

# Effective Denoising Architecture for Handling Multiple Noises

Na Hyoun Kim and Namgyu Kim\*

The Graduate School of Business IT of Kookmin University, Seoul, 02707, Korea

\*Corresponding Author: Namgyu Kim. Email: ngkim@kookmin.ac.kr

Received: 10 March 2022; Accepted: 18 April 2022

**Abstract:** Object detection, one of the core research topics in computer vision, is extensively used in various industrial activities. Although there have been many studies of daytime images where objects can be easily detected, there is relatively little research on nighttime images. In the case of nighttime, various types of noises, such as darkness, haze, and light blur, deteriorate image quality. Thus, an appropriate process for removing noise must precede to improve object detection performance. Although there are many studies on removing individual noise, only a few studies handle multiple noises simultaneously. In this paper, we propose a convolutional denoising autoencoder (CDAE)-based architecture trained on various types of noises. We also present various composing modules for each noise to improve object detection performance for night images. Using the exclusively dark (ExDark) Image dataset, experimental results show that the Sequential filtering architecture showed superior mean average precision(mAP) compared to other architectures.

**Keywords:** Object detection; computer vision; nighttime; multiple noises; convolutional denoising autoencoder

## 1 Introduction

Object detection is a technology that solves the problem of finding the location of a specific object (e.g., person, car, and tree) in an image and classifying the class to which the object belongs. Various attempts have been made in the field of object detection, classified into “general object detection” [1,2] and “specific object detection” [3,4] approaches. The former aims to increase the accuracy and speed of detecting general objects, and the latter refer to approaches for detecting specific objects or general objects in a specific environment.

In other words, “specific object detection” detects a specific instance or improves the object detection performance in a specific environment, where it is difficult to identify an object. These approaches have been widely used in the machinery industry [5] where a specific instance (e.g., a defect) must be detected, and in the security field, such as nighttime CCTV (Closed-circuit television) detection. In particular, night object detection is a technology used in special environments such as border surveillance [6] and autonomous driving at night [7]. It is a very challenging task in that it has to detect objects [8] in a noisy situation [9].

In the case of nighttime, various types of noises such as darkness, haze, and light blur cause deterioration of image quality. Therefore, in order to improve the performance of object detection, various methods have



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

been investigated to remove or adjust such noises. Research on removing darkness through a deep convolutional neural network (DCNN), such as U-net [10], research on removing haze using Dark channel prior [11] and removing light blur using Multi-scale CNN [12] are representative examples of studies to improve image quality through noise removal. As such, many studies have aimed to remove specific noises that occur at night; however, studies for removing multiple noises (not just one) have received relatively little attention. Therefore, this study proposes a convolutional denoising autoencoder (CDAE)-based architectures for filtering multiple noises to improve object detection performance for night images. Specifically, to remove various noises occurring at night, we introduce an integrated architecture that performs filtering on all noises at once, a top one filtering architecture that selectively applies an optimal filter for each image, and a sequential architecture that performs filtering on each noise sequentially. We also present the performance comparison results of these three architectures.

The rest of the paper is organized as follows. Brief reviews of previous works on noise removal in night object detection are presented in Section 2. In Section 3, we introduce three architectures for multiple noise removal. The experimental results of denoising through these three architectures are summarized in Section 4. Finally, Section 5 concludes this paper.

## 2 Related Work

### 2.1 Object Detection

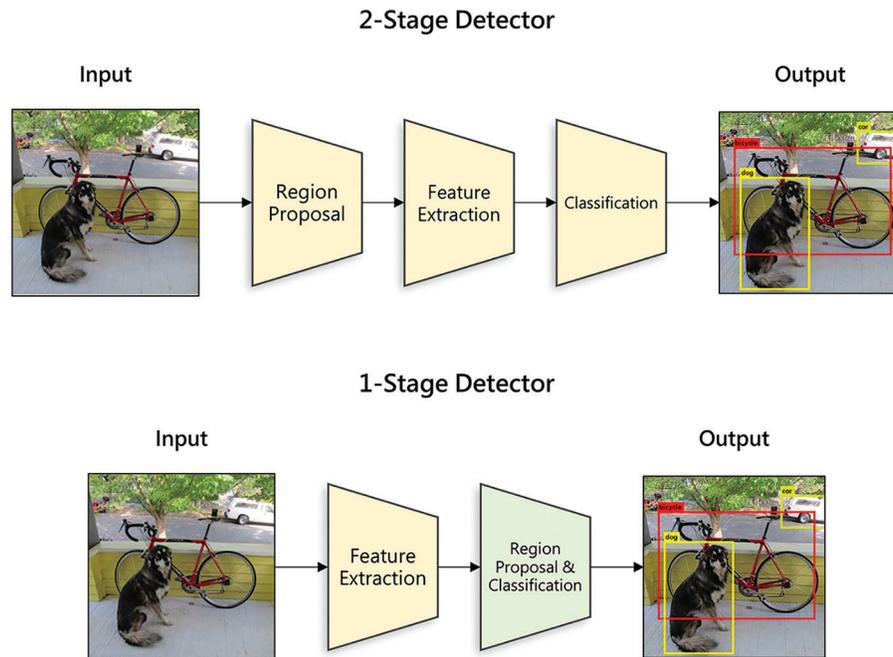
In computer vision, there have been several attempts to solve the problem of recognizing and detecting various objects, such as faces, people, and vehicles, for a long time. In traditional object detection approaches, a feature engineering technique that detects an object using a predefined object feature has been widely used. Various methods, such as histogram of oriented gradient (HOG) [13], scale-invariant feature transform (SIFT), local binary pattern (LBP) [14], and deformable part-based model (DPM) [15], have been developed for feature extraction. However, useful features cannot be extracted without a sufficient understanding of the image data; these limitations are surmountable. Fortunately, with the recent development of graphics processing unit (GPU) computing resources, it has become much easier to collect large amounts of data [16]. In addition, due to the integration of continuous research efforts in the field of neural networks, it has become possible to utilize convolutional neural network (CNN)-based learning for feature extraction. Since CNN can learn hierarchical feature representations, it improves performance than traditional feature extraction methods. In particular, since feature extraction must be accurate for object detection to be performed properly, CNN, which achieves state-of-the-art with excellent performance, has been more actively used in the field of object detection.

Most recent object detection models operate based on neural networks and generally consist of the following three stages.

- (i) Region proposal to find out where a specific object exists in the image,
- (ii) Feature extraction to extract features of each proposed region,
- (iii) Classification to identify the class to which objects belong by applying a classifier to the extracted features

Current object detection techniques can be divided into two categories such as 2-stage detector and 1-stage detector (Fig. 1). The detectors are categorized according to whether the region proposal and classification process are performed in a sequential or integrated manner. 2-stage detector sequentially performs region proposal and classification, and R-CNN series models, such as RCNN [17], Fast-RCNN [18], and Faster-RCNN [19], are representative examples of the 2-stage detector. Although these 2-stage detectors showed better performance in terms of object detection accuracy, the 2-stage detector showed a limitation in that their speed (FPS) was rather slow. In order to improve the detection speed, 1-stage

detector that integrates the region proposal and classification steps have been proposed. The representative technologies are YOLO [20], SSD [21], and Retina-Net [22], and they are widely used in real-time applications such as a biomedical service [23] and assistive system for visually impaired people [24].



**Figure 1:** 2-stage detector and 1-stage detector

Night object detection is a technique that detects objects in a difficult environment with various noises such as darkness, fog, and light blur. Night object detection is used in various fields such as border surveillance, night CCTV detection, and autonomous driving during nighttime. In particular, since it is very difficult to identify an object in a night image, studies are being conducted to facilitate object identification by converting an image or removing a specific noise. Representative examples of these studies include a study that improved object detection performance by converting a night image into a day image using variational autoencoder (VAE) and generative adversarial networks (GAN) [25]. A study performed object detection by solving the low-illumination [26] problem with RFB-net [27].

## 2.2 Image Denoising

Noise refers to unintentional distortion in data. For example, noise in image data may be caused by various environmental conditions, such as camera performance, light intensity, and weather. Noise degrades data quality, negatively affecting the performance and accuracy of algorithms. A series of noise removal processes would remove the negative effect; we call this process denoising.

Image denoising aims to remove as much noise as possible while maintaining as much original information as possible. In other words, image denoising aims to effectively remove noise while maintaining information on corners, curves, and textures, which are essential elements of an image. Image denoising is a basic and essential task in various image processing fields, such as image restoration, segmentation, and classification.

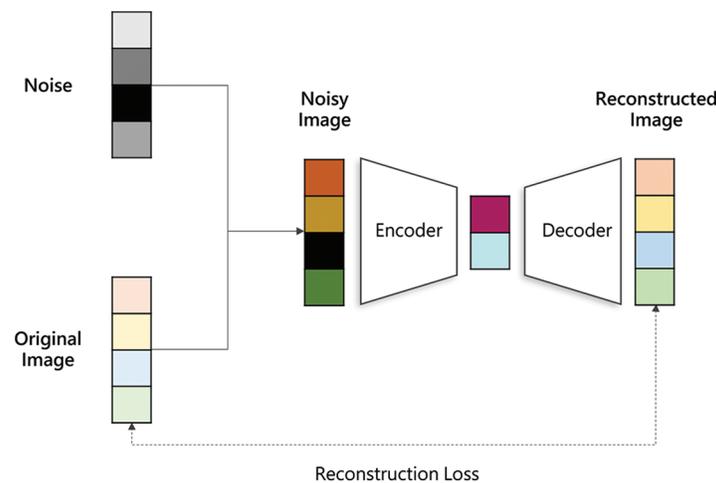
The recently proposed deep learning-based image denoising techniques show much better performance than the classical methods such as spatial domain filtering, total variation (TV)-based regularization, and

sparse representation [28]. Many studies suggesting image denoising based on deep convolutional neural network (DCNN) [29], and studies suggesting ways to improve image quality using a convolutional autoencoder and skip connection [30]. In addition, there have been studies that individually define noise elements such as fog and darkness, and suggest ways to remove each noise by using various neural network models [31].

### 2.3 Convolutional Denoising Autoencoder

Convolutional denoising autoencoder (CDAE) is a neural network-based denoising technique consisting of an encoder and decoder. The encoder extracts important features of the input data in a compressed form, and the decoder restores them. In CDAE, CNN was applied to the encoder and decoder to precisely extract the features of the image, thus, improving the ability to reconstruct the image.

CDAE accepts noise image, which is an image with noise added to the original image, as input. Representative examples of the added noise include gaussian, impulse, and uniform noises. The encoder of CDAE compresses the noisy image, and the decoder generates a reconstructed image by restoring the compressed information. CDAE repeatedly performs learning to minimize the difference between the reconstructed and original images, thus removing noise (Fig. 2).



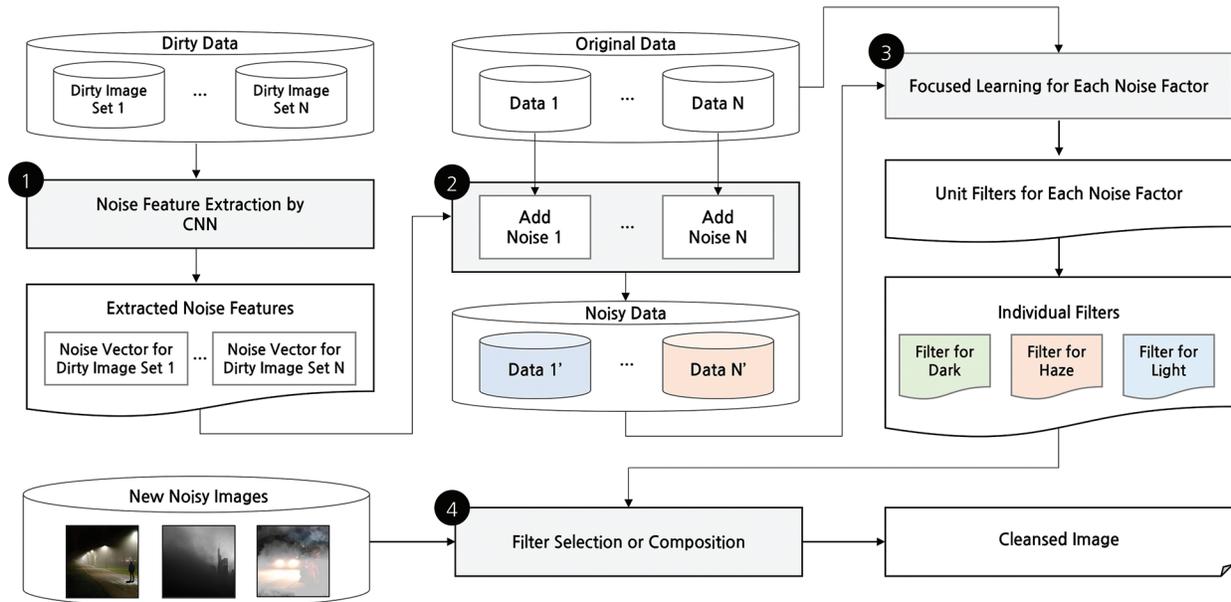
**Figure 2:** Architecture of convolutional denoising autoencoder (CDAE)

Recently, studies on improving performance by adding skip connections to the encoder and decoder of CDAE [32] have been actively conducted. These studies aim to minimize information loss by directly delivering through the encoding process to the decoder with a skip connection. Representative examples of these studies include a study applied to image restoration and super-resolution by adding skip connections to the convolutional and deconvolutional layers and a study to propose an image denoising model adding skip connections to all layers of CDAE.

## 3 Proposed Architecture

The overall architecture of the proposed scheme is shown in Fig. 3. First, we construct a dirty image set for each major noise at night, and extract the features of each noise from each data set (phase 1, subsection 3.1). Next, we prepare a noise data set for each noise by inserting each noise feature into additional image dataset (phase 2, subsection 3.2). Next, an individual filter for each noise factor is built through CDAE (phase 3, subsection 3.3). Finally, we perform denoising by selecting one optimal filter among several

individual filters (Top-one architecture) or applying several individual filters sequentially (Sequential architecture). Another architecture dealt with in this study, Integrated architecture, creates only one filter without creating individual filters for each noise in phase 3 of Fig. 3. The three denoising architectures are introduced in detail in each subsection of this section.



**Figure 3:** Overall process of the proposed scheme

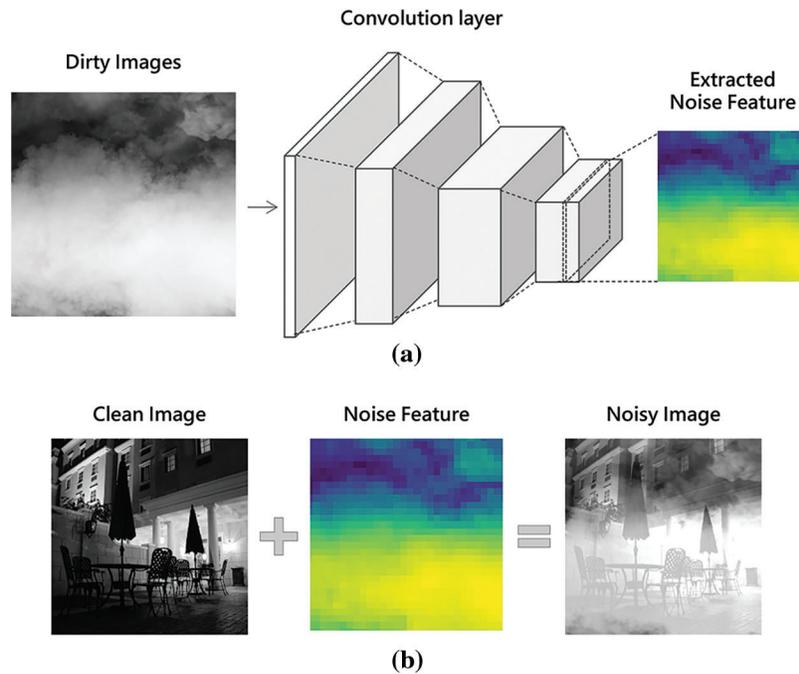
### 3.1 Creating Noisy Images

This subsection introduces the process corresponding to phases 1 and 2 of Fig. 3, extracting each noise feature and creating a noisy image by inserting these noise features into new images. Phase 1 requires images containing various noises, called dirty images. We extracted common features of each noise in dirty images using CNN. In the case of CNN, we extracted the feature value of each noise from the last convolutional layer, considering the characteristic that the deeper the layer, the more unique features of the image are identified. Fig. 4a shows an example of extracting and visualizing features of haze noise from the dirty image.

Next, phase 2 creates a noisy dataset by inserting each noise feature acquired in phase 1 into additional images. We perform this process to construct the training and verification datasets for performance evaluation. First, a clean image and a noise feature are expressed as matrices of the same size to generate a noise image. Then, a noise image is generated by performing an additional operation between the two matrices. Fig. 4b shows an example of artificially creating a noisy image by inserting the noise feature extracted from Fig. 4a into a clean image.

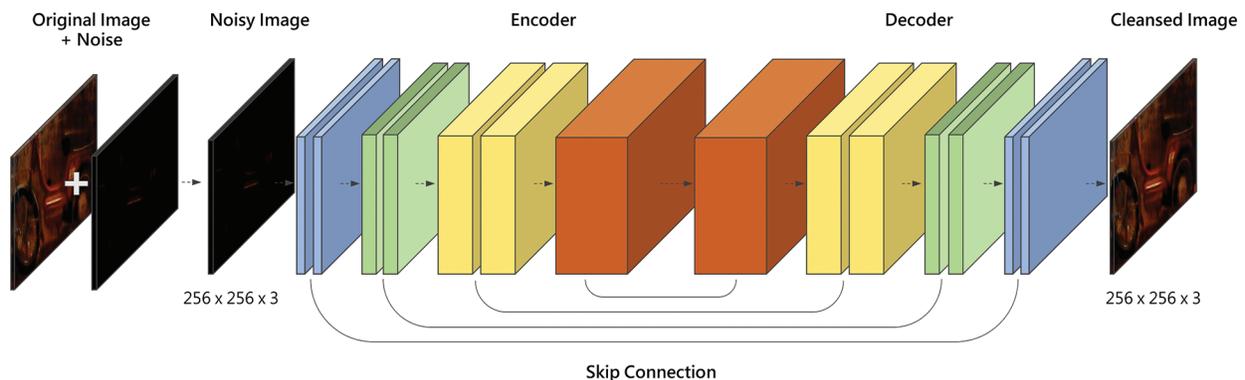
### 3.2 Construction and Application of Filtering Architectures

This subsection introduces the processes corresponding to phase 3 and phase 4 of Fig. 3. First, phase 3 utilizes convolutional denoising autoencoder (CDAE) to create a filter for a given noise. Specifically, CDAE creates filters by learning to restore original data from noisy data in Fig. 3. CDAE consists of an encoder and a decoder that are symmetrical. This study applied five skip connections to the basic CDAE to minimize interlayer information loss.



**Figure 4:** Generation of noisy images. (a) feature extraction of noise, (b) add extracted noise

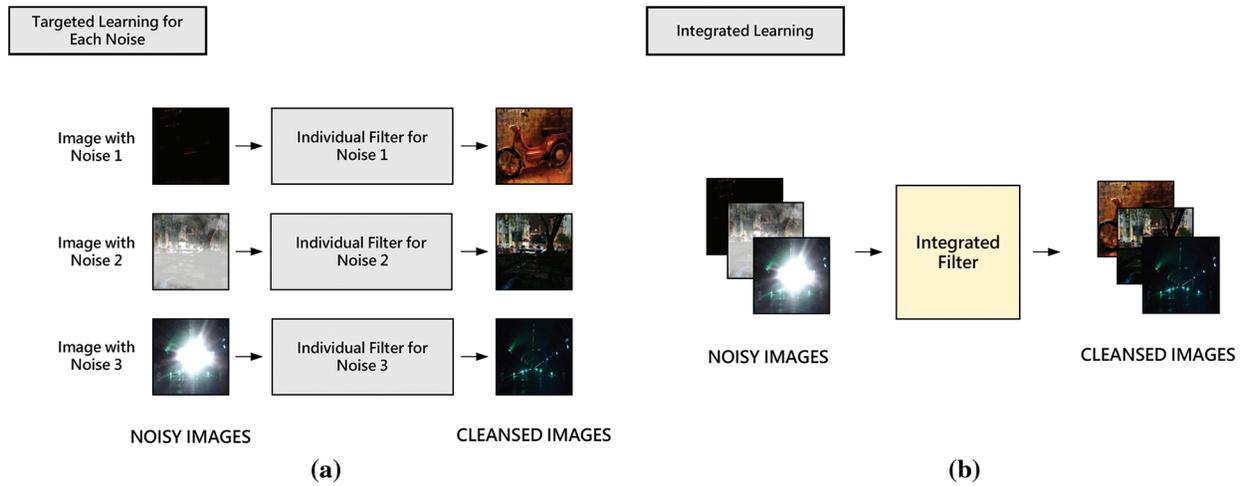
CDAE receives noise image as input and first learns important features of the input image from the encoder, a convolutional part. The decoder, a deconvolutional part, learns to reconstruct the original image from the feature values obtained through the encoder. After this learning, CDAE becomes a filter that receives noisy images as input and performs denoising to generate a clean image. Learning can be performed individually for each noise or performed for multiple noises at once (Fig. 5). Targeted learning (Fig. 6a) performs denoising on images with the same type of noise; thus, creating a filter for the noise. Meanwhile, Fig. 6b shows integrated learning in which images with different noises are integrated into one dataset, and learning is performed on them at once.



**Figure 5:** Structure of CDAE with skip connections

This study implements denoising using the above two approaches and compares their performance through experiments. In particular, we implement the target learning approach in two ways, depending on how we apply individual filters to denoising. Consequently, we propose three filtering architectures:

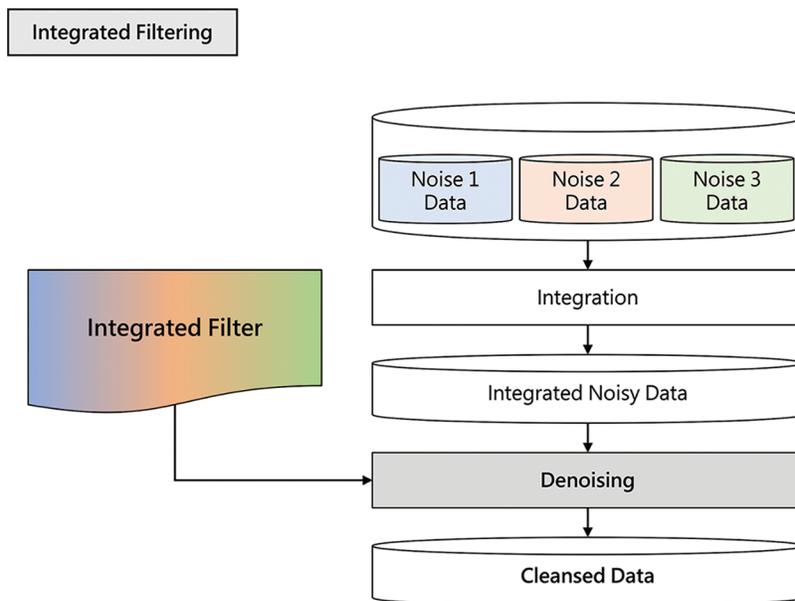
top-one filtering and sequential filtering based on the target learning approach and integrated filtering based on the integrated learning approach.



**Figure 6:** Two approaches of filter creations

*3.2.1 Integrated Filtering*

This subsection introduces the integrated filtering architecture based on the integrated learning approach. Integrated filtering builds an integrated dataset by integrating the entire noisy image into one dataset without considering the type of noise in each noisy image. Next, we perform training for denoising this integrated dataset to create an integrated filter. Fig. 7 shows the process of performing inference using the integrated filter.

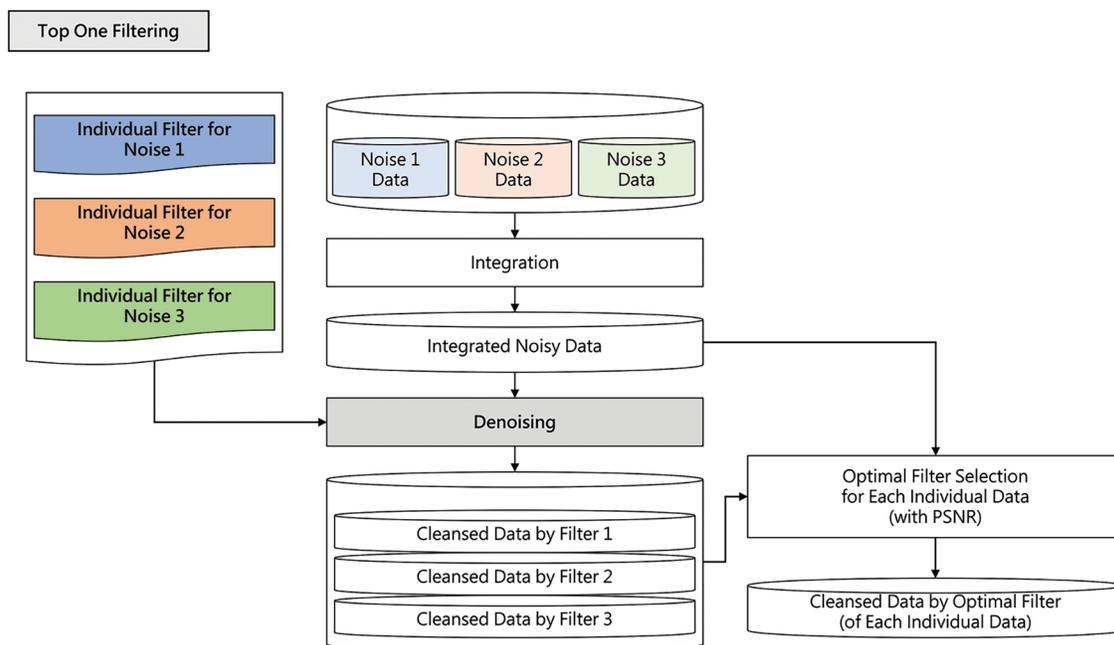


**Figure 7:** Integrated filtering architecture

As shown in Fig. 7, the integrated filtering architecture performs denoising by applying only one filter, the integrated filter, to all noisy data. In other words, all images for inference are integrated into one dataset, integrated noise data, without considering the type of noise in each image. Next, denoise by applying an integrated filter to obtain cleansed data from integrated noise data. The integrated filtering architecture is the simplest denoising structure that does not use any feature of each noise. This study is uses this architecture as a baseline for performance comparison with other architectures.

### 3.2.2 Top One Filtering

This part introduces the top-one filtering architecture of target learning approaches. The target learning approach uses several individual filters created through denoising learning for each noise type. The top-one filtering architecture denoises by selecting a filter to perform optimal denoising for a given image among these individual filters. Fig. 8 shows the process of performing inference in the top-one filtering architecture.

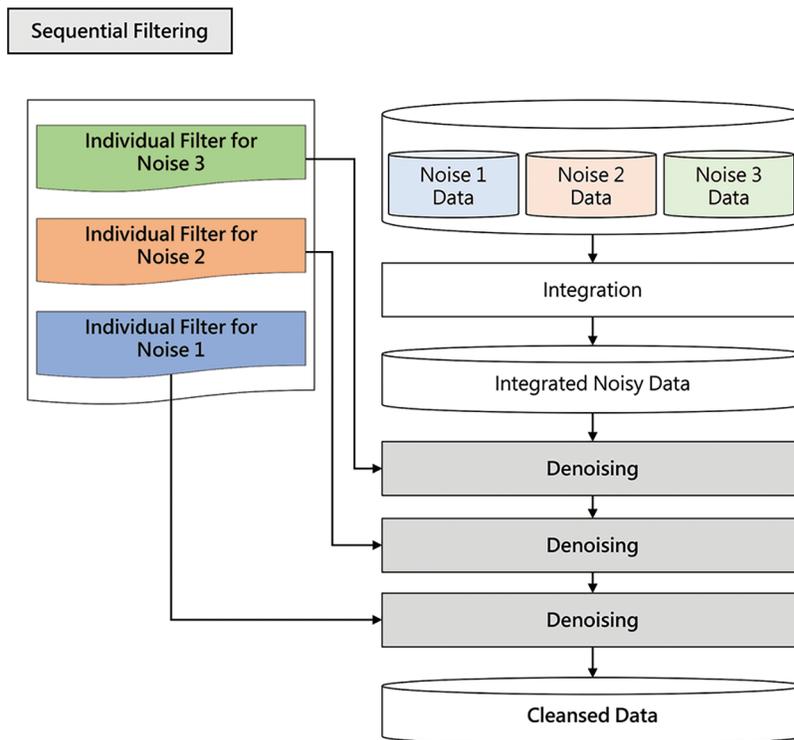


**Figure 8:** Top one filtering architecture

As shown in Fig. 8, the process of building integrated noise data is the same as in Fig. 7. However, unlike the integrated filtering architecture, the top-one filtering architecture applies three individual filters to each noisy image. Through this, we can obtain three cleansed images for each noisy image. This architecture selects only the cleanest image among the three cleansed images by utilizing a peak signal-to-noise ratio (PSNR), and this process is named as optimal filter selection for each individual data in Fig. 8. The same process is repeatedly applied to all noisy images. Then, only the images selected as the cleanest image among the resulting cleansed images are collected to construct the cleansed data by optimal filter dataset.

### 3.2.3 Sequential Filtering

This subsection introduces the sequential filtering architecture of target learning approaches. Sequential filtering architecture performs denoising by sequentially applying individual filters generated by target learning to a given noisy image. Fig. 9 shows the process of performing inference in sequential filtering architecture.



**Figure 9:** Sequential filtering architecture

As shown in Fig. 9, the process of building integrated noise data is the same as in Fig. 8. However, unlike the top-one filtering architecture, the sequential filtering architecture applies all three individual filters sequentially to each noisy image. There are six combinations of sequentially combining the three filters. The denoising performance differs depending on the order of combining the filters. This study found the optimal filter combination sequence through a pilot experiment. According to the result, we sequentially applied the filter for Noise 3 (dark), filter for Noise 2 (haze), and filter for Noise 1 (light blur) to perform denoising.

## 4 Experiment

### 4.1 Construction and Application of Filtering Architectures

This section presents the experimental process and results for verification of the proposed architectures, as presented in Section 3. In the subsection, we describe the data and experimental environment used in the experiment (Tab. 1). In this experiment, we constructed the original data of Fig. 3 using the exclusively dark (ExDark) image dataset, which is data for object detection in a low-light environment. Data for target learning consists of 4,800 cases for training, 1,200 cases for verification, and 1,190 cases for the test. Meanwhile, data for integrated learning consists of 14,400 cases for training, 3,600 cases for verification, and 1,190 cases for the test. Since the target learning learns three individual filters, the data used for training and validation of the target learning and integrated learning is the same as 18,000 cases. Each image has a size of  $256 \times 256$  and has a total of 12 object classes, such as bicycles (9%), buses (7%), cars (9%), people (8%), etc. Each image has a size of  $256 \times 256$  and 12 classes. We built an experiment environment using Python 3.7 on Tensorflow and Pytorch deep learning frameworks and performed experiments using hardware resources, such as CPU 8 Core and Tesla V100 GPU.

**Table 1:** Experimental data and environments

Experimental data			
Domain	A collection of low-light images from night to dawn		
	Train	Validation	Test
Target learning	4,800	1,200	1,190
Integrated learning	14,400	3,600	1,190
Total number of classes	12 classes		
Image size	256 × 256		
Software environment			
Hardware	CPU	8 core	
	GPU	Tesla V100 * 2ea	
	Memory	60GB	
Language	Python	3.7	
	Tensorflow	2.3.0	
Deep learning framework	Pytorch	1.7.1+cu101	

#### 4.2 Construction and Application of Filtering Architectures

This subsection presents the results of generating noisy data by inserting noise features extracted in Phase 1 of the proposed architectures into the original data. We conducted an experiment using the three main noises at night: darkness, fog, and light blur. After extracting the unique features of these noises using CNN, we generated noise data by adding the extracted features to each image in the original data. Fig. 10 shows some examples of the noisy data generated in this way.

Fig. 10 shows an example of noisy images generated by inserting three noise features of (a) dark, (b) haze, and (c) light blur for four original images. Three types of filtering architectures perform denoising on noisy data, a set of noisy images generated in this way.

#### 4.3 Performance Comparison of Three Filtering Architectures

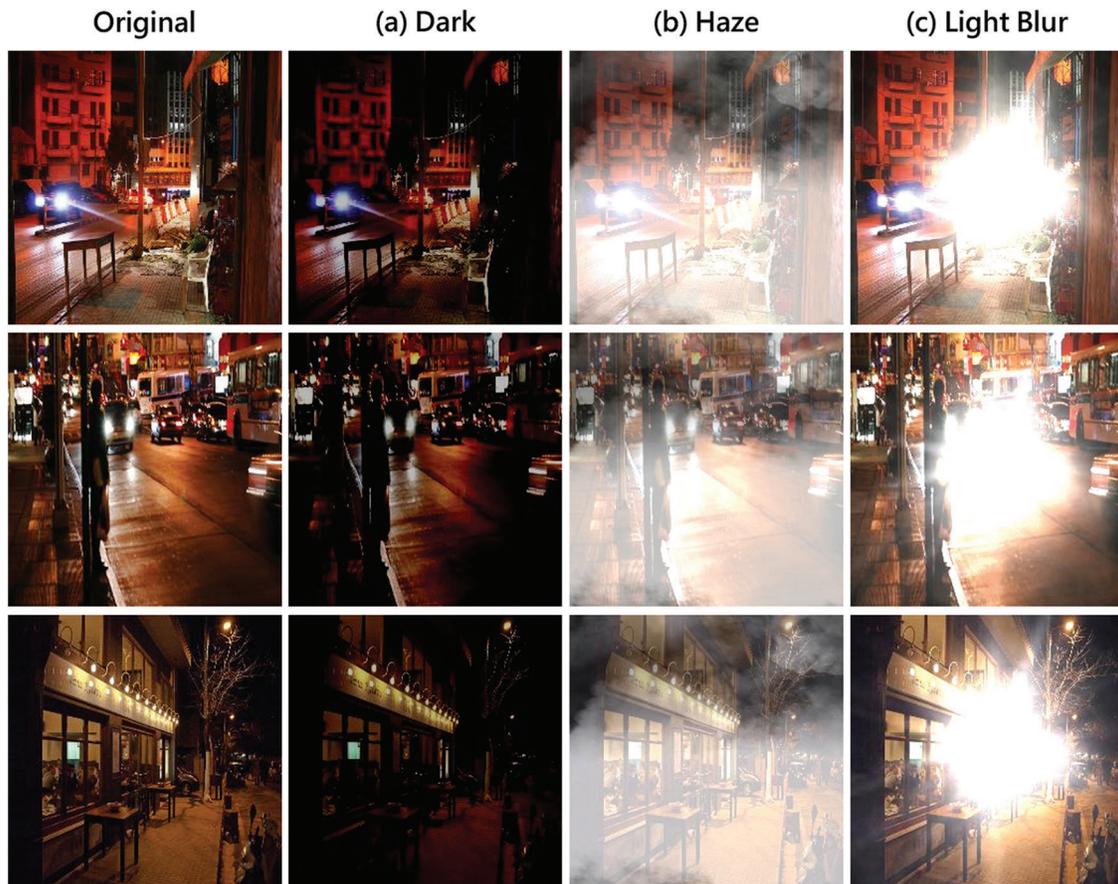
This subsection analyzes the performance of the three filtering architectures proposed in Section 3: integrated filtering, top-one filtering, and sequential filtering. We evaluated performance by comparing the image quality's improvement through each filtering architecture. We indirectly evaluated image quality by measuring object detection accuracy for the image. In this experiment, we used two evaluation indicators for performance evaluation: PSNR (an image quality evaluation indicator) and the mean average precision (mAP) (an object detection performance evaluation indicator).

PSNR is one of the representative indicators for evaluating image quality. It is used to determine the quality of the reconstructed image compared to the original image. The PSNR value is calculated using Eq. (1).

$$\text{PSNR}(x, \hat{x}) = 10 \log_{10} \left( \frac{\text{MAX}^2}{\text{MSE}} \right) \quad (1)$$

In Eq. (1),  $x$  represents the original image and  $\hat{x}$  represents the reconstructed image. MAX represents the maximum value of the image pixel values, and MSE represents the mean square error of the original image  $x$

and the reconstructed image  $\hat{x}$ . According to Eq. (1), the PSNR value increases as the MSE value, which represents the difference between the original image and the reconstructed image, becomes smaller. In this experiment, the average PSNR of the Cleansed Images generated through three filtering architectures is compared based on the original image. Through this, we can verify which filtering architectures generate the cleansed images that are most similar to the originals by denoising.



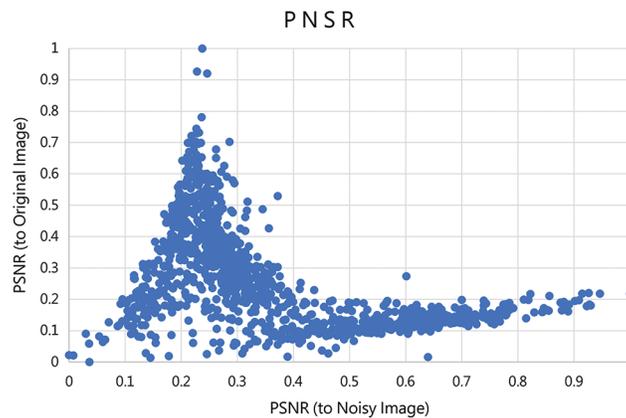
**Figure 10:** Noisy images generated by noise feature insertion

Tab. 2 summarizes the PSNR values of noisy and cleansed data generated by three filtering architectures. As presented in Tab. 2, all three versions of the cleansed data had higher PSNR values than the noisy data without applying any noise removal scheme. Additionally, among the three filtering architectures, the PSNR value was high in the order of integrated filtering, top-one filtering, and sequential filtering.

**Table 2:** PNSR of noisy images and cleansed images

Noisy data		10.05
Cleansed data	Integrated filtering	<b>24.01</b>
	Top One filtering	16.96
	Sequential filtering	12.53

This experiment used PSNR to perform denoising by applying top-one filtering architecture. It means that in the case of top-one filtering, one of the filters for each of the three noises must be selected. At this time, the PSNR value of the image generated by each filter is used. However, only noisy images are given in a real situation, and original images are not given. Therefore, an issue arises regarding which reference image we should compare with the cleansed image to calculate the PSNR value. To solve this problem, we conducted additional experiments to analyze the relationship between the PSNR of the cleansed image to the noisy image and the PSNR of the cleansed image to the original image. The results are presented in Fig. 11.



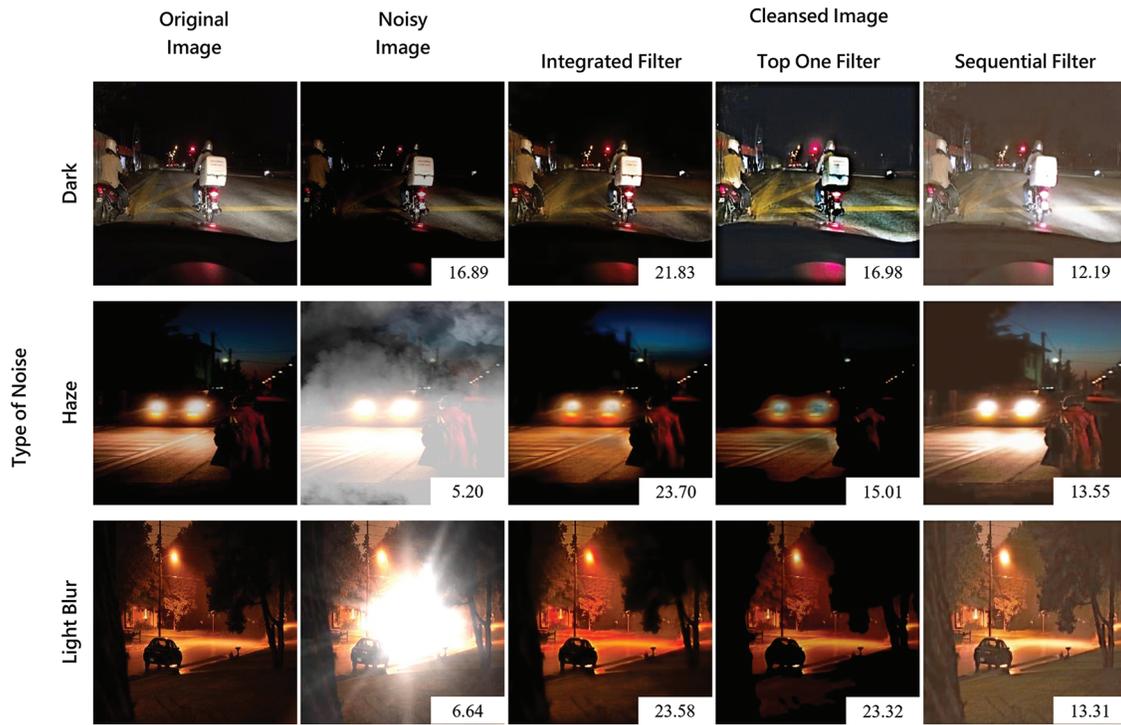
**Figure 11:** Relation of PSNR to noisy image and PSNR to original image

As shown in Fig. 11, as PSNR to noise image increases, PSNR to original image gradually increases and then decreases. This result can be interpreted as the larger the PSNR to noise image value, the more noise remains. Additionally, when the PSNR to noise image value is too small, noise and essential characteristics of the original image are lost. In this experiment, the value of PSNR to the original image was highest when PSNR to noise image was near 0.2. Therefore, to select the optimal filter in the top-one filtering, we selected the filter that generated the image showing the PSNR to noise image value closest to 0.2 as the optimal filter.

Fig. 12 shows the results of denoising through three filtering architectures with the original and noisy images. In the case of sequential filtering, the three filters can be combined in various orders. However, in this experiment, the sequential filters in the order of filter for dark, filter for haze, and filter for light blur were applied.

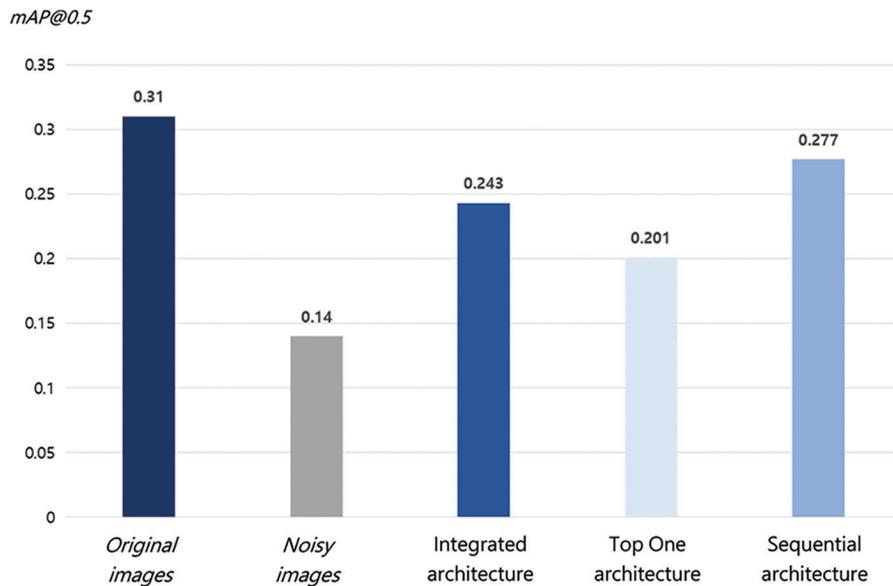
In the sample images shown in Fig. 12, the PSNR of the image cleansed through the integrated filter was the highest, which is consistent with the results in Tab. 2. However, since high PSNR values and object detection accuracy are separate issues, object detection performance should be evaluated. Therefore, we performed object detection by applying the YOLO v5 model, which currently recognized as a state-of-the-art (SOTA), to the cleansed data generated through the three filtering architectures and measured the performance using the mAP index. The mAP index is widely used to evaluate the performance of an object detection algorithm. It is an average of APs, which are the detection accuracy of each class (Eq. 2).

$$\text{mAP} = \frac{1}{n} \sum_{k=1}^{k=n} \text{AP}_k \quad (2)$$



**Figure 12:** Example of cleansed images and their PSNR

Fig. 13 summarizes the object detection accuracy for noise, original, and cleansed data generated through the three filtering architectures. In Fig. 13, sequential, integrated, and top-one filtering architectures showed high mAP in the order, and all three architectures showed significantly higher mAP than noise images. This implies that the object detection accuracy has improved through denoising.



**Figure 13:** Performance evaluation for object detection

The reason for different object detection performances in the three architectures can be found in the different types and amounts of noise that each architecture handles. First, the integrated filtering architecture learns denoising by integrating three types of noisy images into one. Thus, this architecture can denoise all three types of noise contained in a newly given image. Next, since the sequential filtering architecture also applies all three filters, it is possible to remove all three types of noise included in the newly given image. In particular, since this architecture performed learning and inference by focusing on one type of noise at a time, it showed better object detection performance than the integrated filtering architecture, which performed learning and inference on three types of noise at once. Finally, top one filtering architecture showed the lowest object detection accuracy among the three architectures. This can be interpreted as the reason that this architecture performs denoising only one type of noise for each image and does not attempt to improve other types of noise in the given image.

## 5 Conclusion

In the field of object detection, various studies are being actively conducted on daytime images, which are easy to detect objects. Unfortunately, object detection studies on night images have not been sufficiently conducted. Since various noises, such as lighting and weather, degrade the image quality in images taken at night, it is necessary to remove these noises through appropriate processing in order to improve the performance of object detection. Therefore, in order to improve the object detection performance of night images, we proposed three denoising architectures in this paper to remove major noises that occur at night.

In this paper, we proposed denoising architectures of various structures and presented their performance evaluation results. Experimental results revealed that the sequential filtering architecture showed better mAP than other denoising architectures. Academically, we proposed a method of creating individual filters for each noise and combining them, and it is expected that follow-up studies to effectively construct these individual filters will be actively performed in the future. Practically, the various denoising architectures proposed in this study can be used to improve the performance of image segmentation and classification as well as object detection. Additionally, our proposed denoising architectures are expected to improve the performance of applications in situations where various night noises exist, such as border surveillance and night autonomous driving.

The limitations of this study and future research directions are as follows. In our experiments, we analyzed noisy images by artificially inserting some noises into normal images. However, we need to perform more intensive experiments using actual noisy images because there may be various noises other than dark, haze, and light blur in real-world images. In this study, the quality of denoising was also indirectly evaluated using mAP, which measures object detection accuracy. Future studies should conduct intensive experiments with various methods to evaluate the denoising quality directly.

**Funding Statement:** This work was supported by the Ministry of Education of the Republic of Korea and the National Research Foundation of Korea (NRF-2021S1A5A2A01061459).

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

## References

- [1] Z. Zou, Z. Shi, Y. Guo and J. Ye, "Object detection in 20 years: A survey," 2019. [Online]. Available: <https://arxiv.org/abs/1905.05055>.
- [2] W. Sun, L. Dai, X. Zhang, P. Chang and X. He, "RSOD: Real-time small object detection algorithm in UAV-based traffic monitoring," *Applied Intelligence*, vol. 264, no. 6, pp. 1–16, 2021.

- [3] L. Zhang, L. Lin, X. Liang and K. He, "Is faster rcnn doing well for pedestrian detection?," in *European Conf. on Computer Vision*, vol. 9906, pp. 443–457, 2016.
- [4] W. Sun, G. Dai, X. Zhang, X. He and X. Chen, "TBE-Net: A three-branch embedding network with part-aware ability and feature complementary learning for vehicle re-identification," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–13, 2021.
- [5] L. Malburg, M. P. Rieder, R. Seiger, P. Klein and R. Bergmann, "Object detection for smart factory processes by machine learning," *Procedia Computer Science*, vol. 184, pp. 581–588, 2021.
- [6] P. A. Dhulekar, S. T. Gandhe, N. Sawale, V. Shinde and S. Khute, "Surveillance system for detection of suspicious human activities at war field," in *Int. Conf. On Advances in Communication and Computing Technology (ICACCT)*, Sangamner, India, pp. 357–360, 2018.
- [7] X. Chen, H. Ma, J. Wan, B. Li and T. Xia, "Multi-view 3d object detection network for autonomous driving," in *Proc. of the IEEE conf. on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, pp. 1907–1915, 2017.
- [8] J. Wang, T. Zhang, Y. Cheng and N. Al-Nabhan, "Deep learning for object detection: A survey," *Computer Systems Science and Engineering*, vol. 38, no. 2, pp. 165–182, 2021.
- [9] A. Singha and M. K. Bhowmik, "Moving object detection in night time: A survey," in *2019 2nd Int. Conf. on Innovations in Electronics*, Shillong, India, pp. 1–6, 2019.
- [10] O. Ronneberger, P. Fischer and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Int. Conf. on Medical Image Computing and Computer-Assisted Intervention*, Munich, Germany, pp. 234–241, 2015.
- [11] K. He, J. Sun and X. Tang, "Single image haze removal using dark channel prior," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, pp. 2341–2353, 2010.
- [12] S. Kuanar, K. R. Rao, D. Mahapatra and M. Bilas, "Night time haze and glow removal using deep dilated convolutional network," 2019. [Online]. Available: <https://arxiv.org/abs/1902.00855>.
- [13] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *Computer Vision and Pattern Recognition*, vol. 1, pp. 886–893, 2005.
- [14] S. Liao, X. Zhu, Z. Lei, L. Zhang and S. Z. Li, "Learning multi-scale block local binary patterns for face recognition," in *Int. Conf. on Biometrics*, Berlin, Heidelberg, vol. 4642, pp. 828–837, 2007.
- [15] P. Felzenszwalb, D. McAllester and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," *Computer Vision and Pattern Recognition*, Anchorage, AK, USA, pp. 1–8, 2008.
- [16] Y. LeCun, Y. Bengio and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436, 2015.
- [17] R. Girshick, J. Donahue, T. Darrell and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Columbus, OH, USA, pp. 580–587, 2014.
- [18] R. Girshick, "Fast r-cnn," in *Proc. of the IEEE Int. Conf. on Computer Vision*, Santiago, Chile, pp. 1440–1448, 2015.
- [19] S. Ren, K. He, R. Girshick and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in Neural Information Processing Systems*, Montreal, Quebec, Canada, vol. 39, pp. 91–99, 2015.
- [20] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, pp. 779–788, 2016.
- [21] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed *et al.*, "Ssd: Single shot multibox detector," in *European Conf. on Computer Vision*, Amsterdam, The Netherlands, pp. 21–37, 2016.
- [22] T.-Y. Lin, P. Goyal, R. Girshick, K. He and P. Doll'ar, "Focal loss for dense object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, pp. 2980–2988, 2018.
- [23] U. Masud, F. Jeribi, A. Mohammad, F. Akram and T. Tahir, "Two-mode biomedical sensor build-up: Characterization of optical amplifier," *Computers Materials & Continua*, vol. 70, no. 3, pp. 5487–5489, 2021.

- [24] U. Masud, T. Saeed, H. M. Malaikah, F. U. Islam and G. Abbas, "Smart assistive system for visually impaired people obstruction avoidance through object detection and classification," *IEEE Access*, vol. 10, pp. 13428–13441, 2022.
- [25] M. Schutera, M. Hussein, J. Abhau, R. Mikut and M. Reischl, "Night-to-day: Online image-to-image translation for object detection within autonomous driving by night," *IEEE Transactions on Intelligent Vehicles*, vol. 6, no. 3, pp. 480–489, 2021.
- [26] K. G. Lore, A. Akintayo and S. Sarkar, "LLNet: A deep autoencoder approach to natural low-light image enhancement," *Pattern Recognition*, vol. 61, no. 6, pp. 650–662, 2017.
- [27] S. Liu and D. Huang, "Receptive field block net for accurate and fast object detection," in *Proc. of the European Conf. on Computer Vision*, Munich, Germany, pp. 385–400, 2018.
- [28] L. Fan, F. Zhang, H. Fan and C. Zhang, "Brief review of image denoising techniques," *Visual Computing for Industry*, vol. 2, no. 1, pp. 1–12, 2019.
- [29] A. Khan, A. Sohail, U. Zahoor and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *Artificial Intelligence Review*, vol. 53, no. 8, pp. 5455–5516, 2020.
- [30] X. J. Mao, C. Shen and Y. B. Yang, "Image restoration using convolutional auto-encoders with symmetric skip connections," 2016. [Online]. Available: <https://arxiv.org/abs/1606.08921>.
- [31] M. Feng, T. Yu, M. Jing and G. Yang, "Learning a convolutional autoencoder for nighttime image dehazing," *Information-an International Interdisciplinary Journal*, vol. 11, no. 9, pp. 42, 2020.
- [32] L. F. Dong, Y. Z. Gan, X. L. Mao, Y. B. Yang and C. Shen, "Learning deep representations using convolutional auto-encoders with symmetric skip connections," in *2018 IEEE Int. Conf. on Acoustics*, Calgary, AB, Canada, pp. 3006–3010, 2018.