Edge Computing Driven Low-Light Image Dynamic Enhancement for Object Detection

Yirui Wu, *Member, IEEE*, Haifeng Guo, Chinmay Chakraborty, *Senior Member, IEEE*, Mohammad R. Khosravi, Stefano Berretti, *Senior Member, IEEE*, and Shaohua Wan, *Senior Member, IEEE*

***Abstract*—With fast increase in volume of mobile multimedia data, how to apply powerful deep learning methods to process data with real-time response becomes a major issue. Meanwhile, edge computing structure helps improve response time and user experience by bringing flexible computation and storage capabilities. Considering both technologies for successful AI-based applications, we propose an edge-computing driven and end-to-end framework to perform tasks of image enhancement and object detection under low-light conditions. The framework consists of a cloud-based enhancement and an edge-based detection stage. In the first stage, we establish connections between edge devices and cloud servers to input re-scaled illumination parts of low-light images, where enhancement subnetworks are dynamically and parallel coupled to compute enhanced illumination parts based on low-light context. During the edge-based detection stage, edge devices could accurately and rapidly detect objects based on cloud-computed informative feature map. Experimental results show the proposed method significantly improves detection performance in low-light conditions with low latency running on edge devices.**

***Index Terms*—Low-light image enhancement, object detection, edge-driven deep learning method.**

I. Introduction

As MORE cameras are applied to acquire images and videos from real-life, how to efficiently analyze multimedia data becomes a hot topic. Motivated by highly distinguish capability of deep neural networks (DNN), researchers have successfully deployed DNN based applications on powerful PCs [1], [2]. Among multimedia data analytics applications, object detection plays a pivotal role by identifying and localizing all objects instances with category labels in an input image, which is the major concerned topic in this paper.

In many multimedia analytics scenarios, simply uploading and handling data at the cloud end leads to annoying user experience, especially considering time-sensitive property of mobile computing. The edge computing paradigm thus appears to provide efficient distributed computing resources at the edge of networks, posing significant challenges on running deep learning methods for multimedia data analytics [3]–[5].

Since prices of high-resolution cameras have been falling off recently, object detection networks appear with larger capacity with deeper layers or higher dimension of feature map to fulfill the requirements of high-resolution inputs. The inherent complexity of DNNs even largely increases by the unexpected captured conditions, such as conditions of low light, low resolution and color distortion, resulting in a prohibitively high latency when detecting objects on edge devices. Considering limited computing and storage resource in edge devices, the tension between resources-constrained edge devices and compute-intensive inference workloads becomes the major challenge to deploy DNNs for object detection task [6]–[8].

To fill the gap between the needs of object detection in low-light environment, and of limited computation resource in edge devices, we propose an edge computing driven framework, which simultaneously performs the tasks of low-light image enhancement and object detection with the support of the edge computing infrastructure. The proposed method strikes the balance between edge resources and object detection performance by enhancing high-resolution images in cloud and running object detection network on intermediate computations in edge devices. Unlike existing solutions that run one processing network on input images [9]–[11], the proposed method adopts multiple parallel subnetworks to compute enhanced images that tend to be more informative, hence resulting in a higher detection accuracy.

The design of the proposed method faces two core technical challenges. First, the overall structure design is non-trivial, considering the aim of to best balancing the trade-off between accuracy and computing requirement. Second, how to deal with complexity brought by low-light conditions, requiring the awareness of the variations to design specific DNNs for enhancement purpose is often ignored by popular object detection networks. To cope with the first issue, the proposed method facilitates as many as possible subnetworks

Manuscript received 26 August 2021; revised 14 January 2022; accepted 6 February 2022. Date of publication 14 February 2022; date of current version 20 September 2023. This work was supported in part by the National Key R&D Program of China under Grant 2021YFB3900601 and in part by the National Natural Science Foundation of China under Grant 62172438. Recommended for acceptance by Dr. Octavia Dobre. *(Corresponding author: Shaohua Wan.)*

Yirui Wu is with the College of Computer and Information, Hohai University, Nanjing 210093, China (e-mail: wuyirui@hhu.edu.cn).

Haifeng Guo is with the National Key Lab for Novel Software Technology, Nanjing University, Nanjing 210093, China (e-mail: guo-haifeng@outlook.com).

Chinmay Chakraborty is with the Department of Electronics and Communication Engineering, Birla Institute of Technology, Ranchi 814142, India (e-mail: cchakrabarty@bitmesra.ac.in).

Mohammad R. Khosravi is with the Shiraz University of Technology, Shiraz, Iran (e-mail: m.r.khosravi.taut@gmail.com).

Stefano Berretti is with the Department of Information Engineering (DINFO), University of Florence, 50139 Florence, Italy (e-mail: stefano.berretti@unifi.it).

Shaohua Wan is with Shenzhen Institute for Advanced Study, University of Electronic Science and Technology of China, Shenzhen 518110, China (e-mail: shaohua.wan@ieee.org).

Digital Object Identifier 10.1109/TNSE.2022.3151502

2327-4697 © 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Hohai University Library. Downloaded on November 22,2023 at 02:01:04 UTC from IEEE Xplore. Restrictions apply.

Fig. 1. Diversity examples of low-light conditions taken by mobile devices, where (a) and (b) correspond to extremely low light conditions, (c) and (d) refer to weak illumination conditions with blurred or foggy scenes, (e) and (f) represent objects in darkness with point light source, (g) and (h) show objects in shadows with too bright scenes.

in resource-abundant cloud to enhance extremely low-light images, and designs two detection networks to match with edge devices owning different resources.

To deal with complexity of low-light conditions, we firstly analyze effects by representing samples in Fig. 1 to reveal cases of blurred or foggy conditions, dark objects with point light source, objects in shadows. General thoughts to deal with low-light conditions are to manually design sequential filters for enhancement before detection, resulting in two drawbacks. Firstly, evaluation metrics for low-night image enhancement and object detection are inconsistent, where we generally adopt measurements of image quality like SSIM/PSNR for enhancement and label correctness like Precision/Recall for detection. The proposed method designs an end-to-end paradigm to solve the problem of inconsistent evaluation metric, where we train the whole network by connecting enhancement and detection stages, and enabling gradient propagation between stages, achieving an unified evaluation metric, *i.e.*, label correctness. During testing, enhancement and detection stages are separately performed on cloud and edge devices, making the best of computing resources by performing time-consuming enhancement in cloud and light-weight detection in edge devices.

Secondly, sequential filters may interrelate, like denoising filter may simultaneously blur edges and texture of objects to be detected, and is time-consuming on high-resolution images with resource-constrained edge devices. The proposed method adopts multiple and parallel subnetworks for enhancement in cloud, which not only solves filter interaction by considering specific categories of low-light conditions, but largely reduces computing time by designing parallel processing in cloud. In summary, the main contribution is three-fold:

- We propose an edge computing driven deep learning method for object detection in low-light conditions, which designs the overall structure of cloud-based enhancement and edge-based detection stages to keep balance between edge compute resources and object detection accuracy.
- An end-to-end paradigm for the tasks of image enhancement and object detection is introduced, which not only performs training optimization within a unified measurement metric to improve detection performance, but also moves time-consuming enhancement to cloud, reducing time latency during testing.
- We propose parallel and dynamic enhancement subnetworks for extremely low-light image enhancement in cloud computing stage, which not only saves computing time by parallel running subnetworks, but also involves low-light context information to improve enhancement effects.

II. Related Work

We classify the related existing methods into three topics: low-light image enhancement, edge computing paradigm and object detection.

A. Low-Light Image Enhancement

Since low-light condition could largely decrease performance of semantical comprehension tasks, large amount of image enhancement methods are proposed by researchers to obtain high-quality images from original ones. Traditional methods use either adaptive histogram equalization or Retinex theory to perform enhancement, where the latter methods enhance pixels based on well-designed illumination map. Following the idea of the Retinex theory, Ying *et al.* [12] proposed to enhance low-light images by dynamically adjusting exposure time to create more images at first, and then fusing images with the help of estimated illumination map.

Inspired by significant power in constructing distinguish feature maps, deep learning methods are applied by researchers to complete low-level vision tasks. Following the general steps to process low-level image information, Gharbi *et al.* [13] proposed the HDR-Net with pairwise supervision training, which utilizes thoughts of bilateral grid processing and local affine color transforms to work within structure of deep neural networks. To fuse advantages of deep learning methods and Retinex theory, Wei *et al.* [14] proposed an end-to-end framework, where their proposed Decom-Net is designed for image decomposition and another Enhance-Net is responsible for illumination enhancement based on decomposed information. With the conception of "learning to see in the dark," Chen *et al.* [15] proposed to learn the pipeline of color transformations, demosaicing and denoising, which successfully prevent artifacts during low-light enhancement. They later conducted experiments on several public datasets showing impressive visual effects. Afterwards, Zhu *et al.* [16] proposed the Edge-Enhanced Multi-Exposure Fusion Network (EEMEFN) to deal with extremely low-light image enhancement, which employed a multi-exposure and an edge enhancement fusion module to address the high contrast and color bias issues, and refined the initial images with edge information, respectively. Liu *et al.* [17] proposed the RUAS model to characterize the intrinsic underexposed structure of low-light images to obtain a top-performing image enhancement network. Most recently, Zhang *et al.* [18] enforced the temporal stability in low-light video enhancement with only static images by learning and inferring

Authorized licensed use limited to: Hohai University Library. Downloaded on November 22,2023 at 02:01:04 UTC from IEEE Xplore. Restrictions apply.

motion field (optical flow) from a single image and synthesize short range video sequences.

Due to unsupervised training property, GANs are popular to be applied for image restoration and enhancement, where the GANs structure is powerful in handling a quantity of unpaired images without enough prior knowledge. For example, Zhu *et al.* [19] adopted a two-way GAN with a novel cycle-consistent loss, which could perform image synthesis and translation tasks between two different domains with unpaired data. However, such work is often hard to train and unstable in performance. Differ from complicated structure design, Jiang *et al.* [20] proposed a lightweight and one-path unsupervised GAN named as EnlightenGAN, which designs a global-local discriminator structure and a self-regularized perceptual loss fusion to enhance various real-world images.

B. Edge Computing Paradigm

Internet of Things (IoT) is fast developing, where billions of smart devices are applied on objects to collect information from real-life [21]. However, the cost of hardware equipments restricts the speed of data processing in smart devices, thus causing slow response, time delay and so on. To solve such problem, edge computing emerges to provide real-time services and mitigate the bandwidth demand. In contrast to the idea of processing data in cloud, edge computing tries to process most amount of data at the edge and upload small size of data to cloud, where the uploaded data might request global information or quantity of computation resources for further processing [22].

We could roughly classify methods for edge computing into three categories, *i.e.*, fog computing, cloudlets and mobile edge computing (MEC). We pay special attention to MEC, which acts as a beneficial complement to cloud infrastructures by encouraging computing works to be operated at the end of IoT network, *i.e.*, mobile devices. As a novel computing paradigm, MEC brings advantages of low transmission delay, enough battery life, low bandwidth expenditure, and privacy preservation in the mobile IoT [23]. To solve the high computation and communication uncertainty of servers' resources, Apostolopoulos *et al.* [24] proposed to use a non-cooperative game among the users a distributed low-complexity algorithm that converges to the Pure Nash Equilibrium.

As the major data types generated by mobile devices, mobile multimedia data (*e.g.*, images, videos, and voice recordings) processing becomes more critical in the domain of MEC. To excavate valuable information from raw mobile multimedia data [25], researchers have developed quantity of applications by involving technologies like computer vision (CV), natural language processing (NLP) and so on. How to properly deploy these applications under the guidance of the MEC paradigm becomes a major challenge. To this end, we migrate and implement the traffic flow detection algorithm to the edge device, and the experiments have demonstrated the advantage of our framework with a good performance [26]–[28].

Essentially, applications on multimedia data processing require large computation resources, which might exceed local computing capacity in edges. By constructing a powerful computing system in cloud for computation supplement, Ali *et al.* [29] proposed a dynamic priority-based resource allocation scheme to provide media processing services with high QoS. For edge server quantification and placement, Xu *et al.* [30] developed a collaborative method, which could deal with vehicular social media in a manner of low time delay and high robustness. Most recently, Lu *et al.* [31] proposed a collaborative learning approach named Colla, which allows cloud and devices to learn collectively and continuously and build tailored model for each device, greatly shortening the training time and improving the accuracy. Jian *et al.* [32] proposed a cost-efficient system named Spatula, which enables scaling cross-camera analytics on edge compute boxes to large camera networks by leveraging the spatial and temporal cross-camera correlations. It can drastically reduce the communication and computation costs.

With the similar idea to build computing servers in cloud for large computation burden, the proposed framework is carefully designed to allocate time-consuming computation for cloud based on small-size data and other computing tasks for edges, thus resulting in fast response and low-bandwidth requirement.

C. Object Detection

Object detection is to find out all interested objects in an input image and determine their categories and positions, which is one of the most fundamental problems in computer vision. Early, the Viola-Jones algorithm [33] was proposed to detect a frontal face by using Haar-like features to describe face, and several different rectangular features were obtained by establishing integral images. Afterwards, they adopted an AdaBoost algorithm to construct a cascaded classifier for face detection.

Currently, we can category most of object detection methods into two kinds, *i.e.*, one stage and two stage, where two stage means that the detection algorithm needs to be completed in two steps. Specifically, the candidate regions computed by the first step need to be classified in the second step, such as R-CNN series [34]. In contrast, a one-stage detection directly regresses labels and locations of possible objects, such as in the SSD [35] and Yolo [36] approaches.

To design for special application scenarios, ORSIm detector [37] adopted a novel spatial-frequency channel feature (SFCF) by jointly considering the rotation-invariant channel features constructed in the frequency domain and the original spatial channel features, which meet the demand for effectively and efficiently handling image deformations, particularly objective scaling and rotation. Later, Wu *et al.* [38] proposed a Fourier-based rotation-invariant feature boosting method, which solved the sensitivity of object deformations from the view of frequency domain.

Considering the complexity of classification on objects in remote sensing images, Hong *et al.* [39] proposed a multimodal deep learning (MDL) framework, which focuses on "what," "where," and "how" to fuse with different fusion strategies as well as how to train deep networks and build the network

Authorized licensed use limited to: Hohai University Library. Downloaded on November 22,2023 at 02:01:04 UTC from IEEE Xplore. Restrictions apply.

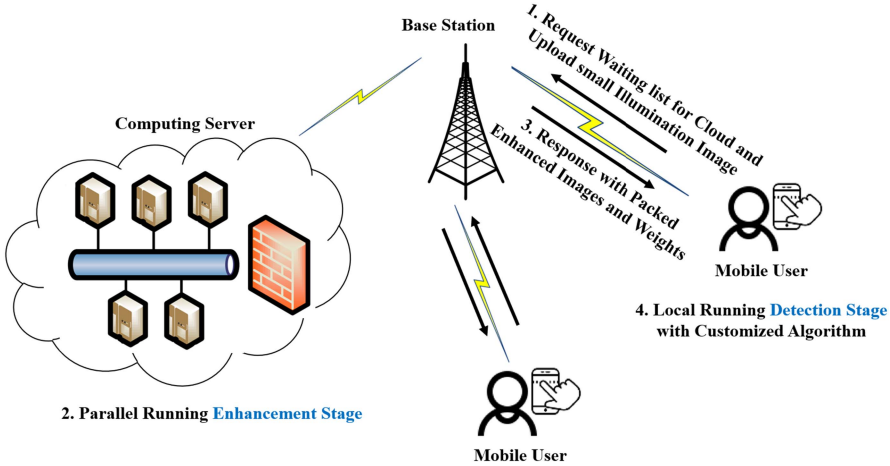

Fig. 2. Framework design of cloud-edge computing for one typical mobile multimedia processing task, *i.e.*, dynamical image enhancement for object detection.

architecture. Later, miniGCN [40] was proposed that allowed training large-scale GCNs in a minibatch fashion, in order to minimize the computational cost of Hyperspectral Image Classification.

Most related to our work, Jiang *et al.* [8] proposed a flexible framework for high-resolution object detection on edge devices, which divides the input image into different regions and allocates different computing power to minimize latency. Later, Zhang *et al.* [11] proposed the design of Elf, which employs a recurrent region proposal prediction algorithm, a region proposal centric frame partitioning, and a resource-aware multi-offloading scheme. It can largely speed up the applications and save bandwidth usage.

III. The Proposed Method

In this section, we describe the proposed edge-driven and multi-stage image enhancement method for object detection. We firstly introduce the overall framework, which offers a global view on how the proposed method works. Then, we show the structure of one enhancement subnetwork to perform dynamical image enhancement. Afterwards, we illustrate the process to perform object detection on weighted feature maps with two customized detectors. Finally, we will demonstrate training procedures with loss functions.

A. Overall Framework

In this subsection, we offer descriptions of the proposed cloud-edge computing framework and multi-stage network design, respectively.

Cloud-Edge Computing Framework. The structure of the proposed cloud-edge computing framework is shown in Fig. 2, where we can notice the cloud-located computing server connects with all mobile devices with good network connectivity by a wireless network. The computing server then responds to mobile users' queries on the computation workloads. When the computing server hears multiple requests from mobile users, it performs parallel computing in cloud and offers proper responses to users after computing. In our case, if a mobile user wants to know the object categories inside his/her camera captured image with bad light condition, the proposed framework follows these four steps:

1) Users are given options to perform enhancement locally or on the cloud server. If they choose the latter, the mobile device will send the request for enhancement tasks to the computing server and upload small-size illumination images for further computation via cellular network;
2) The computer server arranges computation workloads for multiple mobile users and performs parallel running on enhancement stage for one specific user;
3) The computing server sends small-size responses to users via a cellular network, which contains packed enhanced images and corresponding weights computed by the last step;
4) Mobile devices work as edges to locally run the detection stage with the customized algorithm, which offers accurate detection results to users.

It is worth noting that during actual use, users have the right to choose whether to upload image data to the ECS. If the user choose 'yes,' that means the data subject has provided informed consent to data processing for lawful purposes. This meets the requirements of General Data Protection Regulation and California Consumer Privacy Act. In summery, two communication budget should be calculated, where only the component of luminance-chrominance color space, represented by Y, for an input image is sent from the device to the serve, and the data sent from the server to mobile devices refers to exposure maps.

We design the proposed framework based on several considerations. To mitigate bandwidth requirements, we firstly decrease the size of the transmission data between mobile devices and computing server by transmitting several vital data, *i.e.*, illumination image, enhanced images and corresponding weights. To deal with extreme low-light situations, it is essential to place as many enhancement subnetworks for dynamical enhancing. However, the quantity of subnetworks brings large computation burden, which can only be performed in cloud with sufficient computation resources. Furthermore, subnetworks are dual connected, which are suitable to be parallel performed in

Authorized licensed use limited to: Hohai University Library. Downloaded on November 22,2023 at 02:01:04 UTC from IEEE Xplore. Restrictions apply.

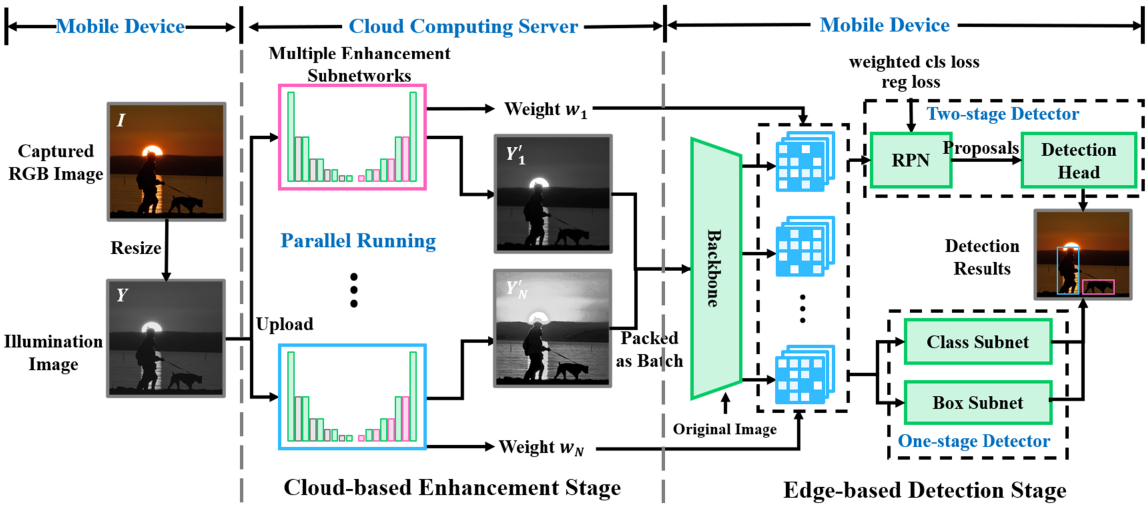

Fig. 3. The proposed Edge-driven and multi-stage network design consists of multiple enhancement subnetworks and two versions of detectors, *i.e.*, one-stage and two-stage detectors.

cloud. Last but not least, mobile devices could have large gap in computation capability, which leads us to design different versions of local running algorithms for fast response.

Edge-driven and Multi-stage Network Design. As illustrated in Fig. 3, the proposed multi-stage network design consists of two stages: a cloud-based enhancement stage and an edge-based detection stage. In the enhancement stage, we adopt a number of fully-convolutional subnetworks to generate sample-specific convolution kernels for dynamic enhancing of low-light images. During the detection stage, either a one-stage, *i.e.*, variant of RetinaNet [41], or a two-stage detector, *i.e.*, variant of Faster R-CNN [34], is adopted for object detection, where they perform inferences based on weighted feature maps extracted from enhanced images. It is noted that the global goal of the proposed method is to accurately detect objects other than image enhancement, where low-level features corresponding to represent a low-light property are further shared by a high-level vision task, *i.e.*, object detection, for performance boosting.

Specifically, the mobile captured RGB image I is firstly transformed into a luminance-chrominance color space, which outputs luminance component Y and the chrominance components (including Cb and Cr). After transforming, Y is processed to be an illumination image by resizing into a fixed size, *i.e.*, 224×224, which is a rather small number for conveniences of fast transmitting and processing.

Afterwards, if the user choose to complete the enhancement on a cloud server, Y is uploaded to the cloud computing server and processed by N enhancement subnetworks in parallel, which could be represented as

$$Y_i' = \mathcal{N}_i(Y) \quad 1 \leqslant i \leqslant N, \tag{1}$$

where Y_i' is computed by the i-th subnetwork represented as a function $\mathcal{N}_i()$.

After the image enhancement in cloud step, outputs of multiple enchantment subnetworks Y_i' are packed as a batch. Both set of enhanced images $\{Y_i' | i = 1, \ldots, N\}$ and weight vector ω with size N are transmitted to mobile devices for further local processing. After utilizing a backbone network for feature map generation, we multiply weights ω and feature map for task-specified enhancement for feature map. Finally, either the one-stage or two-stage detector is applied to generate bounding box and classify object category, where we specifically involve weighted classification loss and regression loss for training of both detectors.

B. Structure of Enhancement Subnetwork

We show the structure design of the enhancement subnetwork in Fig. 4, where we can notice two important outputs for enhancement, *i.e.*, dynamical filter and exposure map. We design dynamic filter to simulate a specific manually designed enhancement filter, while the exposure map could offer exposure ratios for low-light images to make dark areas be lighter for detection.

Essentially, we design both modules in one subnetwork due to several advantages. First of all, sharing feature maps between two modules could largely reduce computation cost, and keep consistent performance after operations of enhancing and exposing. By performing same operation on image pixels, convolutional operation with pre-trained kernel is beneficial to noise suppression. However, it is difficult for dynamic filter to well simulate nonlinear enhancement methods, due to linear property of the convolutional operation. Moreover, the convolutional operation involves neighbouring pixel information to blur edges of objects. On the contrary, the exposure map is a pixel-wise operation to provide ability of non-linear modeling. Meanwhile, it could retain noise and avoid blur effects without introducing neighbouring information. Therefore, it is a wise option to involve strengths of both modules for better low-light enhancement, thus improving performance of object detection.

Specifically, the illumination image Y is fed into the down-sampling part of a U-Net [42] to compute a feature vector with a fixed size. Afterwards, the feature vector K could be computed following with an additional fully-connected layer:

Authorized licensed use limited to: Hohai University Library. Downloaded on November 22,2023 at 02:01:04 UTC from IEEE Xplore. Restrictions apply.

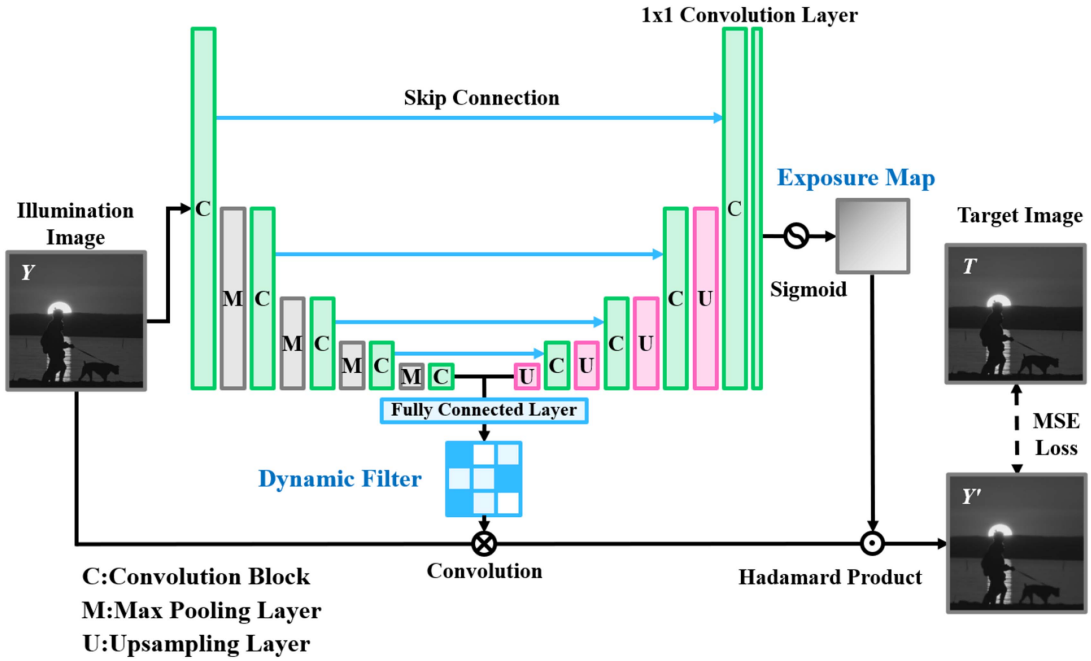

Fig. 4. Architecture design of the proposed enhancement subnetwork, where heights of layers correspond to size of generated feature map, and convolution blocks contain two consecutive convolutional layers activated by ReLU.

$$K = f_{fc}(f_d(Y)), \tag{2}$$

where $K \in \mathbb{R}^{s\times s\times 1}$, s is the pre-defined kernel size and will be determined by parameter setting experiments in Section IV-B, functions $f_d()$ and $f_{fc}()$ refer to the corresponding down-sampling and fully-connected layers. After computing, we regard the resulting feature vector as dynamic filter K for further processing. It is worthy to be noted that the number of channels of the feature map is continuously increasing during down-sampling, which proves that operations shown in (3) can provide sufficient information for feature map generation.

Following the down-sampling process, the proposed exposure map module firstly performs the up-sampling operations of the U-Net. Then, it re-scales the feature with a 1×1 convolution layer and a sigmoid function. Finally, it resizes the generated feature with the original size of the input illumination image Y using a bilinear interpolation method. All these processes can be represented as:

$$E = f_b(sig(f_1(f_d(Y)))), \tag{3}$$

where $E \in \mathbb{R}^{h\times w\times 1}$, h and w are the height and width of Y, functions $f_1()$, $sig()$, and $f_b()$ represent operations of the 1×1 convolution layer, sigmoid function and bilinear interpolation, respectively. It is noted the up-sampling process is opposite to down-sampling in the number of feature channels.

Finally, the input illumination image Y will be enhanced with operations of dynamic filter K and exposure map E:

$$Y' = (Y * K) \circ E, \tag{4}$$

where $*$ and $\circ$ denote the convolution operation and the Hadamard product, respectively. Specifically, the Hadamard product can be regarded as applying one weighted feature map to another one, which could fuse two feature maps with a small amount of computation. Therefore we choose it for morphological operations other than other operations.

We note that each subnetwork $\mathcal{N}_i$ is prevently and separately trained by calculating the enhancement loss L_i^E between the output of the ith subnetwork Y_i' and the enhanced image T_i computed by a manually designed enhancement filter like a bilateral filter and histogram equalization:

$$L_i^E = \sum_{i=1}^{M} \mathrm{MSE}\left(Y_i', T_i\right). \tag{5}$$

Here, the function $MSE()$ refers to the mean squared error, and M is the total number of training samples.

In fact, we employ the manually designed enhancement filter to constrain the processing of each enhancement subnetwork, leading the pre-trained subnetwork to simulate operations of enhancement filters in the testing workflow. The reason for such training procedure lies in the fact that manually designed filters could be considered as prior knowledge to guarantee good enhancement effects without enough pairwise training samples. The pseudo code is shown in Algorithm 1.

C. Object Detection Based on Weighted Feature Maps

In this subsection, we first introduce a weight computing scheme based on multiple enhancement subnetworks, then illustrate the steps to detect objects based on weighted features.

Comparing with a traditional learning object detection methods, *i.e.*, the Viola-Jones algorithm [33], which integrates Haar-like features, the AdaBoost algorithm, and a cascade structure for accurate detection, the reason that the proposed method, *i.e.*, a typical deep neural network, generally outperforms traditional

Authorized licensed use limited to: Hohai University Library. Downloaded on November 22,2023 at 02:01:04 UTC from IEEE Xplore. Restrictions apply.

Algorithm 1: Cloud-based Enhancement.

Data: single channel map Y
Result: enhanced images Y', calculated weight ω
$N \leftarrow$ the amount of sub networks;
for $N_i \in N$ **do**
  $D_i \leftarrow f_d^i(Y)$;
  $K_i \leftarrow f_{fc}^i(D_i)$;
  $E_i \leftarrow f_b(sig(conv_{1\times1}^i(D_i)))$;
  $Y_i' \leftarrow (Y * K_i) \circ E$;
  $T_i' \leftarrow$ manual designed filters computing on Y_i;
  $L_i^E \leftarrow MSE(Y_i', T_i)$;
  $\omega_i \leftarrow \left(1 - \frac{L_i^E}{\sum_{k=1}^N L_k^E}\right) \times \frac{N}{N-1}$;
end
return Y', ω;

Algorithm 2: Edge-based Detection.

Data: image X, calculate option C on cloud or not, O is 0 for one-stage detector, 1 for two-stage detector, 2 for adaptively determined by score S
Result: detection result Z
$X' \leftarrow$ transform X into luminance-chrominance color space;
$Y \leftarrow$ luminance component of X';
if $C = True$ **then**
  $Y', \omega_n \leftarrow$ enhance Y on cloud;
end
else
  $Y', \omega_n \leftarrow$ enhance Y on edge;
end
$\psi_i \leftarrow \omega_i \cdot f_{bone}(Y_i')$;
if *use one stage detector determined by O and S* **then**
  $Z \leftarrow f_{class}(\psi_i) \oplus f_{box}(\psi_i)$;
end
else
  $Z \leftarrow f_detect(RPN(\psi_i))$;
end
return Z;

methods lies in the fact that deep models have strong nonlinear modeling ability facing real-life complex or semantical centered tasks, thus achieving better performance with more layers for processing. On the contrary, traditional learning methods generally adopt shallow model for processing, which is more effective facing simple linear situations rather than highly non-linear object detection methods.

We note that the first computing task occurs in cloud, and the second one is designed to be performed in edges, *i.e.*, on smart devices. After computing weights, we first assemble enhanced illuminate image set $\{Y_i'|i = 1, \ldots, N\}$ and weight vector ω with size N as a minibatch, and then transmit them into mobile devices for object detection via a wireless network.

After calculating the enhancement loss with (5) in the testing workflow, we could achieve dynamic weight ω_i for each subnetwork by regrading L_i^E as a context information to describe the fitness and effectiveness of applying the ith subnetwork for enhancement, which can be calculated as:

$$\omega_i = \left(1 - \frac{L_i^{\mathrm{E}}}{\sum_{k=1}^N L_k^{\mathrm{E}}}\right) \cdot \frac{N}{N-1}, \tag{6}$$

where ω_i reflects the effectiveness to apply the ith enhancement subnetwork for the enhancement task of this specific low-light image. In other words, a higher value in ω indicates the corresponding enhancement subnetwork is fit to deal with such low-light condition. Based on the weight vector ω, the proposed method can adaptively choose the most effective enhancement subnetwork for further processing.

After enhancing on edge or on cloud and transmitting data from cloud to edge, we transform Y' into the RGB color space with other two chrominance components, thus generating an enhanced image I'. During the edge-based detection stage, we first extract feature maps ψ via the backbone network by inputting set of enhanced images $\{I_i'|i = 1, \ldots, N\}$ and the original image I. Afterwards, we assign the corresponding weight ω_i to a feature map for informativeness evaluation, which can be represented as:

$$\psi_i = \omega_i \cdot f_{bone}(I_i'), \tag{7}$$

where the function $f_{bone}()$ refers to operations in the backbone network, and $\cdot$ denotes element-wise multiplication. It is noted that we defined the weight as 1 for the feature map extracted from I.

In that case, we design two versions of object detectors to match with mobile devices of different computation capabilities. Specifically, the core algorithm of the one-stage detector is RetinaNet [41], which is a single and effective network containing two task-specific subnetworks. The class subnet operates object classification based on weighted feature maps, and the box subnet regresses bounding box's locations. These two subnetworks make the proposed one-stage detector fast and simple enough for robust and dense inferences with less computation requirement.

The two-stage detector based on Faster R-CNN, which firstly generates RoIs through a regional proposal network (RPN), and then classifies objects and refines boxes through detection head. It iss a crucial problem to determine which group of feature maps to extract ROIs. We try three different setting, *i.e.*, use the feature map extracted from the original image, use mean of total $N + 1$ feature maps, use weighted mean of all feature maps. After experiments, we find the first choice contributes to the most stable detection results. Due to the additional structure of RPN, both computation requirement and detection results achieved by the two-stage detector is higher than those obtained with the one-stage detector. The pseudo code is shown in Algorithm 2. Specifically, we choose the one-stage or the two-stage detector by either users' option O or based on a calculation score S related to the computation resource, *i.e.*, CPU and RAM mostly.

D. Multi-Stage Joint Optimization

In this subsection, we unify the enhancement and detection stages in a multi-stage framework for end-to-end joint

Authorized licensed use limited to: Hohai University Library. Downloaded on November 22,2023 at 02:01:04 UTC from IEEE Xplore. Restrictions apply.

optimization. Through the joint optimization, the proposed enhancement network can learn a sample-specific set of parameters to improve detection performance. It is noted that we train the proposed model in an end-to-end manner, while we deploy parts of the trained network in either cloud or edge.

Firstly, we define the enhancement loss L^{E} to evaluate total loss for multiple enhancement subnetworks as:

$$L^{\mathrm{E}} = \alpha \cdot \sum_{i=1}^{N} L_i^{\mathrm{E}}, \tag{8}$$

where α is a weight value for the enhancement loss to be determined by experiments in Section IV-B.

Then, we define a multi-stage loss for the enhancement network and one-stage detector as

$$L = L^{\mathrm{E}} + \frac{1}{N+1} \cdot \sum_{i=0}^{N} L_i^{\mathrm{clsF}} + \frac{1}{N+1} \cdot \sum_{i=0}^{N} L_i^{\mathrm{reg}}, \tag{9}$$

where L_i^{reg} refers to the box regression loss based on the ith weighted feature map ψ_i, and L_i^{clsF} is the $\alpha-$balanced focal loss defined in [41] to address keep balance between foreground and background classes for training.

Finally, we use the weighted classification loss for backpropagation, which could increase the transmitted amount of gradient corresponding to effective enhancement subnetwork, thus improving classification performance of RPN. Based on this conception, we define multi-stage loss of enhancement network and two-stage detector as

$$\begin{aligned} L = L^{\mathrm{E}} &+ \frac{1}{N+1} \cdot \sum_{i=0}^{N} \omega_i L_i^{\mathrm{rpn\_cls}} \\ &+ \frac{1}{N+1} \cdot \sum_{i=0}^{N} L_i^{\mathrm{rpn\_reg}} + L^{\mathrm{cls}} + L^{\mathrm{reg}}, \end{aligned} \tag{10}$$

where ω_0 equals 1, $L_i^{\mathrm{rpn\_reg}}$ and $L_i^{\mathrm{rpn\_cls}}$ refer to the box regression and class classification loss achieved by RPN, respectively. We note that the regression loss of RPN is not weighted, since such design would lead to inaccurate object localization.

IV. Experiments

In this section, we firstly introduce dataset and measurements. Then, we conduct multiple sets of parameter setting and ablation experiments to evaluate sensitivity to parameters and impact of different structure designs, respectively. Afterwards, we perform comparative studies and offer discussions on performance. Finally, we provide implementation details.

A. Dataset and Measurement

Exclusively Dark (ExDark) [43] is an open-access image collections composed by low-light images with object level annotations, which contains 7,363 images and is structured with 12 separate folders, named bicycle, boat, bottle, bus, car, cat, chair, cup, dog, motorbike, people, and table. Specifically, we follow its guidance to use total 3,000 images and 250 images per class for training, a total of 1,800 images and 150 images per class for validation, and 2,563 images for testing. Moreover, the ExDark dataset identifies 10 categories of low-light conditions in indoor and outdoor environments, *i.e.*, extremely *Low* illumination, *Ambient* with weak illumination, *Object* are bright in the dark, *Single* light source, multiple *Weak* light source, multiple *Strong* light source, indoor with bright *Screen*, indoor with bright *Window*, bright outdoor with objects in the *Shadow*, and outdoor with *Twilight*, where bold texts represent key information for low-light category. Based on the above analysis, we believe the ExDark dataset is large in size to provide sufficient information for training and covers as many as low-light environments to abstract and simulate complexity of real-life low-light scene. Therefore, we perform all experiments on the ExDark dataset to show the detection performance under low-light condition.

Since the ExDark dataset does not provide paired high exposure and low-light images, it is difficult to apply supervised enhancement methods on low-light images. In fact, most datasets provide pairwise images by applying low-light filters on bright images, which simplifies the problem of low-light condition. By observing and analyzing complexity of low-light scenes in real-life, we believe it is wise to use convinced object-level annotations for joint optimization, which is appropriate in order to deal with images from real-world environment.

Following the standard measurements for object detection in [44], we apply AP for evaluation, where AP is defined as the mean precision value over multiple IoU thresholds (Intersection over Union) and all the object classes:

$$AP_{U_j} = \frac{1}{10 \times C} \sum_{i=1}^{C} \sum_{j=1}^{10} P(i, U_j), \tag{11}$$

where i and j refer to the index of class and threshold respectively, C is the total number of classes, the IoU values U_j corresponds to a range from 0.5 to 0.95 with a step size of 0.05, and the function $P(i, U_j)()$ calculates precision values for the ith object class under a fixed IoU threshold U_j. Moreover, AP_{50} and AP_{75} refer to mAP values over the IoU thresholds of 0.5 and 0.75 respectively, while AP_{S}, AP_{M} and AP_{L} are the AP for small, medium and large objects, respectively.

Furthermore, we define AR_q as the average recall over detections on Q images

$$AR_q = \frac{1}{Q} \sum_{q=1}^{Q} R(q), \tag{12}$$

where Q is the number of total detected images. AR_q offers another view on robustness of detection results.

B. Parameter Setting Experiment

In this subsection, we perform experiments to determine two important hyper-parameters, *i.e.*, size s of the dynamical filter, and weight for enchantment loss α. Since both hyper-parameters

Authorized licensed use limited to: Hohai University Library. Downloaded on November 22,2023 at 02:01:04 UTC from IEEE Xplore. Restrictions apply.

TABLE I
PERFORMANCE COMPARISON ON DIFFERENT SIZE OF DYNAMICAL FILTERS

Filter Size	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
3×3	31.5	61.9	29.1	5.3	18.9	36.3
5×5	31.7	61.7	29.3	4.0	19.1	36.5
7×7	**32.1**	**62.1**	29.9	3.6	**20.0**	**36.9**
9×9	31.8	61.5	**30.5**	4.1	18.9	36.4
11×11	31.7	61.9	28.8	**5.6**	18.7	36.3

TABLE II
PERFORMANCE COMPARISON ON DIFFERENT STRUCTURE DESIGNS, WHERE EM REPRESENTS THE DESIGN OF THE EXPOSURE MAP

Structure Design	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
With EM	**32.1**	**62.1**	**29.9**	3.6	**20.0**	**36.9**
Without EM	31.5	61.4	29.0	**3.7**	19.7	36.1
Weighted Sum Feature Map	30.8	**62.4**	26.8	**8.0**	**19.1**	35.1
Weighted Feature Maps	**31.8**	62.2	**29.1**	5.4	18.8	**36.4**

are related with design of the enhancement network, we adopt a two-stage detector in the parameter setting experiments for fairness.

As shown in Table I, we test with five groups of s, where we can notice that group with 7×7 filter shows best performance in AP, AP_{50}, AP_M, and AP_L, achieves the second best performance in AP_{75}, and the worst performance in AP_S. Such results can be explained by variant properties brought by different size of filters [45]. For example, a small filter size works well in small object detection, which can be proved by the best performance for the 3×3 filter in AP_S. In our case, we aim to perform object detection without prior knowledge on object size. Therefore, we can observe that a large filter size leads enhanced images to be smooth without enough edge or boundary information, while a small filter size makes it difficult for the trained enhancement subnetwork to simulate manually designed enhancement methods, due to the lack of capability of mathematical modeling. Based on the above discussions, we choose 7×7 as the hyper-parameter of size for dynamical filter.

The basic idea of defining α is to ensure that the enhancement loss can share the same order of magnitude as other losses. We tried several values, where we find a large value like 1 can cause the proposed method to be collapsed, due to the problem of gradient explosion. Meanwhile, too small values of α result in slow convergency speed. By considering multiple factors, we define $\alpha = 0.1$ by experiments, which guarantees the magnitude of the enhancement loss to be reasonable for fast training.

C. Ablation Experiment

To show the efficiency of the proposed designs, we performed two groups of ablation experiments as shown in Table II, where the first group is designed to compare the effectiveness with or without exposure map (EM), and the second group is to compare among different designs on weighting schemes. We note that all testings are carried out with parameter settings determined in the last subsection.

By removing the generating steps for exposure map, the proposed enhancement network only computes dynamic filters for enhancing. From Table II, we can observe the performance with EM is better than one without EM in most of the measurements. Essentially, the dynamical filter is designed as a convolutional operator, which is short of capability in non-linear modeling and could blur edges or areas with high gradients by involving neighboring pixels for calculations. As a beneficial supplement to dynamical filter, EM serves to provide non-linear capability in modeling and avoids blurring effects to a certain extent. During training, we observe fast decreasing speed in the enhancement loss by introducing EM for experiments, which proves that EM can help to achieve a faster converged training. It is worth noting that EM slightly decreases performance on AP_S. Such phenomenon can be explained by the fact that EM may easily interfere with the original feature representation of small objects, and this may cause small objects to lose some valuable information. After all, this effect is slight, and other benefits brought by EM have hedged against its disadvantages.

As shown in Table II, we offer two structure designs on weight schemes. Design of weighted sum feature map represents that we sum all feature maps into one after assigning weights on different feature maps. Therefore, later process of RPN is performed on only one feature map. The overall performance of weighted sum feature map is worse than the weighted feature map, since the weighted feature map largely improves AP, AP_{75} and AP_L, and slightly decreases performance on AP_{50} and AP_M. However, the weighted sum feature map largely improves performance on AP_S. Such phenomenon can be explained by the fact that it is difficult to accurately locate small objects with insufficient information, where summing all the feature maps guarantee amount of information to find small objects. After comparing, we adopt the weighted feature map as the proposed weight scheme, since it results in a higher performance improvement.

D. Comparative Experiment and Performance Discussion

We report detection results achieved by the proposed method and existing methods in Table III, where RN, FR, BFilter, GFliter, HistE and Ims refer to RetinaNet [46], Faster R-CNN [34], Bilateral Filter, Guided Filter [49], Histogram Equalization and Image sharpening, respectively. We note that any listed method without specific defined detector is equipped with a Faster R-CNN for detection. We further categorize methods into three groups. Specifically, the first group shows performance of directly applying the listed object detection methods, meanwhile the second group performs low-light enhancement with manually filters and then utilizes detectors for detection. The last group is designed to show the effectiveness of applying a multi-stage framework for task-specified enhancing, where the proposed enhancement network is pre-trained by HistEq and Imsharpen. We also list several existing methods to compare detection performance. Unlike algorithms for object detection, the purpose of the proposed method is to improve detection performance in low-light conditions by involving strengths of edge computing paradigm and multi-stage learning.

Authorized licensed use limited to: Hohai University Library. Downloaded on November 22,2023 at 02:01:04 UTC from IEEE Xplore. Restrictions apply.

TABLE III
PERFORMANCE COMPARISON BETWEEN THE PROPOSED METHOD AND THE EXISTING METHODS ON THE EX-DARK DATASET, WHERE * IMPLIES WE MODIFY ITS USAGE TO MATCH TOPIC OF LOW-LIGHT ENHANCEMENT

Methods	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	AR_1	AR_{10}	AR_{100}	AR_S	AR_M	AR_L
RN[46]	27.6	52.7	25.9	4.50	16.0	31.8	29.4	45.7	48.6	15.8	34.5	53.3
FR[34]	30.4	61.0	27.3	4.30	18.8	35.2	28.3	43.5	44.5	7.30	31.4	49.4
BFilter+FR	27.8	57.6	23.2	2.20	17.3	32.0	27.0	42.5	43.6	5.80	30.1	48.4
BFilter+RN	24.6	48.6	22.5	3.90	13.1	29.0	28.1	44.3	47.0	12.6	33.4	51.8
GFilter+FR	25.8	53.9	21.7	1.90	14.6	30.2	26.1	41.2	42.5	5.50	28.5	47.6
GFilter+RN	19.0	38.6	16.4	3.40	9.80	22.6	25.5	41.5	44.0	10.5	28.9	49.3
HistE+FR	28.4	57.4	25.6	2.60	17.6	32.7	27.1	42.7	43.7	7.00	28.5	48.6
HistE+RN	23.8	46.4	22.3	3.40	14.0	27.6	27.4	43.9	46.7	12.1	33.1	51.4
Ims+FR	28.9	59.2	25.6	3.40	14.6	28.7	28.5	45.9	48.2	16.9	35.2	55.1
Ims+RN	24.1	46.7	22.5	3.10	14.2	28.0	28.0	45.1	47.5	16.5	34.5	52.1
Jiang et al. [20]	30.7	61.4	27.5	4.39	19.0	35.4	28.4	43.9	44.8	7.38	31.6	49.7
Lim et al.*[47]	31.2	61.8	27.9	4.62	**19.1**	35.7	28.8	44.2	45.2	7.53	32.1	50.2
Wu et al.*[48]	31.3	61.8	28.1	4.69	**19.1**	35.9	28.9	44.1	45.2	7.59	32.0	50.4
Viola-Jones*[33]	12.4	35.9	10.6	2.01	6.9	12.7	7.8	19.6	22.6	2.90	18.6	24.4
HistE+Ims+FR	31.6	61.6	28.5	3.70	18.6	36.4	29.1	44.5	45.9	9.40	29.4	50.6
HistE+Ims+FN	27.7	52.5	26.3	4.20	16.8	31.7	29.3	**46.8**	49.5	17.7	**36.3**	53.9
Proposed+FR	**31.8**	**62.2**	**29.1**	**5.40**	18.8	**36.4**	29.0	45.0	46.1	9.50	30.9	51.1
Proposed+FN	28.6	54.1	27.4	**5.40**	17.0	33.1	**29.7**	46.7	**49.4**	**25.2**	35.8	**54.0**

It is claimed that we achieve public codes of RetinaNet [46], Faster R-CNN [34], Jiang *et al.* [20], Lim *et al.* [47], Wu *et al.* [48] and the Viola-Jones algorithm [33] for testing, where Faster R-CNN is further improved with structure of FPN [47]. Since there exist few open-source deep-learning methods for low-light enhancement, we modify two typical super-resolution methods, *i.e.*, EDSR and CASR, and one face detection method, *i.e.*, the Viola-Jones algorithm, to complete the task of enhancement. The reason lies in the fact that enhancement is one of the important tasks in super-resolution. Specifically, we remove the up-sample layer of two networks and retain other parts for modeling, where Sony set in the See-in-the-Dark [15] dataset is adopted for pairwise training on enhancement task. It is noted that we modified the Viola-Jones algorithm to fit for object detection other that face detection, and 422 cascade-connected detectors are adopted for detecting in experiments.

First Group of Tests. Without any enhancement operations for low-light images, directly applying RetinaNet and Faster R-CNN leads to low detection performance, which not only reflects a high degree of difficulty in accurately detecting with extremely low-night condition, but also makes it essential to design task-specified filter for enhancement.

Second Group of Tests. As illustrated in the second group, we could notice that manually designed filters can even reduce detection performance especially for Bilateral Filter and Guided Filter [49], which might be caused by the fact that these two filters would suppress noise, but blur boundary of objects. These unexpected blurring effects would make it more difficult to locate and classify objects in the dark. After comparing among all manually designed filters, we choose histogram equalization and image sharpening filter to construct the proposed enhancement network.

Third Group of Tests. In the last group, we define "HistE +Ims+FR" as sequentially apply two filters on the images for low-light enhancing. We could observe that this simplified integration design could improve the detection performance, comparing with either "HistE+FR" or "Ims+FR". However, utilizing two or more filters can result in interrelation phenomenon, where filters might have opposite operations on the same part of the image, thus generating artifacts or low enhancement effects. Therefore, we need to design sample-specific weight to dynamically combine multiple filters based on context information, which is the core idea of the proposed method. By simulating these two simple filters and adjusting their weights for proper enhancement, the proposed method largely outperforms "HistE+Ims+FR". Essentially, the multi-stage learning framework offers additional object detection specified information for enhancement network in joint training procedure, which not only contributes to sharing of feature

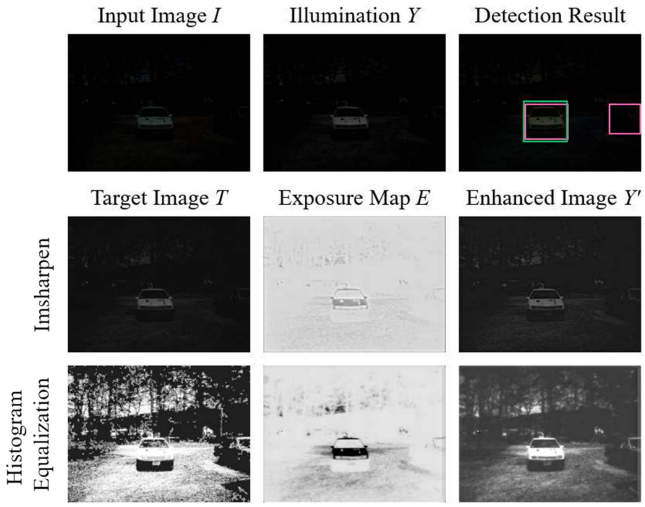

Fig. 5. Samples of intermedin results computed by the proposed enhancement networks, where the first row computes detection results by defining red rectangle as ground truth and green one as prediction, the second row shows training image T output by Imsharpen filter, resulting exposure map E and enhanced image Y, and the third row is arranged as the same as the second one.

Authorized licensed use limited to: Hohai University Library. Downloaded on November 22,2023 at 02:01:04 UTC from IEEE Xplore. Restrictions apply.

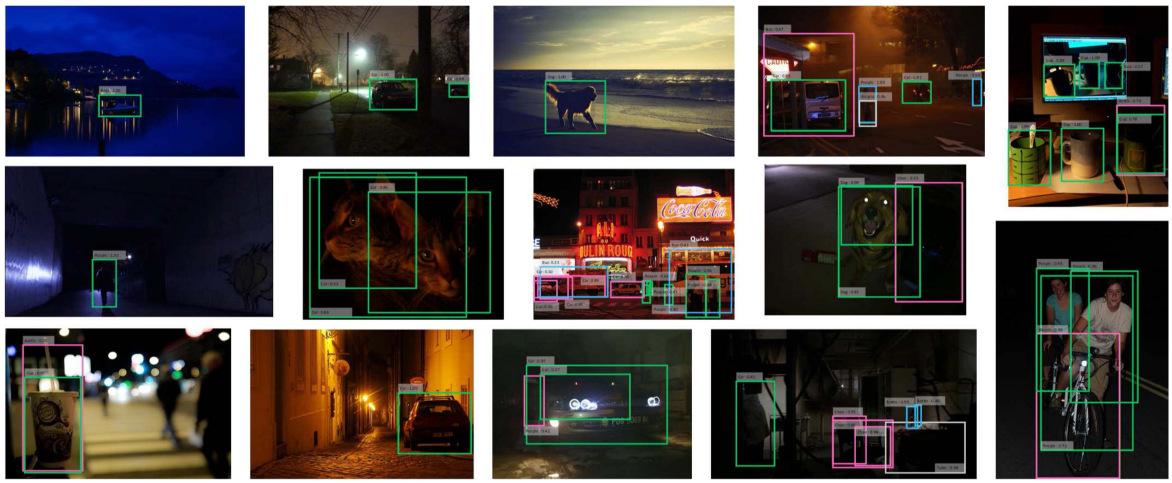

Fig. 6. Quantity of detection samples achieved by the proposed method with a two-stage detector. Rectangles with the same color refer to detection results with the same category. Zoom for the best viewing experience.

Fig. 7. Quantity of detection samples achieved by the proposed method with a one-stage detector.

maps, but also becomes the source of context information for dynamical weighting scheme. Due to the low capability to deal with non-linear complexity, the Viola-Jones algorithm appears to be worst in performance even with quantity of cascade-connected detectors, which proves the power of deep neural networks for multimedia data processing.

As shown in the last group, "proposed+FR" achieves the best precision performance among comparative studies, except for AP_M. In fact, the proposed enhancement network fails in improving performance corresponding to the task of detecting medium-size objects, comparing with AP_M achieved by FR. The reason lies in the fact that we cannot cover up all situations to improve detection tasks by defining a fixed size of the filter before training. Meanwhile, "proposed+FN" achieves the best recall performance among comparative studies, except for AR_{10} and AR_M. We note that the measurement of AR only takes recall values into account, which cannot reflect overall detection performance comparing with AP. However, it could provide information on tendency of detectors. We thus conclude that the one-stage detector, *i.e.*, FN, tends to achieve a high recall value rather than a high precision value, which depends on its strategy in keeping balance between computation and performance.

We could further notice that GAN-based methods, *i.e.*, Jiang *et al.* [20], slightly improve the overall performance comparing with original FR detector, which can be explained by the fact the purpose of their algorithm is to provide visual desirable effects rather than improving detection performance. Both CNN-based methods, *i.e.*, Lim *et al.** [47] and Wu *et al.** [48] achieve significant improvement on overall performance, due to their capability in constructing high distinguish and task-specified feature maps for enhancement. However, their abilities are highly constrained by the training dataset, where the See-in-the-Dark dataset mostly contains indoor images and cannot provide enough flexibility and diversity to deal with extreme low-light situation in the Ex-Dark dataset.

Intermediate Results. We demonstrate the intermediate results, *i.e.*, exposure map E and enhanced illumination image Y', computed by the enhancement network in Fig. 5, where we can observe the proposed enhancement network is powerful in simulating manually designed filter. Moreover, we believe such training procedure can be viewed as transferring of priori knowledge from manually designed filters to a neural network.

Detection Results. Both Figs. 6 and 7 show a quantity of detection examples obtained by the proposed method with

Authorized licensed use limited to: Hohai University Library. Downloaded on November 22,2023 at 02:01:04 UTC from IEEE Xplore. Restrictions apply.

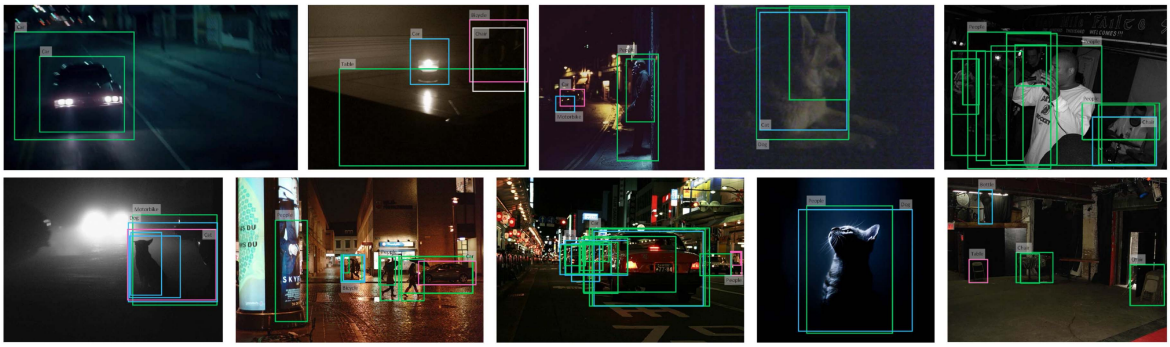

Fig. 8. Quantity of failure cases achieved by the proposed method, where the first and the second row correspond to results output by the two-stage and the one-stage detector, respectively.

different detectors, where we can view accurate locations of bounding boxes and class labels, especially in extremely low-light conditions. All these samples show the effectiveness on the task of object detection in low-light conditions. For readers' convenience to understand the shortages of the proposed method, we also show some failure cases in Fig. 8 computed by our method, where we can notice most of the cases are wrongly predicted due to the complexity of extreme low-light conditions. Several failure cases like the fourth images in both the first and second row are caused by the semantical ambiguity of the visual information, which might require context information for accurate inferences.

Complexity Analysis. The proposed method run either on the cloud or on the target edge devices. According to our measurements, when offloading to the cloud, a typical high-resolution image with 4000×3000 in size is compressed and transmitted in 30 seconds under a 4 G cellular network. The analyzing spend around 15 seconds in total on the server with Intel Xeon E5-2620 v4 (@2.1 GHz) CPUs and four NVIDIA GTX 1080Ti graphics cards. Afterwards, it also cost around 42 seconds for offloading to the mobile phone. The consuming time for the one stage detector is around 36 seconds for an image, which this can be larger to 52 seconds for the two stage detector, where the mobile phone is adopted as Huawei Pro30 with Kylin 990 chips. For a fair comparison, we apply Wu *et al.* [48] for complexity comparisons in cloud, where the analyzing speed is about 12 seconds per image. Being faster than the proposed method, Wu *et al.* [48] performs the super-resolution task without a special design purpose to deal with the complexity of extreme low-light conditions, which can be proved by the lower improvements in object detection.

E. Implementation Details

All our experiments were conducted on a server with two Intel Xeon E5-2620 v4 (@2.1 GHz) CPUs and four NVIDIA GTX 1080Ti graphics cards. Our experimental codes are mainly based on the PyTorch framework. We train all our models for 12 epochs using the SGD optimizer with an initial learning rate of 0.01. The weight decay is set to 0.0001 for the two-stage detector and 0.005 for the one-stage detector, and the momentum is 0.9. Due to the linear warm up mechanism, the learning rate increases from $1/3 \times 0.01$ to 0.01 in the first 500 iterations. We choose the ResNet-50 as the backbone network and a 5-level feature pyramid extracted by FPN. We apply data augmentation by horizontal flip with 0.5 probability for both baselines and our method.

V. Conclusion

This paper presents an edge-computing and multi-stage driven solution for object detection task in low-light conditions. The proposed framework consists of two stages: cloud-based enhancement and edge-based detection stage. In the first stage, we design parallel running enhancement networks to dynamically generate filters and exposure maps. During the second stage in edge, two versions of detectors perform detection task based on weighted feature maps. By testing on the Exclusively Dark dataset, results show that our method significantly improves detection performance by involving ideas of multi-stage learning and edge computing. Our future work is to test the proposed method with potential datasets, such as the SID dataset, which offers pictures in dark indoor environment, and the MEF dataset which contains 20 multi-exposure sequences of dynamic scenes and their corresponding fused images computed by nine MEF algorithms.

References

[1] Q. Zhou, C. Yu, Z. Wang, Q. Qian, and H. Li, "Instant-teaching: An end-to-end semi-supervised object detection framework," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 4079–4088.

[2] L. Chen, T. Yang, X. Zhang, W. Zhang, and J. Sun, "Points as queries: Weakly semi-supervised object detection by points," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 8819–8828.

[3] M. Xu, T. Xu, Y. Liu, and F. X. Lin, "Video analytics with zero-streaming cameras," in *Proc. USENIX Annu. Tech. Conf.*, 2021, pp. 459–472.

[4] R. Bhardwaj *et al.*, "Ekya: Continuous learning of video analytics models on edge compute servers," 2020, *arXiv:2012.10557*.

[5] G. Ananthanarayanan, V. Bahl, L. P. Cox, A. Crown, S. Nogbahi, and Y. Shu, "Video analytics - killer app for edge computing," in *Proc. 17th Annu. Int. Conf. Mobile Syst., Appl., Serv.*, 2019, pp. 695–696.

[6] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, vol. 97, pp. 6105–6114.

[7] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 10778–10787.

[8] S. Jiang, Z. Lin, Y. Li, Y. shu, and Y. Liu, "Flexible high-resolution object detection on edge devices with tunable latency," in *Proc. Annu. Int. Conf. Mobile Comput. Netw.*, 2021, pp. 559–572.

Authorized licensed use limited to: Hohai University Library. Downloaded on November 22,2023 at 02:01:04 UTC from IEEE Xplore. Restrictions apply.

[9] V. Ruzicka and F. Franchetti, "Fast and accurate object detection in high resolution 4 k and 8 k video using GPUs," in *Proc. IEEE High Perform. Extreme Comput. Conf.*, 2018, pp. 1–7.

[10] F. O. Unel, B. O. Özkalayci, and C. Çigla, "The power of tiling for small object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2019, pp. 582–591.

[11] W. Zhang *et al.*, "ELF: Accelerate high-resolution mobile deep vision with content-aware parallel offloading," in *Proc. ACM 27th Annu. Int. Conf. Mobile Comput. Netw.*, 2021, pp. 201–214.

[12] Z. Ying, G. Li, Y. Ren, R. Wang, and W. Wang, "A new low-light image enhancement algorithm using camera response model," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, 2017, pp. 3015–3022.

[13] M. Gharbi, J. Chen, J. T. Barron, S. W. Hasinoff, and F. Durand, "Deep bilateral learning for real-time image enhancement," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 118:1–118:12, 2017.

[14] C. Wei, W. Wang, W. Yang, and J. Liu, "Deep retinex decomposition for low-light enhancement," in *Proc. Brit. Mach. Vis. Conf.*, 2018, pp. 155–167.

[15] C. Chen, Q. Chen, J. Xu, and V. Koltun, "Learning to see in the dark," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3291–3300.

[16] M. Zhu, P. Pan, W. Chen, and Y. Yang, "EEMEFN: Low-light image enhancement via edge-enhanced multi-exposure fusion network," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 13106–13113.

[17] R. Liu, L. Ma, J. Zhang, X. Fan, and Z. Luo, "Retinex-inspired unrolling with cooperative prior architecture search for low-light image enhancement," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 10 556–10 565.

[18] F. Zhang, Y. Li, S. You, and Y. Fu, "Learning temporal consistency for low light video enhancement from single images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 4965–4974.

[19] J. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2242–2251.

[20] Y. Jiang *et al.*, "EnlightenGAN: Deep light enhancement without paired supervision," 2019, *arXiv:1906.06972*.

[21] X. Wang, L. T. Yang, Y. Wang, L. Ren, and M. J. Deen, "ADTT: A highly-efficient distributed tensor-train decomposition method for IIoT Big Data," *IEEE Trans. Ind. Informat.*, vol. 17, no. 3, pp. 1573–1582, Mar. 2021.

[22] Y. Lv, Y. Duan, W. Kang, Z. Li, and F. Wang, "Traffic flow prediction with Big Data: A deep learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 865–873, Apr. 2015.

[23] C. Chen, Y. Zeng, H. Li, Y. Liu, and S. Wan, "A multi-hop task offloading decision model in MEC-enabled Internet of Vehicles," *IEEE Internet Things J.*, to be published, doi: 10.1109/JIOT.2022.3143529.

[24] P. A. Apostolopoulos, E. Tsiropoulou, and S. Papavassiliou, "Risk-aware data offloading in multi-server multi-access edge computing environment," *IEEE/ACM Trans. Netw.*, vol. 28, no. 3, pp. 1405–1418, Jun. 2020.

[25] L. Ren, Z. Meng, X. Wang, L. Zhang, and L. T. Yang, "A data-driven approach of product quality prediction for complex production systems," *IEEE Trans. Ind. Informat.*, vol. 17, no. 9, pp. 6457–6465, Sep. 2021.

[26] C. Chen, B. Liu, S. Wan, P. Qiao, and Q. Pei, "An edge traffic flow detection scheme based on deep learning in an intelligent transportation system," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 3, pp. 1840–1852, Mar. 2021.

[27] S. Wan, S. Ding, and C. Chen, "Edge computing enabled video segmentation for real-time traffic monitoring in Internet of Vehicles," *Pattern Recognit.*, vol. 121, 2022, Art. no. 108146.

[28] C. Chen, L. Liu, S. Wan, X. Hui, and Q. Pei, "Data dissemination for industry 4.0 applications in Internet of Vehicles based on short-term traffic prediction," *ACM Trans. Internet Technol.*, vol. 22, no. 1, pp. 1–18, 2021.

[29] A. Ali *et al.*, "Priority-based cloud computing architecture for multimedia-enabled heterogeneous vehicular users," *J. Adv. Transp.*, vol. 2018, pp. 1–12, 2018.

[30] X. Xu *et al.*, "Edge server quantification and placement for offloading social media services in industrial cognitive IoV," *IEEE Trans. Ind. Informat.*, vol. 17, no. 4, pp. 2910–2918, Apr. 2021.

[31] Y. Lu *et al.*, "Collaborative learning between cloud and end devices: An empirical study on location prediction," in *Proc. IEEE/ACM Symp. Edge Comput.*, 2019, pp. 139–151.

[32] S. Jain *et al.*, "Spatula: Efficient cross-camera video analytics on large camera networks," in *Proc. 5th IEEE/ACM Symp. Edge Comput.*, 2020, pp. 110–124.

[33] P. A. Viola and M. J. Jones, "Robust real-time face detection," *Int. J. Comput. Vis.*, vol. 57, no. 2, pp. 137–154, 2004.

[34] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. 29th Conf. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[35] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, 2016, vol. 9905, pp. 21–37.

[36] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 779–788.

[37] X. Wu, D. Hong, J. Tian, J. Chanussot, W. Li, and R. Tao, "ORSIm detector: A novel object detection framework in optical remote sensing imagery using spatial-frequency channel features," *IEEE Trans. Geosci. Remote. Sens.*, vol. 57, no. 7, pp. 5146–5158, Jul. 2019.

[38] X. Wu, D. Hong, J. Chanussot, Y. Xu, R. Tao, and Y. Wang, "Fourier-based rotation-invariant feature boosting: An efficient framework for geospatial object detection," *IEEE Geosci. Remote. Sens. Lett.*, vol. 17, no. 2, pp. 302–306, Feb. 2020.

[39] D. Hong *et al.*, "More diverse means better: Multimodal deep learning meets remote-sensing imagery classification," *IEEE Trans. Geosci. Remote. Sens.*, vol. 59, no. 5, pp. 4340–4354, May 2021.

[40] D. Hong, L. Gao, J. Yao, B. Zhang, A. Plaza, and J. Chanussot, "Graph convolutional networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote. Sens.*, vol. 59, no. 7, pp. 5966–5978, Jul. 2021.

[41] T. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 318–327, Feb. 2020.

[42] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Interv.*, 2015, pp. 234–241.

[43] Y. P. Loh and C. S. Chan, "Getting to know low-light images with the exclusively dark dataset," *Comput. Vis. Image Understanding*, vol. 178, pp. 30–42, 2019.

[44] T.-Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.

[45] V. Sharma, A. Diba, D. Neven, M. S. Brown, L. Van Gool, and R. Stiefelhagen, "Classification-driven dynamic image enhancement," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4033–4041.

[46] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2980–2988.

[47] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2017, pp. 1132–1140.

[48] Y. Wu, X. Ji, W. Ji, Y. Tian, and H. Zhou, "CASR: A context-aware residual network for single-image super-resolution," *Neural Comput. Appl.*, vol. 32, no. 18, pp. 14533–14548, 2020.

[49] K. He, J. Sun, and X. Tang, "Guided image filtering," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 1–14.

Authorized licensed use limited to: Hohai University Library. Downloaded on November 22,2023 at 02:01:04 UTC from IEEE Xplore. Restrictions apply.