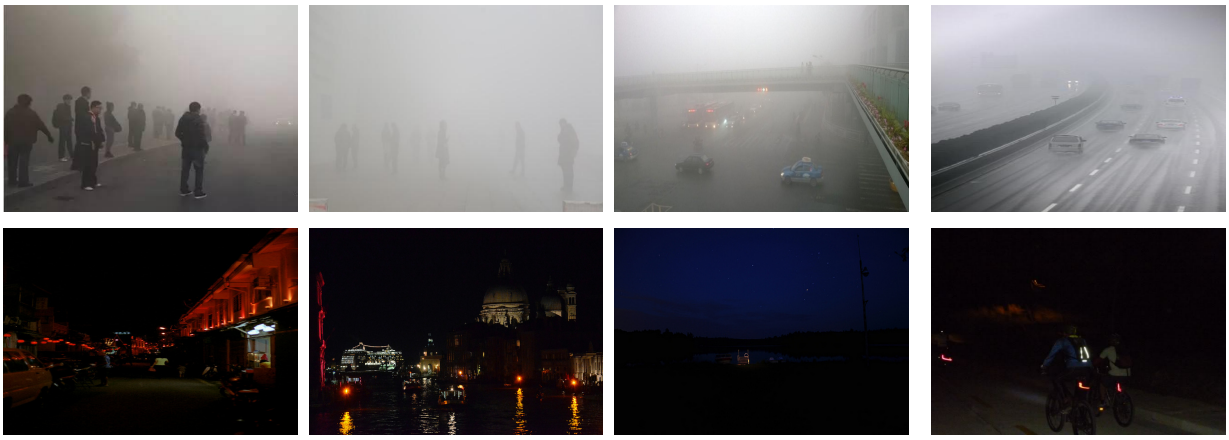

Fig. 1. Intuitive cases explaining the problems of object detection in low visibility conditions. The first row shows the images taken on foggy days. The second row shows the images taken in low-light conditions.

in low-visibility images by introducing a tiny prediction head to detect objects with smaller detection areas. In addition, PDE uses coordinated attention and multi-stage concatenation to further improve detection performance in low visibility.

- PDE incorporates a specially designed image enhancement model by developing an adaptively enhanced weighting model for each "3D Lookup Table" module to achieve better enhancement performance.

To prove the efficiency of our approach in object detection, we evaluate the proposed model on synthetic and real low visibility datasets. Experimental results show that PDE has significantly better object detection performance on two real datasets with fog (8.9%) and low light (20.6%). In addition, case studies show that PDE provides more accurate object detection and clearer rendering than other low visibility models.

The remainder of this article is organized as follows: Section II provides related work, including object detection, image enhancement, and multi-tasking in low visibility. Section III describes the proposed method used for low visibility images. Section IV presents the comprehensive experimental results of our method compared with other methods. Section V concludes our work.

II. Related Work

A. Object Detection

CNN-based target detectors can be divided into two types according to the steps of image processing: 1) Single-stage detectors: YOLOv4 [22], YOLOv5 [23], FCOS [24], and EfficientDet [25]. 2) Two-stage detectors: R-CNN [26], R-FCN [27], Mask R-CNN [28], Fast R-CNN [29], etc. From the point of view of composition, they both consist of two parts. One part is the CNN-based basic framework, which is used to extract image features. The other part is the prediction head, which is responsible for classification and localization. In addition, existing object detectors add some layers between the basic framework and the head, which are called the neck of the detector. The three structures are described in detail below.

Backbone. The backbone often uses VGG [30], ResNet [31], EfficientNet [32], CSPDarknet53 [33], Swin-Transformer [34], etc., rather than networks designed by ourselves, since these networks have been shown to have strong feature extraction capability in computer vision tasks. However, the backbone network can be fine-tuned to make it more suitable for specific tasks.

Neck. The neck was designed to make more efficient use of features extracted from the backbone network. Its main task is to further process and use the features extracted from the backbone in different stages. The neck usually consists of several top-down and several bottom-up paths. The neck is an important component of the object recognition network and connects the backbone to the head. Commonly used linking modules for the neck include FPN [35], NAS-FPN [36], PANet [37], BiFPN [25], ASFF [38], etc. The common point of these modules is the iterative use of various upsampling, downsampling, dot-sum or dot-product methods to develop aggregation strategies.

Head. In the detection task, the backbone cannot perform the localization task. Therefore, the head network is responsible for detecting the location and class of objects based on the feature maps extracted from the backbone. Head networks are generally divided into two categories: single-stage object detectors and two-stage object detectors. The most representative two-stage object detector is the R-CNN [26], [39] series. Compared to the two-stage detector, the single-stage object detector predicts both the bounding box and the object class simultaneously. The most representative single-stage object detectors are YOLO [22], [40], SSD [41], and RetinaNet [42] series.

B. Image Enhancement

Image adjustment determines a threshold based on the gray level range of the image. If it is below the threshold, automatic color gradation enhancement is applied. On the other hand, if it is above the threshold, enhancement methods based on histogram equalization and inverse equalization are performed. The adaptive image enhancement method can enhance not only low-contrast images, but also partially dark and partially

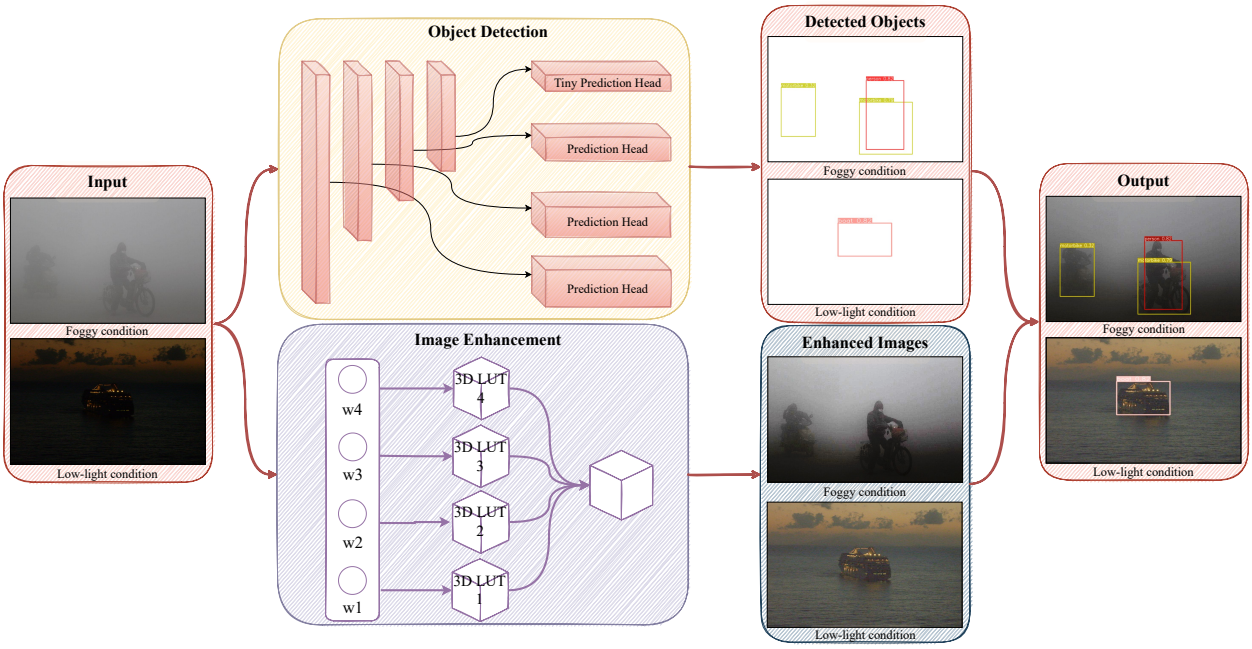

Fig. 2. An overview of the working pipeline with PDE. PDE can detect objects and enhance the low-visibility images in parallel and in real time. The label and coordinate values are obtained by the object detection model. Then we overlay the detected images with the enhanced image to get the final output.

light images with high robustness, so that the enhanced images have a better visual effect. Image adjustment is a widely used technique in image enhancement. Some classical methods [43], [44] use adaptive filters to control the contributions of the various enhancement operations so that contrast enhancement occurs in regions of high detail. [45] proposed a Deep Learning model that trains data on unpaired images. A Deep Reinforcement Learning approach is also used to decide what action to take given the current state of the images. [46] effectively transforms the color and hue of the source image by using a small CNN to learn image-adaptive 3D lookup tables.

C. Multi-Task in Low-Visibility Conditions

Existing models for object detection in low visibility include several tasks, such as image denoising and object detection. Depending on the order in which the different tasks are performed, they can be divided into two classes: two-stage models and joint learning models. Two-stage models [5], [15], [17] use classical visualization enhancement methods to process images before detection. For example, [47] proposed an AOD-Net for foggy conditions that denoises images before detection. However, the object detection models have strict requirements for deriving the time. When the image enhancement model and the object detection model are connected in series, the recognition time of both models is constrained, resulting in suboptimal performance in enhancement and detection. Joint learning models [19], [21] have performed image enhancement and object detection using a joint structure to better recognize images with low visibility. However, it is difficult to adjust the parameters to balance the completely different optimization goals of image enhancement and object detection. For this reason, [48] proposed an unsupervised adaptive system for object detection in rain and fog. After that, many works [49]–[51] emerged to improve the detection performance by using range adaptation. [52] proposed a robust learning method to resist interference from poor visibility and reduce the information loss caused by range adaptation. [20] developed a joint learning model (IA-YOLO) that combines image matching enhancement and object detection to meet the requirements of real-time recognition. In this work, we use the classical single-stage model YOLOv5 [53] as a basis and improve its performance under low visibility conditions.

III. Proposed Method

The PDE synchronizes the input image with the object detection module and the image enhancement module, as shown in Fig. 2. First, the object detection module detects the image to obtain the coordinates and classification information of the target. Second, the image enhancement module reduces the weather noise and increases the brightness of the input image to obtain a more user-friendly reference. Finally, the coordinates and classification information of the target are written into the enhanced image to obtain the result. In this section, we introduce the object detection and image enhancement modules.

A. Detection Network Module

Images captured in low visibility conditions contain interference from environmental information that makes object detection difficult. To overcome this challenge, PDE introduces a customized model for low-visibility object detection by importing a tiny prediction head and further employing coordinated attention and multi-stage concatenation to improve the performance of low-visibility detection. As shown in Fig. 3, the object detection model is a newly developed implementation of YOLOv5. These tricks help deep neural networks accurately locate and identify objects by reducing the detrimental effects

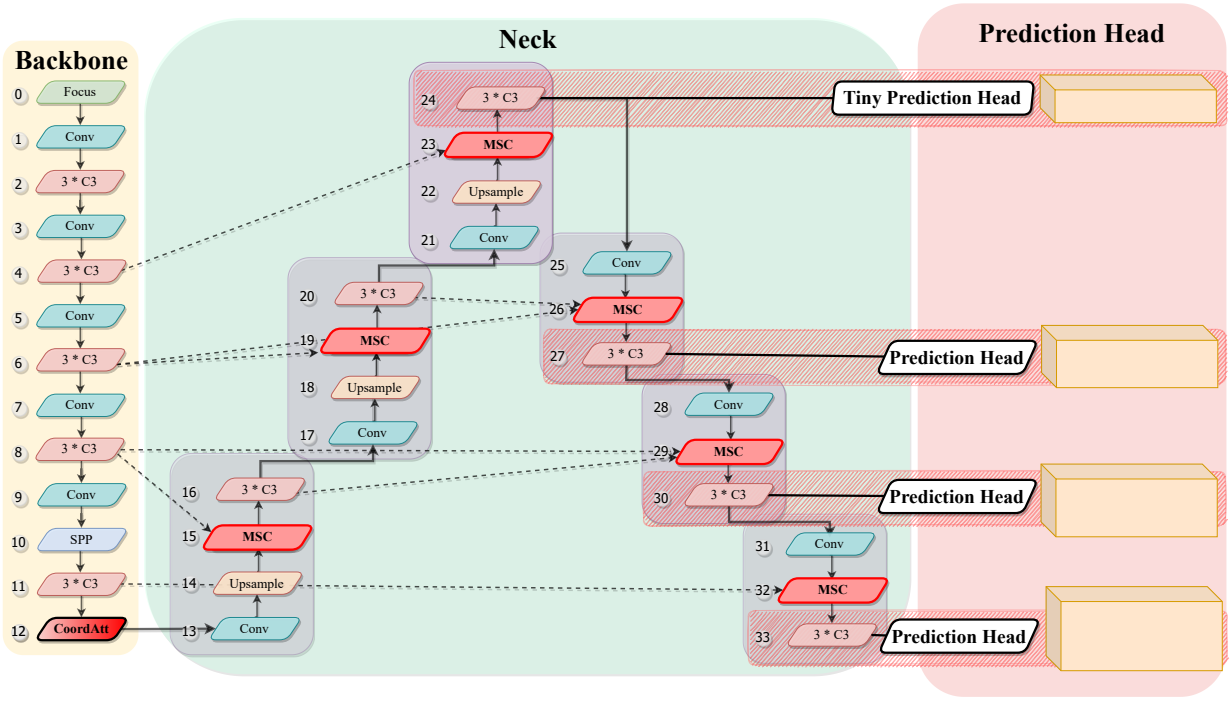

Fig. 3. The object detection module. 1) A coordinate attention block (CoordAtt) is located at the bottom of the backbone network. 2) The multi-stage concatenation module (MSC) replaces the original concatenation module. 3) The number of prediction heads has been increased from 3 to 4.

of detectable zones that are smaller and blurrier than in high-quality images. In addition, the model combines feature maps generated by shallow and deep neural networks so that semantic information and location features can be fully utilized. Moreover, an explicit supervised learning task is formed by setting a learning weight parameter. In this way, excellent learning results are obtained by perfectly distinguishing the importance of feature maps from shallow and deep neural networks. Therefore, the representative features can be accurately retrieved and appropriately represented, which improves the overall detection performance of the model. To get a clearer picture of the core of the object recognition module, we will describe the above methods and loss functions in detail below.

Coordinate Attention. Attentional mechanisms have been shown to be effective in many visual tasks. The core of the attentional mechanism is to enhance the model's ability to extract and represent important features, similar to the way humans selectively focus on important parts rather than the totality of information. However, most attention mechanisms only consider the information between channels and not the information about spatial location. This ignores the part of the information that is hidden in space and fails to extract the optimal representation of the features. Moreover, the convolution operation can only extract local relations, but not relations over long distances. To this end, we use coordinate attention [54] to capture spatial relationships over long distances with precise location data by embedding the location information into the channel attention. Specifically, each input X is decomposed into w and h dimensions, and the decomposed tensors are processed by global pooling to generate X_w and X_h, respectively. Feeding into a convolutional block with the concatenated tensors X_w and X_h generates an encoded Y that summarizes the extracted features of X_w and X_h. The set of operations can be formulated as follows:

$$Y = f(G(\{X_w, X_h\} \circ W)) \tag{1}$$

where $\{X_w, X_h\}$ means concatenating X_w and X_h, $\circ$ means convolution operation, G and f denote normalization and activation function, respectively, and W is the convolution filter. Furthermore, we split Y again to obtain $\hat{X}_w$ and $\hat{X}_h$:

$$\hat{X}_w, \hat{X}_h = Split(Y) \tag{2}$$

Also, $\hat{X}_w$ and $\hat{X}_h$ are convoluted and activated to get the final output:

$$Y_{out} = X \times \sigma(F(\hat{X}_w)) \times \sigma(F(\hat{X}_h)) \tag{3}$$

where F denotes convolution and σ denotes the sigmoid function.

By combining attention along the horizontal and vertical directions of the input sensor, each element of the attention maps can reflect in two directions whether the object of interest is present in the corresponding row and column. In this way, coordinated attention can more accurately determine the exact location of the object so that the entire model can better identify objects.

A Tiny Prediction Head. Feature sensitivity is one of the most important properties of the model for extracting key information from noisy images. The best way to achieve this is to add observations from different viewpoints and combine them to make better use of the fine-grained features from different viewpoints and achieve better feature representation.

However, we find that there are three different scales of prediction heads in the original YOLOv5 model, namely 256×256, 384×384 and 512×512, and the number of anchor images is 9. Although YOLOv5 has been observed from three perspectives, the recognition results are still not satisfactory for noisy images, as shown in Fig. 4. To this end, we add a tiny prediction head with a scale of 128×128 to extract features from a more microscopic perspective. We also increase the number of anchor frames from 9 to 12. Although this is only an incremental change to the detection head, the structure of the multi-detection head contributes significantly to improving the model performance, as shown in the Table IV of experimental results. We also found that the number of detection heads is not as large as possible and that the structure with four detection heads is the most stable and effective. Given the input X, the individual steps are as follows:

$$Z = \sum_{i=1}^{N} Anchor_i \oplus X_i \tag{4}$$

where N denotes the number of prediction heads. $Anchor$ denotes an anchor frame set based on prior knowledge. $\oplus$ denotes the matching of $Anchor$ with X to obtain a set of prediction boxes. Then, the result after matching is concatenated with the original input X. Finally, the convolution operation is used to continuously traverse the entire region to obtain the final feature map. The details are as follows:

$$Out = Conv(Concat(Z, X)) \tag{5}$$

where Out denotes the final feature map.

Multi-Stage Concatenation. The receptive field, the most important component in CNN-based models, is used to extract abstract features layer by layer. In deep layers, the receptive field is relatively large to extract features, and conversely, it is smaller in shallow layers. In general, feature maps captured by larger receptive fields have stronger semantic representation but weaker spatial representation. In contrast, feature maps captured by smaller receptive fields have weaker semantic representation but stronger spatial representation. To this end, we combine the original concatenation module with the BiFPN algorithm [25] to fully exploit the properties of feature maps from deep and shallow layers. The importance of features from different layers is defined by a learnable weighting parameter W. The parameter is defined as :

$$W_i = \frac{X_i}{\sum X_i + \varepsilon} \tag{6}$$

where X_i denotes the input of each layer, and W_i denotes the weight parameter of each input layer. ε is set to 0.0001.

According to Eq. 6 the formation of the result can be formulated as follows:

$$Y = Conv(ReLu(\sum_{i=0}^{N-1} W_i \times X_i)) \tag{7}$$

where Y denotes the result and $Conv$ denotes the convolution operation; $Relu$ denotes the activation function and N denotes the number of input layers. Both semantic features and spatial features can be transferred to different depth layers by feature fusion and mapping. This method improves the model's ability to extract and express features, and thus improves recognition performance.

Loss Function. The loss function of YOLOv5 divides the objective function into three subfunctions, namely object, classification, and regression. However, we found that the original loss function uses the basic Intersection-over-Union (IoU) loss, which limits the ability to measure the overall performance of the model. The total loss is calculated as follows:

$$Loss = w_{obj} \times loss_{obj} + w_{cls} \times loss_{cls} + w_{reg} \times loss_{reg} \tag{8}$$

where $loss_{obj}$, $loss_{cls}$, and $loss_{reg}$ denote the object objective function, classification objective function, and regression objective function, respectively. w_{obj}, w_{cls}, and w_{reg} denote their weighting values set a priori to 0.3, 0.05, and 0.7, respectively.

To better measure the difference between confidence in the predicted object and the true value, we construct this objective function $loss_{obj}$ based on cross-entropy loss. For a given predicted value x and a true value y, the equation is as follows:

$$loss_{obj} = -\frac{1}{n}\sum_{i=1}^{n}(y_i \times \ln i + (1 - y_i) \times \ln(1 - x_i)) \tag{9}$$

Although the predictions are multiclassification, there is only one positive sample, so we use the loss of cross entropy. After we use the cross entropy as the loss function, the gradient of the backpropagation is no longer associated with the derivative of the sigmoid function. This avoids the disappearance of the gradient to some extent. $loss_{obj}$ is the same as $loss_{cls}$.

In the regression task, the most direct indicator to determine the distance between the predicted box and the ground truth is the intersection over union (IoU), and $IoU = \frac{|A \cap B|}{|A \cup B|}$, however, does not accurately reflect the intersection of the two boxes and cannot be trained further due to disjunction. Therefore, we use the complete-IoU [55] to construct the objective function of the regression task. We consider the similarity of the aspect ratio between the ground truth and the predicted box.

$$loss_{reg} = 1 - IoU + \frac{\beta^2 \times (b_p, b_g)}{c^2} + \alpha \times \nu \tag{10}$$

$$\alpha = \frac{\nu}{\nu - IoU + (1 + \varepsilon)} \tag{11}$$

$$\nu = \frac{4}{\pi_2} \times (\frac{w_g}{h_g} - \frac{w_p}{h_p})^2 \tag{12}$$

where b_p and b_g denote the centers of the prediction box and the ground truth, respectively. β denotes the Euclidean distance between the two centers. c denotes the distance of the diagonals of the smallest region containing both the prediction box and the ground truth. α denotes the weighting parameter. ν is used to measure the similarity between the aspect ratio of the prediction box and the ground truth.

B. Image Enhancement Module

The 3D lookup table is an algorithm that reconstructs the hue of an image by creating a color map. The essence of the 3D lookup table is an mapping relation: $(R, G, B) = f(r, g, b)$, where f represents the mapping function. Moreover, it is an intuitive idea to learn a classifier to classify the scene. Suppose M 3D lookup tables, denoted by $\{\mu_n\}_{n=1,\ldots,M}$, are learned. The classifier outputs N probabilities $\{p_n\}_{n=1,\ldots,N}$ for classifying the scene. The process of 3D lookup table selection can be described as follows:

$$q = \mu_i(x), \quad s.t. \quad i = arg \max_n p_n \tag{13}$$

where x denotes an input image and q the output. Another common method for improving image quality is to manually adjust the parameters of a 3D lookup table. However, manually adjusting the parameters is extremely time consuming when processing large images. The parameters need to be adjusted based on scenarios that have different negative effects. Therefore, the applicability of the method is hindered by a lack of flexibility and practicality.

H. Zeng developed an end-to-end adaptive image enhancement method [46] based on 3D lookup tables and a convolutional network. The model learns how to improve image quality based on paired data, namely the affected images and the images optimized by experts. To this end, we propose a specific model for image enhancement by developing adaptive models for each "3D Lookup Table" to improve the weighting evaluation. Moreover, in this work, we train the image enhancement model by combining preprocessed images, i.e., foggy images and low-light images, with clear original images.

The image enhancement model introduces 4 basic 3D lookup tables along with a CNN-based model g that predicts weights for the output of each 3D lookup table. For an input image x, the final enhancement result is as follows:

$$q = \sum_{n=1}^{4} w_n \mu_n(x) \tag{14}$$

where $\{w_n\}_{n=1,\ldots,4} = g(x)$ are the content-dependent weights output by the CNN-based model. Specifically, we use different 3D lookup tables to enhance different images. Moreover, the color space of the image is transformed using 3D lookup tables, while the CNN weight predictor extracts information about the image content, including hue, brightness, contrast, etc. The weights obtained by the CNN predictors are assigned to the corresponding 3D lookup tables. Therefore, our model adaptively improves the image quality according to the image content and scene in low visibility conditions.

IV. Experiments

We evaluate the effectiveness of PDE in fog and low-light conditions. We report the object detection metric mAP (average of all 10 IoU thresholds in the range of [0.5: 0.95]) and the image enhancement metrics PSNR (Peak Signal to Noise Ratio) and SSIM (Structure Similarity). We will present this section under the following aspects.

A. Experimental Details

Datasets. For the two tasks that PDE faces, i.e., target detection and image enhancement, we need to take different approaches to create datasets for the corresponding tasks so that we can effectively evaluate the performance of the model.

In object detection, we first evaluate the detection performance of the model under three conditions, including normal, foggy, and low light. We use the VOC dataset [56], [57] as a benchmark and the RTTS dataset [58] and the ExDark dataset [59] as test sets. To make better use of these datasets, we filtered out the common categories of the datasets. The VOC dataset shares five categories with the RTTS dataset, namely pedestrians, cars, buses, bicycles, and motorcycles. Similarly, the VOC dataset shares 10 categories with the ExDark dataset, namely, bicycles, boats, bottles, buses, cars, cats, chairs, dogs, motorcycles, and people. The VOC_5c training dataset and the VOC_5c test dataset, namely VOC_5c_train and VOC_5c_test, are created after screening and consist of 8111 and 2734 images, respectively.

Although we already have a dataset for normal conditions, we lack sufficient images of foggy conditions and low light conditions. Therefore, we use a weather simulation algorithm to simulate images under low visibility conditions. According [60], for the original input image $O(x)$, the foggy image $F(x)$ can be calculated as follows:

$$F(x) = O(x) \times g(x) + L \times (1 - g(x)) \tag{15}$$

where L denotes global atmospheric light, and $g(x)$ denotes the medium transmission map, which is defined as:

$$g(x) = e^{-\beta} \times s(x) \tag{16}$$

where β denotes the scattering coefficient of the atmosphere, and $s(x)$ denotes the scene depth which is calculated by

$$s(x) = -0.04 \times \rho + \sqrt{max(row, col)} \tag{17}$$

where ρ denotes the Euclidean distance from the current coordinate to the pixel coordinate of the image center, row and col represent the number of rows and columns of the images. Combining the Eq. 15, 16 and 17, we obtain the following equation for the generation of fog images:

$$F(x) = O(x) \times e^{-\beta} \times s(x) + L \times (1 - e^{-\beta} \times s(x)) \tag{18}$$

In this work, L is set to 0.5 and β is calculated using the formula $\beta = 0.05 + 0.01 \times Num$. Num is set to a random integer between 0 and 9. In this way, for each input image, we get up to 10 foggy images with different effects of fog concentration.

Similarly, we simulate low lighting conditions to create the low lighting conditions dataset. For a given input image, each pixel x in the image is transformed as follows:

$$f(x) = x^{\gamma} \tag{19}$$

where γ is determined randomly from a uniform distribution with a range of values of [1.5, 5].

TABLE I. AN OVERVIEW OF ALL DATA SETS USED IN THIS EXPERIMENT

Dataset	Number
VOC_5c_train	8111
VOC_5c_test	2734
VOC_10c_train	12334
VOC_10c_test	3760
VOC_fog_train	8111
VOC_fog_test	2734
VOC_low-light_train	12334
VOC_low-light_test	3760
RTTS	4322
ExDark	2563
LUTs_fog_train	20000
LUTs_fog_test	2000
LUTs_low-light_train	20000
LUTs_low-light_test	2000

To achieve ideal recognition performance under normal and low visibility conditions, we use a hybrid data training scheme for PDE. Each image in the normal datasets has a 2/3 probability of being randomly tagged with some kind of fog or converted to a low-visibility image before being input to the model for training. The hybrid data contains images from both normal and low visibility situations. The model becomes more robust when it learns with normal and low visibility images simultaneously, resulting in high performance.

Second, training an image enhancement model for image enhancement tasks requires a large amount of data to achieve an excellent result. Therefore, we extend the data again based on the simulated images in foggy and low-light conditions in the object recognition task. For the foggy conditions, we first add three random fog patches to each image in the VOC_5c_train dataset. Second, we randomly select 20,000 images from this dataset to form the fog training dataset, i.e., LUTs_fog_train. Similarly, we first randomly add fog to each image in the VOC_5c_test dataset. Second, we randomly select 20,000 images from this dataset to form the test dataset under foggy conditions, i.e., LUTs_fog_test.

In low light conditions, we perform the same steps to obtain the dataset, i.e. LUTs_low-light_train and LUTs_low-light_test.

We count the number of all records for this experiment, as shown in Table I. VOC_5c_train, VOC_5c_test, VOC_10c_train, VOC_10c_test, VOC_fog_train, VOC_fog_test, VOC_low-light_train, and VOC_low-light_test denote training and test sets, respectively, for object detection under normal, foggy, and low-light conditions. RTTS and ExDark are real-world datasets consisting of images taken under foggy and low-light conditions, respectively. LUTs_fog_train, LUTs_fog_test, LUTs_low-light_train, and LUTs_low-light_test denote training and test datasets for image enhancement in foggy and low-light conditions, respectively.

Baselines. This work focuses on improving the accuracy of object detection in low visibility conditions, complemented by image enhancement techniques to obtain more user-friendly references. Therefore, we perform comparison experiments and ablation experiments mainly for the object recognition module, while for the image enhancement module, we only present its experimental results without detailed comparison with other excellent methods.

To evaluate the universality and effectiveness of PDE in fog and low-light conditions, we choose YOLOv5 as our baseline model. In addition, we compare our model with other excellent models for detecting objects in low visibility. We choose the real-time target detection model YOLOv3 as our comparison model. We also choose GridDehaze [35], MSBDN [5], and ZeroDCE [15], the most widely used CNN-based image enhancement methods, to process images before detection and then combine them with the object detection model YOLOv3 [43]. GridDehaze and MSBDN are both image enhancement models for removing fog by developing novel network modules to learn more effective feature representations for image unveiling. ZeroDCE achieves effective image enhancement by implementing intuitive and simple nonlinear curve mapping to adapt to different lighting conditions. For the domain adaptation approach, we choose DAYOLO [19], which combines multiple adaptation paths and corresponding domain classifiers with the YOLO object detector to produce domain-invariant features. For the multi-task learning algorithm, we choose DSNet [22], which can learn denoising and detection together. We also choose IA-YOLO [34], which can adaptively enhance each image to improve detection performance.

B. Experiments Results

To fully demonstrate detection performance, for each model we evaluate the model's ability to recognize objects under different conditions, namely normal, foggy, and low light. The improvements are calculated by comparing PDE with the best baseline (underlined). From Table II and Table III, it can be seen that PDE significantly outperforms the other SOTA models at low visibility in the detection scene in all data sets and at all settings. In particular, for the mAP metric, PDE outperforms the baseline model by 8.9% (RTTS) and 19.7% (VOC_fog_test) in foggy conditions. In low-light conditions, PDE outperforms the baseline model by 20.6% (ExDark) and 15.8% (VOC_low-light_test).

These results demonstrate the consistent superiority of our PDE in detection performance under poor visibility conditions. Moreover, the PDE also performs better than the corresponding best baselines in a normal scene. This phenomenon proves the strong scalability of PDE.

Image enhancement is a secondary task that helps improve the display for the user, as this work focuses on target detection. Therefore, we did not perform comparison experiments for the image enhancement task. We evaluate the model's ability to enhance images in fog and low-light conditions. For the PSNR metric, PDE achieves a score of 23.64 (fog test set) and 23.97 (low-light test set). For the SSIM metric, PDE achieves a value of 0.838 (fog test set) and 0.827 (low-light test set). In the following subsection IV-D, we conduct a case study to illustrate the excellent results of the image enhancement task.

C. Ablation Study

To test the effectiveness of the object detection model in PDE, we compare the detection performance of our model

TABLE II. COMPARISON OF DETECTION PERFORMANCE WITH BASELINES IN TWO SCENARIOS, INCLUDING NORMAL AND FOGGY CONDITIONS. THE IMPROVEMENTS ARE COMPUTED BY COMPARING OUR MODEL WITH THE CORRESPONDING BEST BASELINES (UNDERLINED)

Model	Train data	VOC_5c_test	VOC_fog_test	RTTS
MSBDN [5]	VOC_5c_train	-	57.4	30.2
GridDehaze [35]	VOC_5c_train	-	58.2	31.4
DAYOLO [19]	Hybrid data	56.5	55.1	29.9
DSNet [22]	Hybrid data	53.3	67.4	28.9
IA-YOLO [34]	Hybrid data	73.2	72.0	37.0
YOLOv3 [43]	VOC_5c_train	70.1	31.1	28.8
YOLOv3 [43]	Hybrid data	64.1	63.4	30.8
YOLOv5 [23]	VOC_5c_train	86.2	68.5	45.1
YOLOv5 [23]	Hybrid data	85.6	71.4	50.5
PDE	Hybrid data	**86.7**(1.3%↑)	**85.5**(19.7%↑)	**55.0**(8.9%↑)

TABLE III. COMPARISON OF RECOGNITION PERFORMANCE WITH BASELINES IN TWO SCENARIOS, INCLUDING NORMAL AND LOW LIGHT CONDITIONS. THE IMPROVEMENTS ARE COMPUTED BY COMPARING OUR MODEL WITH THE CORRESPONDING BEST BASELINES (UNDERLINED)

Model	Train data	VOC_10c_test	VOC_low-light_test	ExDark
ZeroDCE [15]	VOC_10c_train	-	33.6	34.4
DAYOLO [19]	Hybrid data	41.7	21.5	18.2
DSNet [22]	Hybrid data	64.1	43.8	37.0
IA-YOLO [34]	Hybrid data	70.0	59.4	40.4
YOLOv3 [43]	VOC_10c_train	69.1	45.9	36.4
YOLOv3 [43]	Hybrid data	65.3	52.3	37.0
YOLOv5 [23]	VOC_10c_train	78.2	60.8	43.2
YOLOv5 [23]	Hybrid data	77.1	64.5	45.0
PDE	Hybrid data	**79.5**(3.1%↑)	**74.7**(15.8%↑)	**54.3**(20.6%↑)

TABLE IV. ABLATION ANALYSIS OF MODULES OF OUR MODEL IN REAL DATA SETS UNDER LOW VISUAL CONDITIONS. CA DENOTES THE COORDINATE ATTENTION MODULE. MSC DENOTES THE CONCATENATION MODULE COMBINED WITH MULTI-STAGE FEATURE FUSION. MH DENOTES THE MODULE WITH MULTIPLE PREDICTION HEADS. THE IMPROVEMENTS ARE COMPUTED BY COMPARING THE VARIANTS WITH YOLOV5 (UNDERLINED)

Model	Method			RTTS	ExDark
	CA	MSC	MH	mAP	mAP
YOLOv5	✗	✗	✗	50.5	45.0
PDE w/o MH	✓	✓	✗	52.5 (3.9%↑)	49.9 (10.9%↑)
PDE w/o MSC	✓	✗	✓	52.9 (4.7%↑)	53.6 (19.1%↑)
PDE w/o CA	✗	✓	✓	**55.0** (8.9%↑)	53.8 (19.5%↑)
PDE	✓	✓	✓	**55.0** (8.9%↑)	**54.3** (20.6%↑)

with its variants on two real datasets (RTTS and ExDark) in Table IV. In the following experiments, we use the data as our training dataset. “PDE w/o MH” means we omit the tiny prediction head in PDE. “PDE w/o MSC” means we omit the multi-stage concatenation module in PDE. “PDE w/o CA” means we omit the coordinate attention module in PDE.

As shown in Table IV, “PDE w/o MH” is 3.9% and 10.9% higher than YOLOv5 in RTTS and ExDark, respectively. However, “PDE w/o MSC” is 4.7% and 19.1% higher than YOLOv5, respectively, whereas “PDE w/o CA” is 8.9% and 19.5% higher than YOLOv5, respectively. Although the coordinate attention module can improve the performance of the model, the effect is not very large when the Table IV is analyzed. On the contrary, the tiny prediction head and multi-stage concatenation module significantly improve the performance of the model. In particular, the growth rate obtained with “PDE w/o MSC” reaches the maximum in ExDark, which proves that the module uses the features extracted from the backbone network very effectively under low light conditions. Moreover, the growth rate of “PDE w/o CA” reaches the maximum in RTTS, where the multi-stage concatenation module fully utilizes the effective features in the images combined with the tiny prediction head to perform target detection.

According to Table IV, PDE consistently outperforms the other variants, underscoring the need for and effectiveness of these methods, as noted in III-A.

D. Case Study

In Fig. 4, we visualized the detection result on two real datasets (RTTS and ExDark). In particular, we compare the detection results of the base models YOLOv5 and PDE. As you can see in Fig. 4, PDE can achieve better object detection accuracy and user representation in low- visibility. Moreover, inference time is an important metric to evaluate

Fig. 4. Visualization. Detection results of YOLOv5 (middle row) and PDE (bottom row) on RTTS (columns 1, 2) and ExDark (columns 3, 4). PDE achieves better object detection accuracy and better visualization for the user in low visibility images.

the practicality of models. Therefore, we conduct extensive test experiments to evaluate PDE by processing 480×480 images on a single GTX 2080Ti GPU. The experiment showed that PDE can process more than 30 frames per second. Therefore, PDE can achieve better user representation and detection while meeting real-time requirements.

V. Conclusion

In this paper, we note that the existing low visibility models suffer from the wobble phenomenon caused by the absence of better detection and image enhancement performance. We propose the parallel detection and enhancement model (PDE) to ensure that image enhancement and object detection perform their tasks. For object detection, PDE introduces a tailored model for low-visibility object detection by introducing a tiny prediction head, combined with coordinate attention and multi-stage concatenation modules. For image enhancement, PDE proposes a dedicated image enhancement model by developing an adaptively enhanced weighting model for each "3D Lookup Table" module. By decoupling these two concepts, PDE can improve the overall performance. Extensive experiments show that PDE achieves better accuracy in detecting low-visibility objects and more user-friendly reference in real time in all situations.

Acknowledgment

This work was supported by the National Natural Science Foundation of P. R. China (Nos. 62002068).

References

[1] Lin S, Xiao G, Yan Y, Suter D, Wang H. "Hypergraph optimization for multi-structural geometric model fitting," In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019, 33(1): 8730-8737.

[2] Lin S, Luo H, Yan Y, Xiao G, Wang H. "Co-clustering on Bipartite Graphs for Robust Model Fitting," *IEEE Transactions on Image Pro cessing*, 2022, 31: 6605-6620.

[3] Lin S, Wang X, Xiao G, Yan Y, Wang H. "Hierarchical representation via message propagation for robust model fitting," *IEEE Transactions on Industrial Electronics*, 2020, 68(9):8582-8592.

[4] Z. Xia, S. Song, L. E. Li, and G. Huang, "3d object detection with pointformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7463–7472.

[5] Yang H, Lin S, Cheng L, Lu Y, Wang H. "SCINet: Semantic Cue Infusion Network for Lane Detection," *IEEE International Conference on Image Processing*, 2022, 1811-1815.

[6] Y. Gong, X. Yu, Y. Ding, X. Peng, J. Zhao, and Z. Han, "Effective fusion factor in FPN for tiny object detection," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2021, pp. 1160–1168.

[7] P. Sun et al., "Sparse r-cnn: End-to-end object detection with learnable proposals," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 14454–14463.

[8] T. Yin, X. Zhou, and P. Krahenbuhl, "Center-based 3d object detection and tracking," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 11784–11793.

[9] A. Pfeuffer, M. Schön, C. Ditzel, and K. Dietmayer, "The ADUULM-Dataset-a Semantic Segmentation Dataset for Sensor Fusion.," 2020.

[10] T. Song, Y. Kim, C. Oh, and K. Sohn, "Deep Network for Simultaneous Stereo Matching and Dehazing.," in *BMVC*, 2018, p. 5.

[11] Y. Zhang, J. Zhang, and X. Guo, "Kindling the darkness: A practical low-light image enhancer," in *Proceedings of the 27th ACM international conference on multimedia*, 2019, pp. 1632–1640.

[12] C. Guo et al., "Zero-reference deep curve estimation for low-light image enhancement," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1780–1789.

[13] F. Lv, F. Lu, J. Wu, and C. Lim, "MBLLEN: Low-Light Image/Video Enhancement Using CNNs.," in *BMVC*, 2018, vol. 220, no. 1, p. 4.

[14] Q. Zhu, J. Mai, and L. Shao, "Single image dehazing using color attenuation prior.," 2014.

[15] H. Dong et al., "Multi-scale boosted dehazing network with dense

feature fusion," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2157–2167.

[16] K. He, J. Sun, and X. Tang, "Single image haze removal using dark channel prior," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 12, pp. 2341–2353, 2010.

[17] X. Liu, Y. Ma, Z. Shi, and J. Chen, "Griddehazenet: Attention-based multi-scale network for image dehazing," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 7314–7323.

[18] W. Ren et al., "Low-light image enhancement via a deep hybrid network," *IEEE Transactions on Image Processing*, vol. 28, no. 9, pp. 4364–4375, 2019.

[19] B. Li, X. Peng, Z. Wang, J. Xu, and D. Feng, "Aod-net: All-in-one dehazing network," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 4770–4778.

[20] W. Liu, G. Ren, R. Yu, S. Guo, J. Zhu, and L. Zhang, "Image-Adaptive YOLO for Object Detection in Adverse Weather Conditions," *arXiv preprint* arXiv:2112.08088, 2021.

[21] S.-C. Huang, T.-H. Le, and D.-W. Jaw, "DSNet: Joint semantic learning for object detection in inclement weather conditions," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 8, pp. 2623–2633, 2020.

[22] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint* arXiv:2004. pp. 10934, 2020.

[23] G. Jocher et al., "ultralytics/yolov5: v5. 0-YOLOv5-P6 1280 models AWS Supervise. ly and YouTube integrations," *Zenodo*, vol. 11, 2021.

[24] Z. Tian, C. Shen, H. Chen, and T. He, "Fcos: Fully convolutional one-stage object detection," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 9627–9636.

[25] M. Tan, R. Pang, and Q. V. Le, "Efficientdet: Scalable and efficient object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 10781–10790.

[26] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[27] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," *Advances in neural information processing systems*, vol. 29, pp. 379-387, 2016.

[28] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[29] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[30] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint* arXiv:1409.1556, 2014.

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[32] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*, 2019, pp. 6105–6114.

[33] C.-Y. Wang, H.-Y. M. Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, "CSPNet: A new backbone that can enhance learning capability of CNN," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2020, pp. 390–391.

[34] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10012–10022.

[35] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[36] G. Ghiasi, T.-Y. Lin, and Q. V. Le, "Nas-fpn: Learning scalable feature pyramid architecture for object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 7036–7045.

[37] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8759–8768.

[38] S. Liu, D. Huang, and Y. Wang, "Learning spatial fusion for single-shot object detection," *arXiv preprint* arXiv:1911.09516, 2019.

[39] T. Cheng, X. Wang, L. Huang, and W. Liu, "Boundary-preserving mask r-cnn," in *European conference on computer vision*, 2020, pp. 660–676.

[40] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint* arXiv:1804.02767, 2018.

[41] W. Liu et al., "Ssd: Single shot multibox detector," in *European conference on computer vision*, 2016, pp. 21–37.

[42] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[43] A. Polesel, G. Ramponi, and V. J. Mathews, "Image enhancement via adaptive unsharp masking," *IEEE transactions on image processing*, vol. 9, no. 3, pp. 505–510, 2000.

[44] W. Wang, Z. Chen, X. Yuan, and F. Guan, "An adaptive weak light image enhancement method," in *Twelfth International Conference on Signal Processing Systems*, 2021, vol. 11719, p. 1171902.

[45] Y. Hu, H. He, C. Xu, B. Wang, and S. Lin, "Exposure: A white-box photo post-processing framework," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 2, pp. 1–17, 2018.

[46] H. Zeng, J. Cai, L. Li, Z. Cao, and L. Zhang, "Learning image-adaptive 3d lookup tables for high performance photo enhancement in real-time," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[47] B. Li, X. Peng, Z. Wang, J. Xu, and D. Feng, "An all-in-one network for dehazing and beyond," *arXiv preprint* arXiv:1707.06543, 2017.

[48] V. A. Sindagi, P. Oza, R. Yasarla, and V. M. Patel, "Prior-based domain adaptive object detection for hazy and rainy conditions," in *European Conference on Computer Vision*, 2020, pp. 763–780.

[49] Y. Chen, W. Li, C. Sakaridis, D. Dai, and L. Van Gool, "Domain adaptive faster r-cnn for object detection in the wild," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3339–3348.

[50] M. Hnewa and H. Radha, "Multiscale domain adaptive yolo for cross-domain object detection," in *2021 IEEE International Conference on Image Processing (ICIP)*, 2021, pp. 3323–3327.

[51] S. Zhang, H. Tuo, J. Hu, and Z. Jing, "Domain Adaptive YOLO for One-Stage Cross-Domain Detection," in *Asian Conference on Machine Learning*, 2021, pp. 785–797.

[52] M. Khodabandeh, A. Vahdat, M. Ranjbar, and W. G. Macready, "A robust learning approach to domain adaptive object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 480–490.

[53] Y. Fang, X. Guo, K. Chen, Z. Zhou, and Q. Ye, "Accurate and Automated Detection of Surface Knots on Sawn Timbers Using YOLO-V5 Model.," *BioResources*, vol. 16, no. 3, 2021.

[54] Q. Hou, D. Zhou, and J. Feng, "Coordinate attention for efficient mobile network design," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 13713–13722.

[55] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-IoU loss: Faster and better learning for bounding box regression," in *Proceedings of the AAAI conference on artificial intelligence*, 2020, vol. 34, no. 07, pp. 12993–13000.

[56] 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. 2, pp. 303–338, 2010.

[57] M. Everingham and J. Winn, "The pascal visual object classes challenge 2012 (voc2012) development kit," *Pattern Analysis, Statistical Modelling and Computational Learning*, Tech. Rep, vol. 8, no. 5, 2011.

[58] B. Li et al., "Benchmarking single-image dehazing and beyond," *IEEE Transactions on Image Processing*, vol. 28, no. 1, pp. 492–505, 2018.

[59] 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, pp. 30–42, 2019.

[60] H. Israël and F. Kasten, "Koschmieders theorie der horizontalen sichtweite," in *Die Sichtweite im Nebel und die Möglichkeiten ihrer künstlichen Beeinflussung, Springer*, 1959, pp. 7–10.