DENet: Detection-driven Enhancement Network for Object Detection under Adverse Weather Conditions

Qingpao Qin[1], Kan Chang[1]*, Mengyuan Huang[1], and Guiqing Li[2]

[1] School of Computer and Electronic Information,
Guangxi University, Nanning, China
qqp@st.gxu.edu.cn, changkan0@gmail.com, Hmorry@163.com
[2] School of Computer Science & Engineering,
South China University of Technology, Guangzhou, China
ligq@scut.edu.cn

Abstract. Recently, the deep learning-based object detection methods have achieved a great success. However, the performance of such techniques deteriorates on the images captured under adverse weather conditions. To tackle this problem, a detection-driven enhancement network (DENet) which consists of three key modules for object detection is proposed. By using Laplacian pyramid, each input image is decomposed to a low-frequency (LF) component and several high-frequency (HF) components. For the LF component, a global enhancement module which consists of four parallel paths with different convolution kernel sizes is presented to well capture multi-scale features. For HF components, a cross-level guidance module is used to extract cross-level guidance information from the LF component, and affine transformation is applied in a detail enhancement module to incorporate the guidance information into the HF features. By cascading the proposed DENet and a common YOLO detector, we establish an elegant detection framework called DE-YOLO. Through experiments, we find that DENet avoids heavy computation and faithfully preserves the latent features which are beneficial to detection, and DE-YOLO is effective for images captured under both the normal condition and adverse weather conditions. The codes and pre-trained models are available at: https://github.com/NIvykk/DENet.

1 Introduction

Recently, the convolutional neural network (CNN)-based object detection methods, including the two-stage detectors [1–4] and the one-stage detectors [5–8], have achieved remarkable performance on benchmark datasets [9, 10]. However, existing object detection models are usually trained on high-quality images. In real applications such as autonomous driving, images may be captured under adverse weather conditions, such as low-light and foggy conditions. Due to the large domain shift between the training and testing images, these object detection models may fail to provide reliable results under adverse weather conditions.

To address this problem, one straightforward solution is to fine-tune the pre-trained object detection models on the target domain. However, it is expensive to collect a new dataset with handcrafted annotations on the target domain. Moreover, the fine-tuned models may suffer from a performance drop on the source domain. Therefore, such a solution is impracticable.

An alternative approach is to apply the unsupervised domain adaptation (UDA). The UDA-based methods [11–19] adopt the strategy of adversarial training, and attempt to learn robust domain-invariant features from both the labeled images on the source domain and the unlabeled low-quality images on the target domain. Such a strategy improves the performance on the target domain, while maintains a satisfactory detection results on the source domain. In addition, by using the UDA-based methods, there is no need to collect a large-scale annotated dataset for the new target domain. Despite the above advantages, if the gap between the two domains is too large, it is still hard for the UDA-based methods to align the features from the two different distributions.

As another potential solution, multi-task learning (MTL) is utilized by some methods [20, 21]. Compared with the UDA-based methods, the MTL-based methods achieve a better performance on the target domain. However, the accuracy of this type of methods usually decreases on the source domain.

Intuitively, it is possible to improve the performance of detection under adverse weather conditions by utilizing the advanced image enhancement techniques [22–32] beforehand. However, in order to establish a sophisticated non-linear mapping from a low-quality image to the corresponding high-quality version, many enhancement models have a large model size. Applying such a complex model before the detector is harmful for real-time detection. Although there are some lightweight models which require short running time, they can only bring limited improvement to the performance of object detection as they are designed only for the human visual system. Another limitation lies in that many enhancement models are trained by using the enhancement loss, which measures the distance between the enhanced image and a clean ground-truth (GT). On one hand, a clean GT image may not be available in real applications. On the other hand, such a loss function treats each pixel equally and does not pay more attention to the structured features that are beneficial to object detection.

To tackle the limitations of the above methods, a detection-driven enhancement network (DENet) is proposed in this paper. Such a network is designed for the detection task, and is able to identify and pay special attention to those latent features that are important to object detection. In DENet, we use Laplacian pyramid [33] to decompose the input image into a low-frequency (LF) component and several high-frequency (HF) components. Usually, the weather-specific information, such as contrast and illumination, are more related to the LF component. Therefore, to alleviate the effects of adverse weather on detection, it is important to well capture and refine the multi-scale information in the LF component. To this end, a global enhancement module (GEM) which consists of four parallel paths with different convolution kernel sizes is designed for the LF component. Due to the reason that weather-specific information interacts with

objects, we extract cross-level guidance information from the LF component, and then apply affine transformation to incorporate the guidance information into the features of each HF component, so that the HF information, such as edges and textures, can be well depicted. To avoid the disadvantages of the normal enhancement loss function, we assume that clean GT image is not available. DENet is combined with a normal YOLOv3 model and the detection loss is directly used for training. As there is no need to establish an accurate mapping to the clean GT image for each pixel, a lightweight design of DENet still leads to satisfactory detection results.

In summary, our contributions are threefold: 1) An extremely lightweight enhancement model (with only 45K parameters) called DENet is proposed. For effective and efficient enhancement, a Laplacian-pyramid-based structure is applied in DENet, where a GEM is designed for enhancing the LF component, and a detail enhancement module (DEM) is developed to refine the HF components adaptively. 2) By cascading DENet and a common detector such as YOLOv3 [7], an elegant end-to-end detection framework called DE-YOLO (cascaded detection-driven enhancement and YOLO) is obtained. When training DE-YOLO, we only use the normal detection loss, and does not require high-quality GT images. 3) Compared with different types of state-of-the-art (SOTA) methods, the proposed method is able to provide the most faithful detection results under both the normal condition and the adverse weather conditions, while requires very limited running time.

2 Related Work

2.1 UDA-based and MTL-based Methods

Recently, some researchers have proposed to apply the UDA-based methods to improve the performance of detection under adverse weather conditions. Chen et al. [11] introduced image-level and instance-level domain classifiers for the two-stage detector faster R-CNN [3]. Following this work, many two-stage-detector-based methods [12–15] have been proposed. For the one-stage detector, MS-DAYOLO [16] employed multi-scale image-level domain classifiers. Based on MS-DAYOLO, multi-scale instance-level domain adaptation and consistency regularization are introduced in DAYOLO [17], which result in a better performance. Sindagi et al. [18] proposed a domain adaptive object detection framework based on the prior knowledge of degradation models.

Some MTL-based methods have also been proposed. For example, Huang et al. [20] designed a framework which jointly learns three tasks, including visibility enhancement, object classification and localization. Cui et al. [21] explored the physical noise model under low-light condition, and trained a model to simultaneously predict the degradation parameters of images and detect objects, so that the intrinsic feature representation can be well extracted.

Note that except the basic detector, no extra parameters are needed in the testing phase of the UDA-based and MTL-based methods. Therefore, applying these two types of methods has no influence on the detection speed.

2.2 Image Enhancement Methods

Image enhancement methods can be used to improve the visual quality of the images taken under adverse weather conditions. For the low-light condition, many CNN-based low light image enhancement (LLIE) methods have been developed, including the retinex-based methods [22–24], the adversarial-learning-based methods [25], the mapping-based models [26], the unsupervised-learning-based methods [27, 28], etc. For the foggy condition, the typical CNN-based defogging methods include the mapping-based methods [29, 30] which directly predict the clean images, and the degeneration-based models [31, 32] which attempt to estimate the transmission map of a hazy input.

2.3 Joint Enhancement and Detection Methods

Only a few joint enhancement and detection (JED) methods have been put forward. Liu et al. [34] proposed a joint low-light enhancement and face detection method, which establishes a reverse mapping to properly model the degradation of images, and a dual-path fusion architecture to fuse the features extracted from both the enhancement and face detection phases. However, this method requires both pair and unpair training data. Liu et al. [35] presented a JED framework called IA-YOLO, which uses a CNN-based predictor to learn the hyperparameters for the filters in a fully differentiable image processing (DIP) module. However, the filters in the DIP module are handcrafted and cascaded in a fixed order, which may limit the ability of DIP module. Different from IA-YOLO, our DENet decomposes the input image into LF and HF components and enhances each component adaptively. Such a structure can provide more flexibility to adaptively suppress the effects of weather-specific information and refine the latent features of objects.

3 Proposed Method

As shown in Fig. 1, our pipeline contains a DENet and a normal object detection model YOLOv3. DENet is responsible for adaptively enhancing the input low-light/foggy images, so that the weather-specific information can be well removed and the latent discriminative features can be well preserved. To reduce computational complexity and guarantee a reliable enhancement, a Laplacian-pyramid-based structure is applied in DENet (Sec. 3.1). Afterwards, the enhanced images are fed to YOLOv3 for detection. By training the cascaded DENet and YOLOv3 models in an end-to-end manner with the normal detection loss, a joint enhancement-detection framework DE-YOLO is obtained.

3.1 Laplacian-pyramid-based Enhancement

By using Laplacian pyramid decomposition [33], an input image $\mathbf{I}$ with a resolution of $h \times w$ can be decomposed into an LF component and several HF

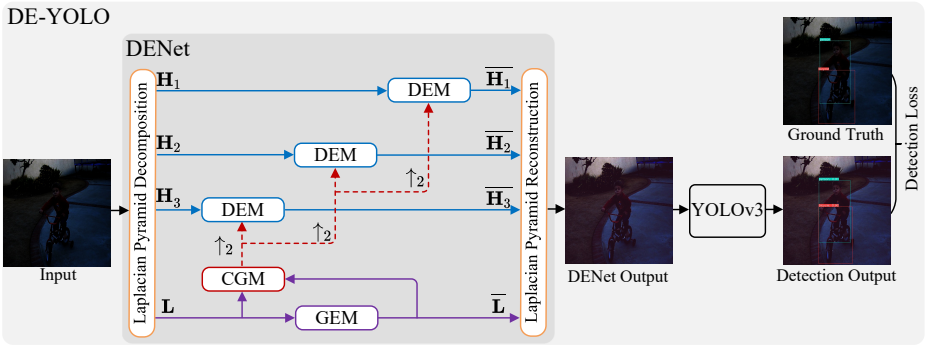


Fig. 1: Architecture of the proposed framework (in our setting, the number of decomposition levels in Laplacian pyramid is 4).

components. The LF component and the HF component at the ith decomposition level in Laplacian pyramid ($1 \leq i < N$) are respectively calculated by

$$\mathbf{L} = G_N(\mathbf{I}) \tag{1}$$

$$\mathbf{H}_i = G_i(\mathbf{I}) - B(G_{i+1}(\mathbf{I}) \uparrow_2) \tag{2}$$

where N is the total number of decomposition levels; $B(.)$ denotes blurring the input by using a 2D Gausssian kernel with a size of 5×5; $\uparrow_2$ stands for up-sampling an image by a factor of 2; $G_i(\mathbf{I}) \in \mathbb{R}^{\frac{h}{2^{i-1}} \times \frac{w}{2^{i-1}} \times 3}$ represents the ith level of image in Gaussian pyramid [36], which can be defined by

$$G_i(\mathbf{I}) = \begin{cases} \mathbf{I}, & i = 1 \\ B(G_{i-1}(\mathbf{I})) \downarrow_2, & 2 \leq i \leq N \end{cases} \tag{3}$$

where $\downarrow_2$ denotes down-sampling an image by a factor of 2. From Equs. (1)$\sim$(3), it is obvious that the decomposition is fully reversible.

As can be observed from Equ. (3), the image at the Nth level in Gaussian pyramid has been blurred $N-1$ times and has the lowest resolution. Thus $\mathbf{L}$ in Laplacian pyramid is an LF component, which is likely to contain global illumination and large-scale structure. On the other hand, according to Equ. (2), $\mathbf{H}_i$ consists of HF residual details and has a larger resolution. From a high decomposition level to a low decomposition level, coarse to fine levels of image details are respectively stored in $\{\mathbf{H}_i\}$.

By taking advantage of Laplacian pyramid decomposition and reconstruction, a lightweight but very effective DENet is proposed. Since the LF component in Laplacian pyramid reveals global illumination, we design a GEM (Sec. 3.2) in DENet to improve the contrast and restore the visibility in the LF component. Note that the LF component in Laplacian pyramid has a small resolution, leading to a low computational burden in GEM. Thus using Laplacian pyramid decomposition and building GEM for the LF component is beneficial to the detection speed of DE-YOLO. When enhancing the global contrast/illumination, it

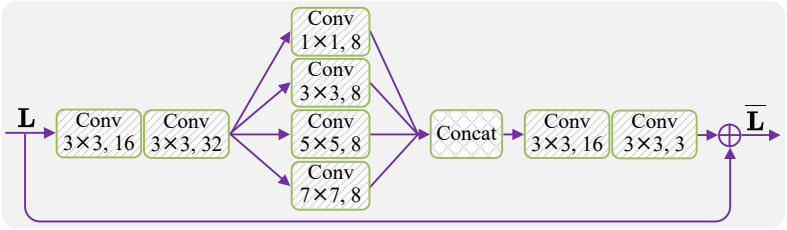


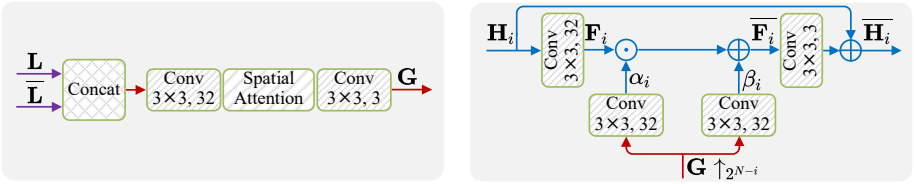
Fig. 2: The structure of GEM. "Conv 3×3, 32" stands for a convolutional layer with a kernel size of 3×3 and 32 output channels. For simplicity, the activation function LReLU is omitted.

is also necessary to enhance the local details accordingly. We notice that the HF components in Laplacian pyramid contain coarse-to-fine local details and those details are highly related to the LF component. Therefore, a DEM (Sec. 3.3) is deployed at each HF level to efficiently and effectively enhance the local details by incorporating the guidance information extracted from a cross-level guidance module (CGM) (Sec. 3.3). Finally, the enhanced LF and HF components are used to progressively reconstruct the enhanced image.

3.2 Global Enhancement Module for the LF Component

The structure of GEM is shown in Fig. 2. Unlike the common low-level vision tasks, the goal of our DENet is not to obtain an enhanced image which is close to the clean GT for human eyes. Thus there is no need to establish a sophisticated mapping from the low-quality image domain to the GT domain. This enables the structure of GEM to be simple enough.

In the front end of GEM, two convolutional layers are first used to extract features from the LF component with a dimension of $\frac{h}{2^{N-1}} \times \frac{w}{2^{N-1}} \times 3$. As GEM is built to enhance the global structure and contrast/illumination in images, it is reasonable to use different sizes of kernels to well capture multi-scale information, which is similar to the idea of the well-known Inception architecture [37]. Here we use four parallel convolutions with 1×1, 3×3, 5×5 and 7×7 filters, respectively. Since the resolution of the LF component is rather small, a kernel size of 7×7 is enough to cover a very large region in the original image. Therefore, environment-specific knowledge such as the lighting condition or the fog spreading over the whole image can be well depicted. To further reduce the computational complexity and the number of parameters, the output features of each parallel path are compressed to 8 channels. Through experiments we found that such a lightweight setting still leads to satisfactory detection results. Afterwards, the features from four parallel paths are concatenated and further fused by two 3×3 convolutional layers. To improve the performance of GEM, skip connection is applied, so that this structure can focus on learning the residual between the input LF component $\mathbf{L}$ and the corresponding enhanced output $\overline{\mathbf{L}}$.



(a) CGM (b) DEM

Fig. 3: The structures of CGM and DEM. The activation function LReLU is omitted.

3.3 Detail Enhancement Module for the HF Components

To enhance the HF components which contain coarse-to-fine local details in Laplacian pyramid, DEM and CGM are established, the structures of which are given in Fig. 3 (a) and (b), respectively.

CGM is used to extract the guidance information from the LF component. To embed the differences between the unprocessed component $\mathbf{L}$ and the enhanced component $\overline{\mathbf{L}}$, both $\mathbf{L}$ and $\overline{\mathbf{L}}$ are fed to CGM. In the front end of CGM, $\mathbf{L}$ and $\overline{\mathbf{L}}$ are concatenated. Then a 3×3 convolutional layer extracts 32 feature maps from the concatenated two components. Since the LF component is spatially correlated to the HF components, a spatial attention module [38] is utilized to localize the positions where LF and HF components are highly correlated. Finally, another 3×3 convolutional layer is used to further refine the 32 feature maps and generate the guidance $\mathbf{G}$ with a dimension of $\frac{h}{2^{N-1}} \times \frac{w}{2^{N-1}} \times 3$.

DEM is utilized to enhance the HF components under the guidance provided by CGM. Since the resolutions of the LF and each HF component are different, before entering DEM, the cross-level guidance information is upsampled by using bilinear interpolation. Note that the resolution of the HF component becomes larger as the decomposition level goes lower. As a result, building a sophisticated enhancement module for each HF component could induce intense computation, which significantly reduces detection speed. To efficiently and effectively enhance the HF components, we use a simple residual block, and apply affine transformation [39] to incorporate the guidance information into the extracted HF features. The affine transformation is defined as

$$\mathcal{M}(\mathbf{F}_i \mid \alpha_i, \beta_i) = \alpha_i \odot \mathbf{F}_i + \beta_i \tag{4}$$

where $\mathbf{F}_i$ stands for the extracted HF feature; $\odot$ denotes the element-wise multiplication; α_i and β_i are the scaling and shifting parameters at the ith decomposition level, respectively, which are learned by feeding the upsampled guidance information $\mathbf{G}$ to two different 3×3 convolutional layers.

4 Experiments and Analysis

4.1 Implementation Details

In our experiment, to facilitate a fair and comprehensive evaluation, the classical YOLOv3 [7] is applied as the detector. The image size for training and testing

is 544×544. During training, data augmentations such as random filp, random scale and HSV augment are used. The batch size is set to 8 and the initial learning rate is 10^{-4}. Adam optimizer [40] with Cosine learning rate schedule is used to train DE-YOLO for 150 epochs, and the early stopping strategy is used to avoid overfitting. The proposed DE-YOLO is implemented with the Pytorch framework, and all the experiments are carried out on a single NVIDIA GeForce RTX 2080 Ti GPU.

4.2 Preparation of Datasets

The low-light and foggy weather conditions are evaluated, and the used datasets are summarized in Table 1. For low-light condition, exclusively dark (*ExDark*) [41] is used, which contains 7363 low-light images, where 12 object categories for detection are annotated. For foggy weather, the real-world task-driven testing set (*RTTS*) is chosen. It consists of 4322 images captured under real-world foggy condition, and 5 object categories are annotated. Besides, the unannotated realistic hazy images (*URHI*) dataset which contains 4807 unannotated natural hazy images is also used for training the UDA-based methods. Both *RTTS* and *URHI* are subsets in the *RESIDE* dataset [42].

Moreover, the well-known dataset *PASCAl VOC* [9] is used to generate synthetic low-light and foggy images. Note that although each synthetic degraded image has a corresponding GT high-quality version, the GT images are not used for training IA-YOLO and our DE-YOLO. To obtain synthetic low-light images, the original RGB images in *VOC* are degraded by gamma transformation:

$$g(\mathbf{I}) = \mathbf{I}^{\gamma} \tag{5}$$

For each image in *VOC*, γ is randomly selected from a range of [1.5, 5]. However, only 10 out of 20 categories in *VOC* dataset match with the 10 categories in *ExDark* dataset. Therefore, we first filter out the unmatched categories in *VOC*, and obtain two sub-sets called *VOC_train_I* and *VOC_test_I* for training and testing, respectively. Then all the images in *VOC_test_I* are degraded by Equ. (5), resulting in a synthetic low-light testing set *VOC_lowlight_test*. By randomly degrading 2/3 images in *VOC_train_I*, a hybrid training set *VOC_hybrid_train_I* is built.

Similarly, to generate synthetic foggy images, we build *VOC_train_II* and *VOC_test_II* by filtering out the *VOC* categories that do not match with *RTTS* dataset. Based on the atmospheric scattering model [43–45], the foggy datasets *VOC_hybrid_train_II* and *VOC_foggy_test* are obtained by:

$$\mathbf{J} = \mathbf{I}e^{-\lambda}\mathbf{d} + A(1 - e^{-\lambda}\mathbf{d}) \tag{6}$$

where $\mathbf{J}$ denotes the synthetic foggy image, A is the global atmospheric light and is set as 0.5 in our experiment, $\lambda = 0.05 + 0.01 * k$, k is a random integer number which ranges from 0 to 9. The scene depth of a pixel is computed by $d = -0.04 \times \rho + \sqrt{\max(h, w)}$, with ρ denoting the Euclidean distance from the current pixel to the central one, h and w being the height and width of the target image, respectively.

Table 1: Overview of the used datasets. NLI and LLI are short for normal-light and low-light images, respectively.

Dataset	Type	Images	Instances	Categories
VOC_train_I	NLI	12334	29135	10
VOC_test_I	NLI	3760	8939	10
VOC_hybrid_train_I	NLI + synthetic LLI	12334	29135	10
VOC_lowlight_test	synthetic LLI	3760	8939	10
ExDark (training)	realistic LLI	4800	-	-
ExDark (testing)	realistic LLI	2563	6450	10
VOC_train_II	fog-free	8111	19561	5
VOC_test_II	fog-free	2734	6604	5
VOC_hybrid_train_II	fog-free + synthetic foggy	8111	19561	5
VOC_foggy_test	synthetic foggy	2734	6604	5
URHI (training)	realistic foggy	4807	-	-
RTTS (testing)	realistic foggy	4322	29577	5

4.3 Object Detection on Low-light Images

For the low-light condition, DSNet, IA-YOLO and proposed DE-YOLO is trained on *VOC_hybrid_train_I*. The *Baseline* methods YOLOv3(N) and YOLOv3(L) are obtained by training the normal YOLOv3 model [7] on *VOC_train_I* and *VOC_hybrid_train_I*, respectively. The LLIE methods, including MBLLEN [26], KinD [22], EnlightenGAN [25] and Zero-DCE [27], are used to preprocess the low-light images before applying YOLOv3(N) for detection. The pre-trained models of the four LLIE methods provided by their authors are directly applied. For the UDA-based methods, MS-DAYOLO [16] and DAYOLO [17] are re-trained on *VOC_train_I* on the source domain with labels and the training set of *ExDark* on the target domain without labels. Since a synthetic low-light dataset is proposed together with MAET in [21], MAET is trained on its own synthetic low-light dataset.

Table 2 shows comparisons among different methods on three testing datasets. The performance is evaluated by using the mean average precision (mAP) at an intersection over union (IoU) threshold of 0.5 (mAP50). From Table 2 we have the following observations: 1) Training YOLOv3(L) on the hybrid dataset achieves better performance than YOLOv3(N) on low-light testing datasets. However, when testing on normal-light dataset *VOC_test_I*, YOLOv3(L) is worse than YOLOv3(N), which suggests that YOLOv3(L) cannot be well generalized from one dataset to another. 2) Simply using the four LLIE methods (MBLLEN, KinD, EnlightenGAN, Zero-DCE) before YOLOv3 cannot significantly improve the performance of YOLOv3(N). 3) The two UDA-based methods MS-DAYOLO and DAYOLO bring limited improvements over YOLOv3(N) on two low-light testing datasets. As the UDA-based methods are trained on the target domain without labels, the performance is less satisfactory when the domain gap is large. 4) The two MTL-based methods achieve higher mAP than

Table 2: Performance comparisons on low-light images (mAP50 (%)).

Method		VOC_test_I	VOC_lowlight_test	ExDark
Baseline	YOLOv3(N) [7]	72.22	56.34	43.02
	YOLOv3(L) [7]	66.95	62.91	45.58
LLIE	MBLLEN [26]	-	58.36	43.49
	KinD [22]	-	52.57	39.22
	EnlightenGAN [25]	-	53.67	39.42
	Zero-DCE [27]	-	56.49	40.40
UDA	MS-DAYOLO [16]	72.01	58.20	44.25
	DAYOLO [17]	71.58	58.82	44.62
MTL	DSNet [20]	61.82	64.57	45.31
	MAET [21]	69.49	58.23	47.10
JED	IA-YOLO [35]	72.53	67.34	49.43
	DE-YOLO (ours)	**73.17**	**67.81**	**51.51**

UDA-based and LLIE-based approaches on the *ExDark* dataset, yet fall behind UDA-based approaches on the normal-light dataset *VOC_test_I*. 5) The two JED-based approaches are superior to other types of methods, and the proposed DE-YOLO yields the best performance among all the competing methods on all the three datasets. Particularly, on realistic low-light dataset *ExDark*, DE-YOLO surpasses YOLOv3(L) and the second best method IA-YOLO by 5.93% and 2.08%, respectively. Moreover, DE-YOLO achieves a even better performance than YOLOv3(N) on normal-light images, which well demonstrates the generalization ability of DE-YOLO.

The detection results obtained by different methods are visualized in Figs. 4 and 5. For the GT results, we just show the GT labels on the original low-light images. Although the LLIE methods and the JED method IA-YOLO are able to enhance the brightness of images, significant noises and artifacts can be observed from their results. On the contrary, DENet can suitably enhance the underexposed regions in images, while suppressing different kinds of artifacts and noises. Therefore, the detection results of DE-YOLO have fewer false-positive and false-negative results.

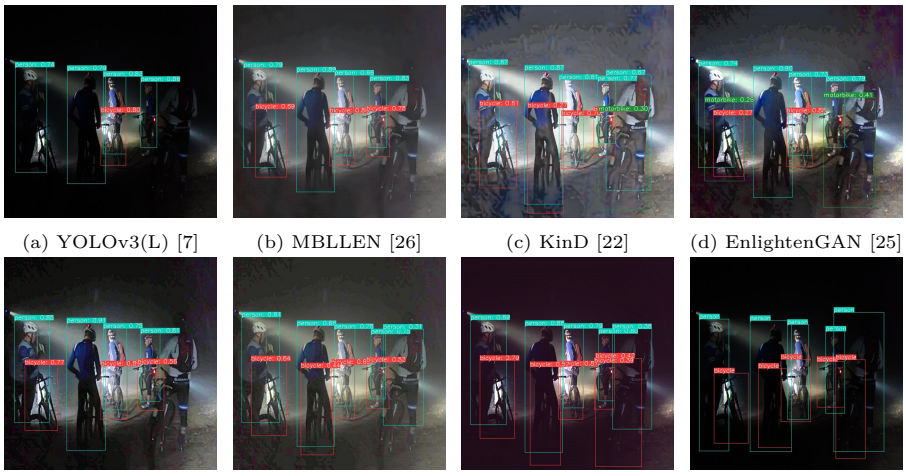
4.4 Object Detection on Foggy Images

Table 3 presents the detection results on foggy images. Similar to the settings on low-light images, YOLOv3(N) and YOLOv3(F) are the YOLOv3 [7] models trained on *VOC_train_II* and *VOC_hybrid_train_II*, respectively. Three representative defogging methods including GridDehazeNet [29], DCPDN [31] and MSBDN [30] are adopted to pre-process the foggy images before detection. The two UDA-based methods MS-DAYOLO [16] and DAYOLO [17] are re-trained on *VOC_train_II* on the source domain with labels and *URHI* on the target domain without labels. The MTL-based method DSNet [20] and the JED-based methods IA-YOLO [35] and DE-YOLO are re-trained on *VOC_hybrid_train_II*.

(a) YOLOv3(L) [7] (b) MBLLEN [26] (c) KinD [22] (d) EnlightenGAN [25]

(e) Zero-DCE [27] (f) IA-YOLO [35] (g) DE-YOLO (ours) (h) GT

Fig. 4: Detection results obtained on image *2015_00402* from *ExDark*. MBLLEN, KinD, Zero-DCE, EnlightenGAN and IA-YOLO all generate significant noises. Although DE-YOLO is darker than the other methods, the image is clear and the contrast is suitable.



(a) YOLOv3(L) [7] (b) MBLLEN [26] (c) KinD [22] (d) EnlightenGAN [25]

(e) Zero-DCE [27] (f) IA-YOLO [35] (g) DE-YOLO (ours) (h) GT

Fig. 5: Detection results obtained on image *2015_00542* from *ExDark*. Compared with other methods, DE-YOLO well suppresses artifacts and enhances the contrast of image. Besides, DE-YOLO is the only method that properly detects all the objects.

It can be seen from Table 3 that: 1) Although GridDehazeNet, DCPDN and MSBDN are worse than YOLOv3(F), applying the three defogging methods significantly improve the performance of YOLOv3(N). 2) Similar to the low-light

Table 3: Performance comparisons on foggy images (mAP50 (%)).

Method		VOC_test_II	VOC_foggy_test	RTTS
Baseline	YOLOv3(N) [7]	83.41	46.53	41.87
	YOLOv3(F) [7]	79.06	78.87	49.45
Defogging	GridDehazeNet [29]	-	72.09	46.03
	DCPDN [31]	-	73.38	44.43
	MSBDN [30]	-	72.04	45.90
UDA	MS-DAYOLO [16]	81.69	65.44	42.94
	DAYOLO [17]	80.12	66.53	44.15
MTL	DSNet [20]	71.49	81.71	49.86
JED	IA-YOLO [35]	84.05	83.22	52.36
	DE-YOLO (ours)	**84.13**	**83.56**	**53.70**

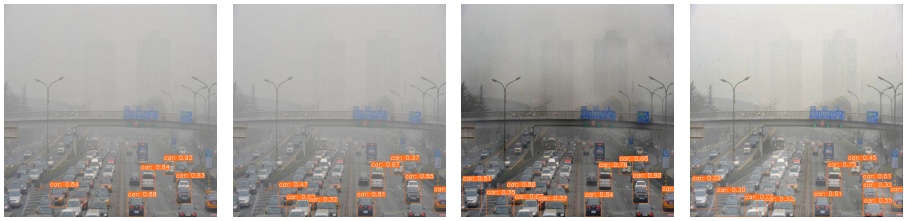
condition, the UDA-based methods are better than YOLOv3(N), but they are worse than YOLOv3(F) on the two foggy datasets. 3) Compared with the defogging and the UDA-based methods, DSNet achieves higher mAP values on the two foggy datasets. However, it suffers from a drop in mAP on the fog-free dataset *VOC_test_II*. 4) The proposed DE-YOLO achieves the best detection results on all the three testing sets. In particular, on the realistic foggy dataset *RTTS*, DE-YOLO provides mAP values 4.25% and 1.34% higher than YOLOv3(F) and IA-YOLO, respectively.

The qualitative comparisons among different methods on the image from *RTTS* are given in Fig. 6. For the GT result, the GT labels are directly showed on the original hazy image. We can find that DE-YOLO is able to deliver images with suitable contrast, which helps to increase the number of true-positive results and the confidences of the detected objects.

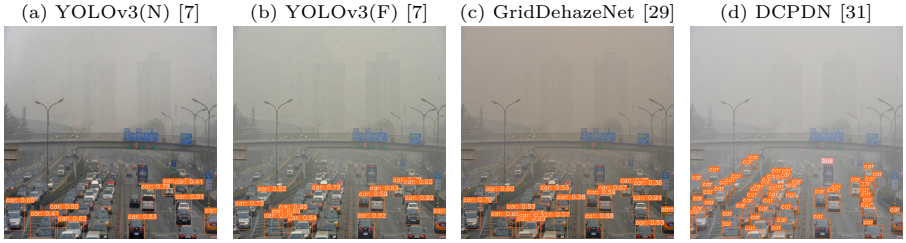
4.5 Ablation Study

Table 4 compares the contributions of GEM, DEM and CGM in DENet, and the results are reported on *ExDark*. Note that for the variant model which does not have CGM, the affine transformation in DEM is removed. Therefore, such a variant of DEM becomes a normal residual block. As can be seen, firstly, applying GEM leads to a mAP improvement of 2.79% over the normal YOLOv3(L). Secondly, additionally using DEM results in a even better performance. Finally, the best performance can be achieved by utilizing all the three modules, which demonstrates that these modules are complementary to each other.

In Table 5, the effects of the number of levels of Laplacian pyramid in DENet is evaluated. Note that when $N = 1$, Laplacian pyramid decomposition is not applied, and only GEM is used to enhance the input image. It is obvious that a larger number of decomposition levels results in more parameters and longer runtime. We find that the mAP50 increases monotonically with N when $N \leq 4$, and the performance decreases when $N > 4$.



(a) YOLOv3(N) [7] (b) YOLOv3(F) [7] (c) GridDehazeNet [29] (d) DCPDN [31]



(e) MSBDN [30] (f) IA-YOLO [35] (g) DE-YOLO (ours) (h) GT

Fig. 6: Detection results obtained by different methods on image *BD_ Baidu_074* from *RTTS*. Our DE-YOLO detects more vehicles than the other methods.

Table 4: Ablation analysis on different modules of our method. Note that only the parameters of DENet are measured.

Method	GEM	DEM	CGM	Parameters (K)	mAP50 (%)
YOLOv3 (L)	-	-	-	-	45.58
DE-YOLO	✓	×	×	32	48.37
	✓	✓	×	35	49.80
	✓	✓	✓	45	**51.51**

Table 5: The effects of the number of levels of Laplacian pyramid. The runtime is tested on a single RTX 2080Ti GPU with an image size of 544×544. Note that the reported numbers of parameters and runtime values are only measured on DENet.

N	Parameters (K)	Runtime (ms)	mAP50 (%)
1	32	3	46.28
2	38	4	50.09
3	42	5	51.27
4	45	6	**51.51**
5	49	7	51.16
6	51	8	51.01

4.6 Efficiency Analysis

Table 6 lists the number of parameters and runtime consumed by different enhancement-based methods. Note that during testing, the UDA-based and

Table 6: The comparison of efficiency. The runtime is tested on a single RTX 2080Ti GPU with an image size of 544×544. Note that the reported numbers of parameters and runtime values do not include those required by the YOLOv3 model.

Method		Parameters	Runtime (ms)
LLIE	MBLLEN [26]	450K	72
	KinD [22]	8M	28
	Zero-DCE [27]	79K	7
	EnlightenGAN [25]	9M	15
Defogging	GridDehazeNet [29]	985K	49
	DCPDN [31]	67M	30
	MSBDN [30]	31M	51
JED	IA-YOLO [35]	165K	9
	DENet (ours)	**45K**	**6**

MTL-based methods do not require extra parameters and computation. Therefore, they are excluded from Table 6 for comparison. Since both IA-YOLO and DE-YOLO include a normal YOLOv3 model, only the image enhancement subnetworks are compared. It can be found from Table 6 that the proposed DENet has a very small number of parameters and requires the shortest runtime. Such a lightweight and fast model is suitable for real-time applications.

5 Conclusion

To enable a faithful detection under adverse weather conditions, an adaptive image enhancement model DENet was presented. In DENet, the input image is decomposed by using Laplacian pyramid. After that, the LF component and HF components are respectively enhanced. To explore the correlation among different components, cross-level guidance information is extracted from the LF component and then incorporated into the features of the HF components. DENet is extremely lightweight, and thus is suitable for the applications that require real-time detection. By training DENet and a common YOLOv3 model in an end-to-end manner, a JED framework DE-YOLO was obtained. Experiments showed that DE-YOLO is able to achieve the highest mAP50 value among all the compared methods under the low-light and foggy conditions. Meanwhile, under the normal condition with clean input images, the performance of DE-YOLO is also better than the original YOLOv3, which suggests that the proposed method has a good generalization ability.

Acknowledgements This work was supported in part by the National Natural Science Foundation of China (NSFC) (62171145, 61761005), and in part by Guangxi Key Laboratory of Multimedia Communications and Network Technology. Part of the experiments were carried out on the High-performance Computing Platform of Guangxi University.

References

1. Girshick, R.B., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). pp. 580–587. Columbus, OH, USA (Jun 2014)
2. Girshick, R.B.: Fast R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 1440–1448. Santiago, Chile (Dec 2015)
3. Ren, S., He, K., Girshick, R.B., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems (NIPS). pp. 91–99. Montreal, Quebec, Canada (Dec 2015)
4. He, K., Gkioxari, G., Dollár, P., Girshick, R.B.: Mask R-CNN. In: Proceedings of the IEEE international conference on computer vision (ICCV). pp. 2980–2988. Venice, Italy (Oct 2017)
5. Redmon, J., Divvala, S.K., Girshick, R.B., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). pp. 779–788. Las Vegas, NV, USA (Jun 2016)
6. Redmon, J., Farhadi, A.: Yolo9000: Better, faster, stronger. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). pp. 6517–6525. Honolulu, HI, USA (Jul 2017)
7. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767 (2018)
8. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: Proceedings of European Conference on Computer Vision (ECCV). pp. 21–37. Amsterdam, Netherlands (Oct 2016)
9. Everingham, M., Gool, L.V., Williams, C.K.I., Winn, J.M., Zisserman, A.: The pascal visual object classes (VOC) challenge. Int. J. Comput. Vis. **88**(2), 303–338 (2010)
10. Lin, T., Maire, M., Belongie, S.J., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: Common objects in context. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 740–755. Zurich, Switzerland (Sep 2014)
11. Chen, Y., Li, W., Sakaridis, C., Dai, D., Gool, L.V.: Domain adaptive faster R-CNN for object detection in the wild. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). pp. 3339–3348. Salt Lake City, UT, USA (Jun 2018)
12. Zhu, X., Pang, J., Yang, C., Shi, J., Lin, D.: Adapting object detectors via selective cross-domain alignment. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). pp. 687–696. Long Beach, CA, USA (Jun 2019)
13. Wang, T., Zhang, X., Yuan, L., Feng, J.: Few-shot adaptive faster R-CNN. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). pp. 7173–7182. Long Beach, CA, USA (Jun 2019)
14. Saito, K., Ushiku, Y., Harada, T., Saenko, K.: Strong-weak distribution alignment for adaptive object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). pp. 6956–6965. Long Beach, CA, USA (Jun 2019)
15. He, Z., Zhang, L.: Multi-adversarial faster-rcnn for unrestricted object detection. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 6667–6676. Seoul, Korea (South) (Nov 2019)

16. Hnewa, M., Radha, H.: Multiscale domain adaptive yolo for cross-domain object detection. In: Proceedings of the IEEE International Conference on Image Processing (ICIP). pp. 3323–3327. Anchorage, AK, USA (Sep 2021)
17. Zhang, S., Tuo, H., Hu, J., Jing, Z.: Domain adaptive YOLO for one-stage cross-domain detection. In: Proceedings of the Asian Conference on Machine Learning (ACML). pp. 785–797. Virtual Event (Nov 2021)
18. Sindagi, V.A., Oza, P., Yasarla, R., Patel, V.M.: Prior-based domain adaptive object detection for hazy and rainy conditions. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 763–780. Glasgow, UK (Aug 2020)
19. Sasagawa, Y., Nagahara, H.: YOLO in the dark - domain adaptation method for merging multiple models. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 345–359. Glasgow, UK (Aug 2020)
20. Huang, S., Le, T., Jaw, D.: DSNet: Joint semantic learning for object detection in inclement weather conditions. IEEE Transactions on Pattern Analysis and Machine Intelligence **43**(8), 2623–2633 (2021)
21. Cui, Z., Qi, G., Gu, L., You, S., Zhang, Z., Harada, T.: Multitask AET with orthogonal tangent regularity for dark object detection. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 2533–2542. Montreal, QC, Canada (Oct 2021)
22. Zhang, Y., Zhang, J., Guo, X.: Kindling the darkness: A practical low-light image enhancer. In: Proceedings of the ACM International Conference on Multimedia (MM). pp. 1632–1640. Nice, France (Oct 2019)
23. Wei, C., Wang, W., Yang, W., Liu, J.: Deep retinex decomposition for low-light enhancement. In: Proceedings of the British Machine Vision Conference (BMVC). p. 155. Newcastle, UK (Sep 2018)
24. Wu, W., Weng, J., Zhang, P., Wang, X., Yang, W., Jiang, J.: Uretinex-net: Retinex-based deep unfolding network for low-light image enhancement. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5901–5910. New Orleans, LA, USA (June 2022)
25. Jiang, Y., Gong, X., Liu, D., Cheng, Y., Fang, C., Shen, X., Yang, J., Zhou, P., Wang, Z.: EnlightenGAN: Deep light enhancement without paired supervision. IEEE Trans. Image Process. **30**, 2340–2349 (2021)
26. Lv, F., Lu, F., Wu, J., Lim, C.: MBLLEN: Low-light image/video enhancement using cnns. In: Proceedings of the British Machine Vision Conference (BMVC). p. 220. Newcastle, UK (Sep 2018)
27. Guo, C., Li, C., Guo, J., Loy, C.C., Hou, J., Kwong, S., Cong, R.: Zero-reference deep curve estimation for low-light image enhancement. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). pp. 1777–1786. Seattle, WA, USA (Jun 2020)
28. Ma, L., Ma, T., Liu, R., Fan, X., Luo, Z.: Toward fast, flexible, and robust low-light image enhancement. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5637–5646. New Orleans, LA, USA (June 2022)
29. Liu, X., Ma, Y., Shi, Z., Chen, J.: Griddehazenet: Attention-based multi-scale network for image dehazing. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 7313–7322. Seoul, Korea (South) (Nov 2019)
30. Dong, H., Pan, J., Xiang, L., Hu, Z., Zhang, X., Wang, F., Yang, M.: Multi-scale boosted dehazing network with dense feature fusion. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). pp. 2154–2164. Seattle, WA, USA (Jun 2020)

31. Zhang, H., Patel, V.M.: Densely connected pyramid dehazing network. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). pp. 3194–3203. Salt Lake City, UT, USA (Jun 2018)
32. Yang, Y., Wang, C., Liu, R., Zhang, L., Guo, X., Tao, D.: Self-augmented unpaired image dehazing via density and depth decomposition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2037–2046. New Orleans, LA, USA (June 2022)
33. Burt, P.J., Adelson, E.H.: The laplacian pyramid as a compact image code. IEEE Transactions on Communications **31**(4), 532–540 (1983)
34. Liu, J., Xu, D., Yang, W., Fan, M., Huang, H.: Benchmarking low-light image enhancement and beyond. International Journal of Computer Vision **129**(4), 1153–1184 (2021)
35. Liu, W., Ren, G., Yu, R., Guo, S., Zhu, J., Zhang, L.: Image-adaptive yolo for object detection in adverse weather conditions. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI). vol. 36, pp. 1792–1800 (2022)
36. Adelson, E.H., Anderson, C.H., Bergen, J.R., Burt, P.J., Ogden, J.M.: Pyramid methods in image processing. RCA engineer **29**(6), 33–41 (1984)
37. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S.E., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). pp. 1–9. Boston, MA, USA (Jun 2015)
38. Woo, S., Park, J., Lee, J.Y., Kweon, I.S.: Cbam: Convolutional block attention module. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 3–19. Munich, Germany (September 2018)
39. Wang, X., Yu, K., Dong, C., Change Loy, C.: Recovering realistic texture in image super-resolution by deep spatial feature transform. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR). pp. 606–615. Salt Lake City, UT, USA (Jun 2018)
40. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Proceedings of the International Conference on Learning Representations (ICLR). San Diego, CA, USA (May 2015)
41. Loh, Y.P., Chan, C.S.: Getting to know low-light images with the exclusively dark dataset. Computer Vision and Image Understanding **178**, 30–42 (2019)
42. Li, B., Ren, W., Fu, D., Tao, D., Feng, D., Zeng, W., Wang, Z.: Benchmarking single-image dehazing and beyond. IEEE Transactions on Image Processing **28**(1), 492–505 (2019)
43. McCartney, E.J.: Optics of the atmosphere: scattering by molecules and particles. New York (1976)
44. Nayar, S.K., Narasimhan, S.G.: Vision in bad weather. In: Proceedings of the seventh IEEE international conference on computer vision. vol. 2, pp. 820–827. IEEE (1999)
45. Narasimhan, S.G., Nayar, S.K.: Contrast restoration of weather degraded images. IEEE transactions on pattern analysis and machine intelligence **25**(6), 713–724 (2003)