CF-YOLO: Cross Fusion YOLO for Object Detection in Adverse Weather With a High-Quality Real Snow Dataset

Qiqi Ding, Peng Li, Xuefeng Yan, Ding Shi, Luming Liang, Weiming Wang, Haoran Xie, *Senior Member, IEEE*, Jonathan Li, *Fellow, IEEE*, and Mingqiang Wei, *Senior Member, IEEE*

***Abstract*— Snow is one of the toughest adverse weather conditions for object detection (OD). Currently, not only there is a lack of snowy OD datasets to train cutting-edge detectors, but also these detectors have difficulties of learning latent information beneficial for detection in snow. To alleviate the two above problems, we first establish a real-world snowy OD dataset, named RSOD. Besides, we develop an unsupervised training strategy with a distinctive activation function, called *Peak Act*, to quantitatively evaluate the effect of snow on each object. Peak Act helps grade the images in RSOD into four-difficulty levels. To our knowledge, RSOD is the first quantitatively evaluated and graded real-world snowy OD dataset. Then, we propose a novel Cross Fusion (CF) block to construct a lightweight OD network based on YOLOv5s (called CF-YOLO). CF is a plug-and-play feature aggregation module, which integrates the advantages of Feature Pyramid Network and Path Aggregation Network in a simpler yet more flexible form. Both RSOD and CF lead our CF-YOLO to possess an optimization ability for OD in real-world snow. That is, CF-YOLO can handle unfavorable detection problems of vagueness, distortion and covering of snow. Experiments show that our CF-YOLO achieves better detection results on RSOD, compared to SOTAs. The code and dataset are available at https://github.com/qqding77/CF-YOLO-and-RSOD.**

Manuscript received 11 June 2022; revised 22 November 2022 and 28 March 2023; accepted 12 May 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 62172218; in part by the Shenzhen Science and Technology Program under Grant JCYJ20220818103401003 and Grant JCYJ20220530172403007; in part by the Natural Science Foundation of Guangdong Province under Grant 2022A1515010170; in part by the Hong Kong Metropolitan University Research Grant under Grant RD/2021/09; in part by the Direct Grant under Grant DR23B2; and in part by the Faculty Research Grant of Lingnan University, Hong Kong, under Grant DB23A3 and Grant DB23B2. The Associate Editor for this article was Q. Ye. *(Corresponding author: Mingqiang Wei.)*

Qiqi Ding, Peng Li, Xuefeng Yan, and Ding Shi are with the School of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China (e-mail: qqding@nuaa.edu.cn; pengl@nuaa.edu.cn; yxf@nuaa.edu.cn; shiding@nuaa.edu.cn).

Luming Liang is with Microsoft, Redmond, WA 98052 USA (e-mail: llmpass@gmail.com).

Weiming Wang is with the School of Science and Technology, Hong Kong Metropolitan University, Hong Kong, China (e-mail: wmwang@hkmu.edu.hk).

Haoran Xie is with the Department of Computing and Decision Sciences, Lingnan University, Hong Kong, China (e-mail: hrxie@ln.edu.hk).

Jonathan Li is with the Department of Geography and Environmental Management and the Department of Systems Design Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: junli@uwaterloo.ca).

Mingqiang Wei is with the Shenzhen Research Institute, Nanjing University of Aeronautics and Astronautics, Shenzhen 518100, China (e-mail: mingqiang.wei@gmail.com).

Digital Object Identifier 10.1109/TITS.2023.3285035

***Index Terms*— CF-YOLO, RSOD dataset, snowy object detection, peak act, cross fusion.**

I. Introduction

CNN-BASED detectors heavily depend on the integrity of objects in images [1], [2], [3] to achieve remarkable performance. Unfortunately, objects are often partially or even fully covered by snow in winter, causing the loss of crucial information for describing objects under such adverse weather. Consequently, most object detection (OD) methods that perform well in normal weather often fail to operate effectively on snowy images. [4], [5], [6].

There exist two major challenges in detecting objects covered by snow. First, capturing snow/snow-free image pairs in real-world scenarios is extremely difficult or nearly impossible. Therefore, existing detectors can only be trained on the benchmark datasets captured under normal weather conditions [7], [8], [9], [10] or synthetic snow (e.g., SnowCityScapes [11]). The detectors trained on such datasets suffer from the well-known domain shift problem, and thus generalize poorly in real-world snowy scenarios. Second, existing detectors struggle to learn latent information that is beneficial for detection in snow, since snow can destroy the low-level vision information (e.g., textures and outlines) of the objects.

The performance of current vision techniques is mainly benchmarked under normal weather conditions. Even the top-performing object detectors exhibit severe performance degradation under adverse weather conditions. Therefore, we raise a practically meaningful question regarding object detection under adverse weather: Does the synergy of establishing a real-snow OD dataset and developing a feature aggregation module to learn latent information, actually enhance the capability of cutting-edge OD networks in the snowy condition?

To answer this question, i) we collect a high-quality outdoor dataset, called RSOD, for object detection in real-world snowy scenarios. RSOD contains 2100 real-world snowy images annotated in the format of COCO and YOLO (with labeled pedestrians, cars, traffic lights, etc). ii) We endeavor to quantitatively evaluate the effect of snow on each object by introducing an indicator called snow coverage rate (SCR). To calculate SCR, we develop an unsupervised training strategy to train

a simple yet effective CNN model with a distinctive activation function called *Peak Act*. The SCR is subsequently exploited to grade the images in RSOD into four-difficulty levels (i.e., easy, normal, difficult, and particularly difficult). As different snow coverage rates can exert varying effects on the performance of detectors, it is crucial to employ such gradation to understand how snow degrades the accuracy of object detection. iii) We propose a plug-and-play Cross Fusion (CF) block. Instead of relying on the traditional top-down and bottom-up structures [12], the CF block simultaneously aggregates features from different stages of the backbone. This direct fusion manner allows for the recovery of the low-level information lost during the high-level feature extraction. Besides, CF supports different in-out stages, making it a more flexible and adaptive plug-and-play module. iv) We propose a lightweight object detection network named CF-YOLO by replacing the neck of YOLOv5s with the CF blocks. Experimental results on RSOD and COCO clearly demonstrate that CF-YOLO not only exhibits superior optimization ability for OD in the real-world snowy scene but also possesses strong generalization ability.

Our main contributions are summarized as follows:

- We present a real-world snowy OD dataset (RSOD), which is labeled in both the COCO and YOLO formats. To our knowledge, this is the first dataset that focuses on improving the accuracy of object detection in real-world snowy scenarios.
- We introduce an indicator called snow coverage rate (SCR) and develop an unsupervised training strategy to train a CNN model with a distinctive activation function, called Peak Act, to evaluate the impact of snow on each object. We grade the images in RSOD into four-difficulty levels based on the SCR. Such grading aids in understanding how the snow degrades the OD performance.
- We propose a new plug-and-play Cross Fusion (CF) block to aggregate features from different stages simultaneously. Users can customize the number of stages, the number of CF blocks, and the kernel sizes in different networks to optimize the models' performance.
- We propose a lightweight and effective CF-YOLO to facilitate snowy OD applications, such that many outdoor vision systems (e.g., autonomous driving, surveillance) can operate smoothly in snowy weather.

The rest of this work is organized as follows. Section II introduces the related work from four aspects: object detection in normal weather, object detection in adverse weather, snowy datasets, and feature fusion. Section III describes the constructed real-world snowy object detection dataset RSOD. Section IV introduces the proposed CF-YOLO. Section V shows sufficient experiment results to demonstrate the effectiveness of CF-YOLO, followed by the conclusion in Section VI.

II. Related Work

In this section, we first discuss the techniques of different object detectors in normal weather. Then we introduce some existing solutions for object detection in adverse conditions. After that, we discuss the existing works and datasets on snowy weather. Finally, we make a brief comparison of different feature fusion modules.

A. Object Detection in Normal Weather

Existing object detection methods can be roughly divided into two categories: two-stage methods and one-stage methods. They usually contain a backbone for feature extraction, a neck for feature fusion, and a head for prediction. The two-stage detectors, such as R-CNN [13], fast R-CNN [14], faster R-CNN [4], R-FCN [15], and Libra R-CNN [16], first generate some regions of interest (RoIs) from images, and then classify and locate the objects in these RoIs. In contrast to the time-consuming two-stage detectors, the one-stage detectors directly infer the object categories and bounding boxes, such as the YOLO series [17], [18], [19], [20], [21], SSD [22], RetinaNet [23] and EfficientDet [24]. Among them, the YOLO series gain widespread adoption across diverse research fields and have many variants [25], [26], [27]. DETR [28] is the first model to introduce the transformer architecture into the object detection task and treats OD as a query prediction problem. Gu et al. [29] propose a homography loss to utilize 2D boxes as guidance in 3D object detection [30]. Recently, with the boom in autonomous driving, semi-supervised object detection gains considerable attention [31].

B. Object Detection in Adverse Weather

A detector trained on clean images usually fails to yield desirable results under adverse weather conditions (e.g., snowy, rainy, hazy, and low-light), due to the domain shift in input images [32]. Currently, there are mainly three solutions to alleviate this problem. The first solution is to dilute the effect of weather-specific information by a pre-processing step, such as image desnowing/derainig/dehazing [11], [33] or low-light image enhancement [34]. Although these methods exhibit satisfactory performance with strong pixel-level supervision on synthetic data, they may encounter limitation in real scenes. The second solution is to jointly perform image restoration and object detection via dual-branch networks [2], where the two branches share a feature extraction module. But balancing the two tasks during training is a significant challenge. The third solution is to exploit unsupervised domain adaptation [1] to align the features of clean images (sources) and images captured under adverse weather (targets). However, this technique may overlook the important latent information that is crucial for effective detection during the image restoration process.

C. Snowy Datasets

Due to the lack of real-world paired snow/snow-free images, existing datasets on snowy weather are generally obtained by adding snow masks to clean images, like Snow100K [35], SnowKITTI2012 [11], and SnowCityScapes [11]. Existing desnowing methods trained on these synthetic datasets usually deteriorate significantly on real-world snowy images. Meanwhile, they only remove snow in the air, ignoring the occlusion of objects by snow that substantially affects the performance

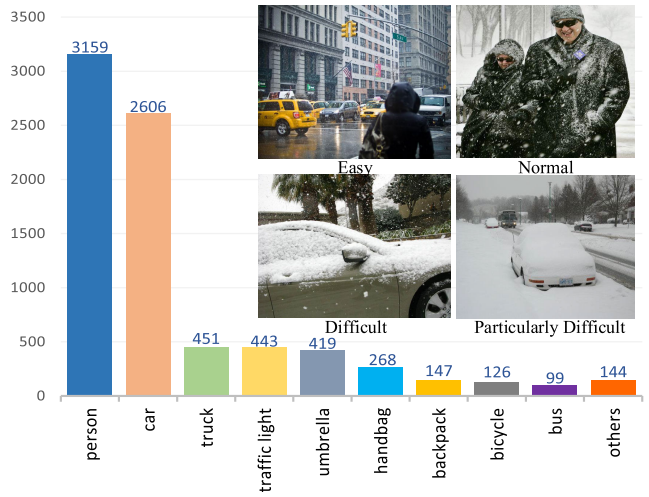


Fig. 1. The label distribution in the RSOD dataset. Most of the annotated objects are cars (2606 labeled), pedestrians (3159 labeled) and trucks (451 labeled). We divide snow images into four difficulty levels (i.e., easy, normal, difficult, and particularly difficult) based on the snow coverage rates.

of object detection. Moreover, these methods still suffer from the loss of image details, which are important cues for accomplishing high-level tasks. Therefore, they inadequately improve the performance of downstream applications.

D. Feature Fusion

Existing feature fusion methods contain FPN [12], PANet [36], NAS-FPN [37], BiFPN [24], ASFF [38], etc. FPN integrates features from different stages of the backbone through a top-down path. PANet is built on the foundation of FPN and exploits a bottom-up path to further enhance the features. The BiFPN Layer [24] is developed for easy and fast multi-scale feature fusion by bidirectional cross-scale connections. OctConv [39] performs frequency decomposition on the features to improve the efficiency of CNNs. gOctConv [40] possesses the advantage of flexible feature fusion for the arbitrary in-and-out branches. Differently. we exploit gOctConv as the fundamental component of our cross fusion block, which exhibits a good ability for feature fusion.

III. Real-World Snowy Object Detection Dataset

A. Dataset Introduction

The established real-world snow object detection dataset, called RSOD, contains 2100 images captured in various real-world snowy scenes. To make RSOD convenient to use for public studies, the labels are fully compatible with MSCOCO. We provide both the COCO and YOLO formats. Fig. 1 shows the label distribution in RSOD. Note that most of the snowy images are about townscape and traffic scenes.

Considering the fact that the performance of cutting-edge detectors will deteriorate when objects are covered under different degrees of snow, we grade the snowy images into four difficulty levels, i.e., easy, normal, difficult, and particularly difficult. This gradation aims to investigate the impact of snow on objects and study how it degrades object detection accuracy.

To conduct a fair gradation, we introduce an indicator, i.e., snow coverage rate (SCR) to judge the difficulty levels of the images in RSOD. By combining SCR and human observation, we grade image numbers 1∼600 to the easy level, 601∼1600 to the normal level, 1601∼2000 to the difficult level and 2001∼2100 to the particularly difficult level. Fig. 1 shows the typical images of different levels.

B. Unsupervised Training for SCR Calculation

Quantitatively evaluating the effect of snow on the covered objects is challenging even for humans due to the lack of benchmarks for evaluation. We assume that the influence of snow on the object depends on the snow coverage rate (SCR) in the object's bounding box. Therefore, SCR can be formulated as $SCR = A_{snow}/A_{bbox}$, where A_{snow}, A_{bbox} represent the areas of snow and bounding box, respectively.

Calculating SCR directly is challenging due to the high cost of labeling snow in the images. To address it, we develop an unsupervised training strategy to train a CNN model which can respond to snow pixels and depress non-snow pixels. The strategy contains three factors based on sparse coding.

First, to respond to snow pixels, we train a CNN model using images where heavy snow covers most of the area. The corresponding ground truth is a map of the same size as the input image, with each pixel assigned a value of 1. This step guides the model to map each pixel to the value of 1, and the convolution kernels in the model encode the snow features through backpropagation.

Second, to depress non-snow pixels, we design an activation function $Peak\ Act$, with a very narrow activation bandwidth. It allows the convolution kernels to respond to some specific features like snow, and depress other features, as shown in Fig. 2(d). Since snow covers the largest area of the images we use for training, it is natural that convolution kernels can respond to snow pixels and depress non-snow pixels.

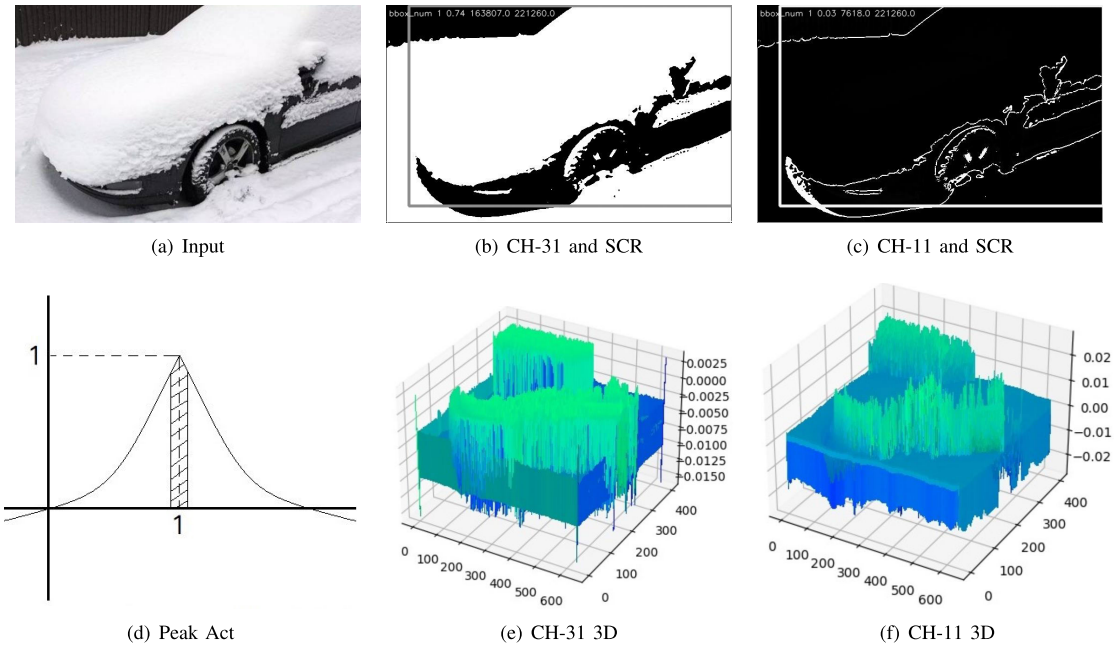
The function of Peak Act lies on the following three rules:

- Being a peak function where the peak is (1, 1). Since the ground truth is a matrix with all elements equal to 1, the training process will drive the model to output 1. And the peak constrains the effective area in a very small bandwidth (see Fig. 2(d)).
- Zero maps to zero. If zero can be mapped to a non-zero value, some *lazy* convolution kernels with all weights being equal to 0 will smooth all the pixels to a non-zero value. As a result, the subsequent layers may converge to the ground truth easily, potentially hindering the model from reaching its optimal performance.
- Being a concave function such that feature values will not get closer to 1 after passing through the activation function. The feature values can only be closer to 1 through optimization.

The proposed Peak Act is defined as:

$$f(x) = \begin{cases} 0.2x & x < 0 \\ x^2 & 0 \le x < 1 \\ (x-2)^2 & 1 \le x < 2 \\ -0.2(x-2) & x \ge 2 \end{cases} \quad (1)$$

Equation 1 shows a non-monotone increasing and continuously differentiable function whose activation area is (0, 2). Since we



(a) Input (b) CH-31 and SCR (c) CH-11 and SCR

(d) Peak Act (e) CH-31 3D (f) CH-11 3D

Fig. 2. Visualization of different channels of the feature map yielded by the CNN model. Different channels of the output feature map respond to specific features, such as snow, edges, etc. (b) is the 31st channel (CH-31) of the output feature map, which specifically responds to snow. We can utilize the snow-labeled map (CH-31) to calculate the snow coverage rate (SCR) of the object. (c) is the 11th channel of the output feature map, which responds to edges. (e) and (f) are the 3D versions of feature maps corresponding to (b) and (c), respectively.

drive the model to output 1, the real activation area is located in a narrow band around 1, as shown in the hatched section of Fig. 2(d), which we call the narrow activation bandwidth. Peak Act aims to drive the network to focus on expressing snow features. More details about the effect of Peak Act are described in Section V-A.

Third, the last layer of the CNN model is a *Max-out* function, which yields the maximum value of every pixel in the feature map along the channel dimension, and forms a one-channel feature map O. We then compute the loss to evaluate the consistency between the map and the ground truth (GT). Due to the fact that the upper limit of Peak Act is 1, the output of the network will be always less than or equal to 1. The *Max-out* layer encourages different channels to respond to different features, leading to highly specific optimizations of the kernels. We define the loss as:

$$Loss = \alpha \frac{1}{W * H} \sum_i \sum_j (GT_{ij} - O_{ij}) + \beta \|P\|_1 \qquad (2)$$

where P denotes the model's parameters, $\alpha = 1$, and $\beta = 0.0001$ are the weights to balance the two terms.

In Equation 2, the first term is the averaged pixel-wise distance between the output and the ground truth. The output of the CNN model has the same size as GT, which is $W * H$. The GT whose pixel values are all equal to 1 guides the model to map the input image $\rightarrow 1$ during training. The second term is an L_1 regularization to make the feature space sparser.

The CNN model employed for calculating the SCR is shown in Fig. 3. We exploit different heads for training and testing (i.e., calculating the SCR). Specifically, the training head is a *Max-out* layer, which reduces the number of feature map channels from 32 to 1 by taking the maximum value along the channel dimension. The output result is supervised by the loss function in Equation 2. When testing, we directly binarize the feature map generated by the last layer and select the channel that responds to snow to calculate the SCR. The selected channel marks the pixels with snow as 1. As shown in Fig. 2(b) and Fig. 2(c), CH-31 responds to snow very specifically, while CH-11 responds to edges. Furthermore, we visualize the 3D surfaces of different channels, as shown in Fig. 2(e) and Fig. 2(f), clearly showing that CH-31 responds to the snow area and depresses the non-snow area distinctly.

C. Explanation of the Unsupervised Training Strategy

We treat the snow as a set of features (i.e., snow features) embedded in the input data (snowy images), and the deep network as a mapping machine from the input data to the output. Since the ground truth of our network is constantly equal to 1, our network aims to be a mapping machine for **Input data $\rightarrow$1**.

Essentially, the training process is to optimize the parameters of the network to encode the features of input data automatically. During this process, if we feed the network with snowy images whose principal components are the snow features, the network becomes a mapping machine: **Snow feature $\rightarrow$1**, primarily.

We can also regard this process as encoding snow features into the network, which is a commonly used technique in the Sparse Coding Algorithms. In these algorithms, the ground truths are equal to the input images, and the training process involves encoding the features into the network through backpropagation and building a feature dictionary. Our unsupervised training strategy can be treated as a sparse coding process with only one dictionary item, i.e., snow.

The proposed Peak Act function and the CNN model with the unsupervised training strategy are crucial for calculating the SCR and grading the snowy images.

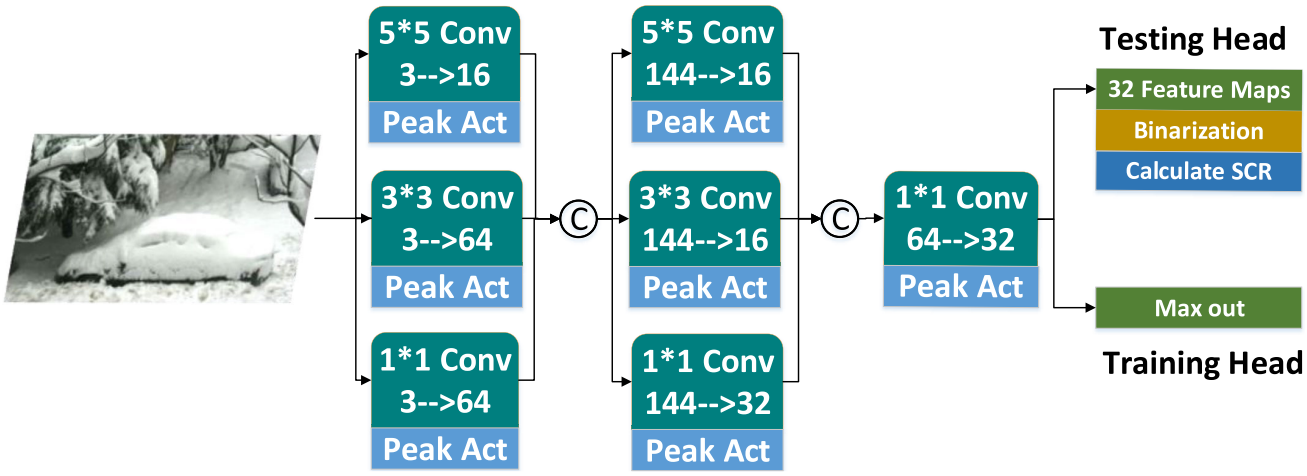


Fig. 3. The structure of the CNN model used to calculate the SCR. During training, we use the training head, i.e., the *Max-out* layer, to generate a single-channel feature map for computing the training loss. While testing (i.e., calculating the SCR), we use the testing head to yield the snow-labeled map.

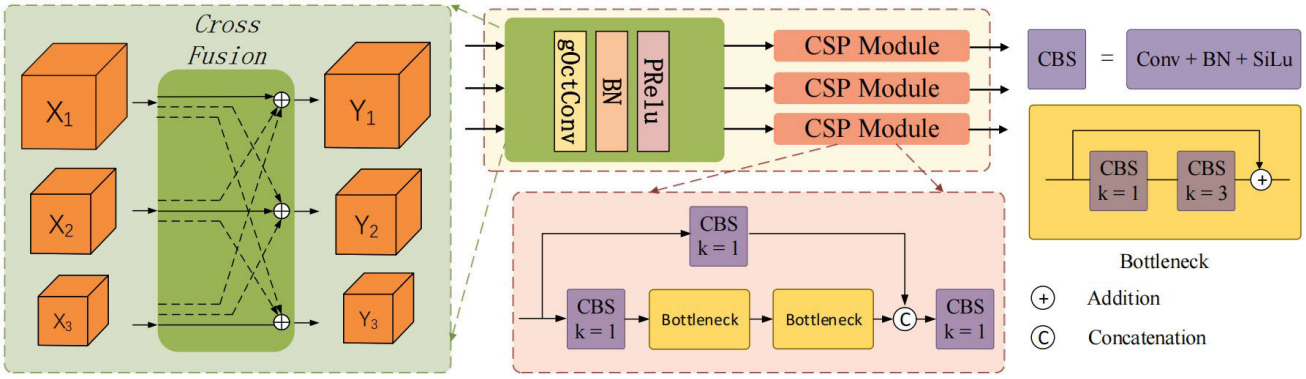


Fig. 4. The structure of Cross Fusion. X_1, X_2, X_3 indicate the inputs from different levels of the backbone. Y_1, Y_2, Y_3 indicate the output branches for different prediction heads of YOLO. The features from each level are sequentially fed into a gOctConv (without sharing weights), a Batch Normalization layer, and a PRelu activation function. The post-processing component of the CF block is Cross Stage Partial (CSP) [41].

IV. Methodology

Recently, cutting-edge detectors achieve remarkable progress and strengthen many outdoor vision systems, e.g., autonomous driving, and surveillance. But these methods suffer from various adverse weather and fail to yield desirable results. The performance of various detectors trained on MSCOCO degrades significantly on RSOD (see Table I).

Besides the enormous degradation, we observe that some large objects are more prone to be omitted by YOLOv5s in snowy images. It may contradict our common sense that detection models typically perform better on large objects. Based on this observation, we make a minor adjustment to YOLOv5s. By setting the detection confidence threshold to 0.01, we surprisingly find that YOLOv5s can indeed detect those large objects in the snowy images (similar phenomena also exist in many other snowy images), but the confidence is too low to pass the Non-Maximum Suppression (NMS) operation, leading to the mis-prediction (see Fig. 10(a)~(d)). The reason is that heavy snow can change the outline, texture and surface of the objects, resulting in the loss and distortion of low-level visual information.

Yolov5s uses Feature Pyramid and Path Aggregation Network (FPN+PANet) as the feature fusion module, where features are processed through a top-down and bottom-up route before reaching the YOLO prediction head. According to the structure of YOLOv5s, the large objects are predicted in the last stage which means that the features of large objects pass through the deepest network. In snowy images, the low-level vision information is destroyed, resulting in extensive meaningless information being fed to the network. As the number of network layers deepens, the receptive field of the model continues to expand, thereby making the model more susceptible to interference from meaningless information and even *dilute* meaningful features.

A. Cross Fusion

To address the abovementioned problems, we propose a new Cross Fusion (CF) block to directly integrate the features from different levels. The purpose of CF blocks is to shorten the propagation route instead of making the model more complicated or deeper. It can alleviate the *dilution* of meaningful features when the network goes deeper. Specifically, inspired by [40], we utilize gOctConv as the fusion component of the CF block. As shown in Fig. 4, the CF layer simultaneously receives the input feature maps from different levels of the backbone and its outputs are fed to different YOLO prediction heads. When detecting large objects, compared with the top-down and bottom-up routes of "FPN+PANet", the CF blocks enable direct access to low-level features through the simultaneous feature fusion process, thereby providing a shorter route between low-level and high-level features. The CF block also allows different in-out branches, making itself a flexible

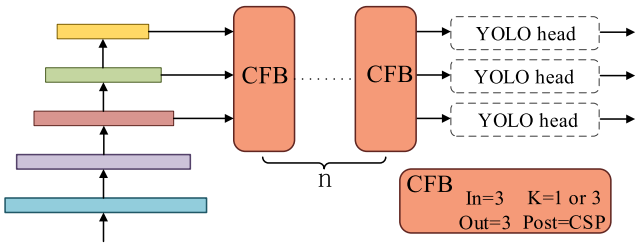


Fig. 5. The architecture of CF-YOLO. We replace the feature fusion module ("FPN+PANet") of YOLOv5s with the CF block and remain the structures of backbone and YOLO head. CF-YOLO can be modified by changing the number of CF blocks (n), in-out stages (In and Out), and the kernel sizes (K) to obtain different versions for different usages.

module to adapt to different models. The post-processing component is the Cross Stage Partial (CSP) module.

One of the feature fusion processes in CF is expressed as:

$$O_1 = f_{CSP}(Conv_{11}(X_1) \oplus Conv_{12}(Resize(X_2)) \oplus Conv_{13}(Resize(X_3))) \quad (3)$$

In Equation 3, f_{CSP} denotes the CSP module and $\oplus$ means the element-wise addition. O_1 is the up-branch of the CF outputs. Notably, the feature fusion process of the CF block occurs before the post-processing component, while the feature fusion process of "FPN+PANet" can only happen sequentially along with the top-down and bottom-up operations.

B. CF-YOLO

We propose CF-YOLO, a modified version of YOLOv5s, in which we replace the neck with the CF block (see Fig. 5). The flexibility of CF-YOLO allows for modifications by adjusting the number of CF blocks (n), in-out stages (In and Out), and the kernel size of gOctConv (K). Herein we stack two layers of CF blocks (n=2) in CF-YOLO. CF-YOLO (K=1) and CF-YOLO (K=3) denote that the kernel sizes of the CF block are equal to 1 and 3, respectively.

V. Experiments and Discussion

A. Comparison of Different Activation Functions

Since the images used to train the CNN model and calculate SCR are heavy snowy images, the statistical distribution of training data concentrates on the snow features. It encourages the network to respond to snow. To verify the effectiveness of the proposed Peak Act, we compare the Peak Act with some general activation functions, including Sigmoid, ReLU [42], and Leaky ReLU [43], and select the best feature map visualization results for comparison.

As shown in Fig. 6, the outputs of CNN models exhibit different characteristics when different activation functions are employed. Specifically, the Sigmoid function (see Column 1) cannot identify the difference between snow and non-snow areas. The best-visualized feature map is almost equal to GT, which is a matrix with all elements equal to 1. In the case of the ReLU (see Column 2) and Leaky ReLU (see Column 3) functions, the output feature maps exhibit a minor difference between snow and non-snow areas, leading to the impracticality to calculate the SCR. The model of using the Peak Act function in the last layer (see Column 4) and the Leaky ReLU function in the other layers shows better performance than the models using only the general activation functions. But the best-visualized feature map generated by this model contains fewer details compared to the model that only uses Peak Act (see Column 5). Experiments show that only Peak Act can ensure the model to separate the snow features from non-snow features. We have to raise the question: **Why the non-snow pixels can be depressed when only positive labels are given?**

The key to depress non-snow pixels is to let the network 'express' the snow with no other choices. According to the Peak Act function, the minimal loss can be obtained when the network's outputs are equal to 1. Conversely, when the network's output deviates from 1, a steep gradient ensues, leading to a significant rise in the loss function. The activation region of the existing activation functions, e.g., ReLU, is the whole positive number field, while the activation area of Peak Act is (0, 2). Besides, the output of Peak Act shrinks sharply when it deviates from 1, making the real activation area lie at the narrow belt around 1. This phenomenon is called the narrow activation bandwidth as mentioned before.

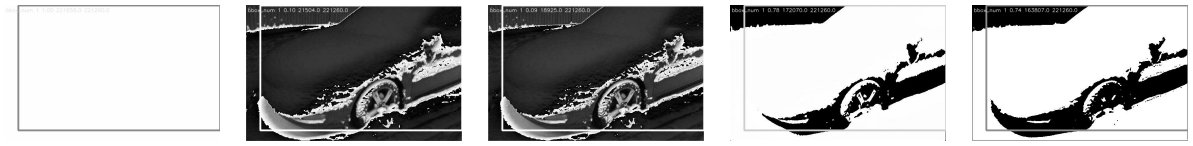
The outputs of the convolutions in the network with Peak Act are limited to around 1, due to the narrow activation bandwidth. It results in a very limited 'expression ability' of the network to map the input data $\rightarrow$1. Given that snow occupies the dominant region of the input image and the corresponding ground truth is an all-ones map, the network must utilize the limited 'expression ability' to map snow $\rightarrow$ 1 for effective loss minimization. Thus, it is very possible that the outputs of non-snow areas would not pass the activation function and be depressed.

In contrast, traditional activation functions, such as Sigmoid, ReLU, and Leaky ReLU, have an excessively large activation bandwidth, resulting in many features accomplishing the mapping process of the input data $\rightarrow$1.

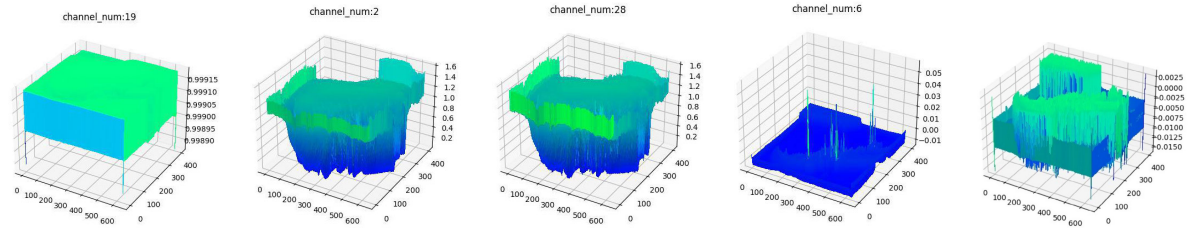
By leveraging Peak Act and the unsupervised training strategy, we can quantitatively evaluate snowy images with only dozens of snow images and a few minutes of training time. Notably, our CNN model accurately figures out the snow areas in images that are not employed during the training process.

As shown in Fig. 8, the images in the second row are the visualized feature maps used to compute the SCR of the images in the first row. For example, in Fig. 8 (a), $SCR_1 = 0.75$; in Fig. 8 (b), $SCR_1 = 0.51$, $SCR_2 = 0.58$; in Fig. 8 (c), $SCR_1 = 0.74$; and in Fig. 8 (d), $SCR_1 = 0.35$, $SCR_2 = 0.56$, $SCR_3 = 0.09$, $SCR_4 = 0.07$. The SCR values in Fig. 8 (a) and (c) are much higher than the other images, which coincides with the human observation. It proves the efficacy of our unsupervised training strategy in quantitatively evaluating the effect of snow.

We combine the SCR and human observation for image difficulty gradations. For instance, humans can easily confirm that Fig. 8 (b) and (c) are affected by snow severely, but they cannot identify whether they belong to the difficult level or the particularly difficult level. In Fig. 8 (b), $SCR_1 = 0.51$, $SCR_2 = 0.58$, indicating that snow covers half of the object, thus, Fig. 8 (b) should be graded into the difficult level. In Fig. 8 (c), $SCR_1 = 0.74$, indicating that snow covers

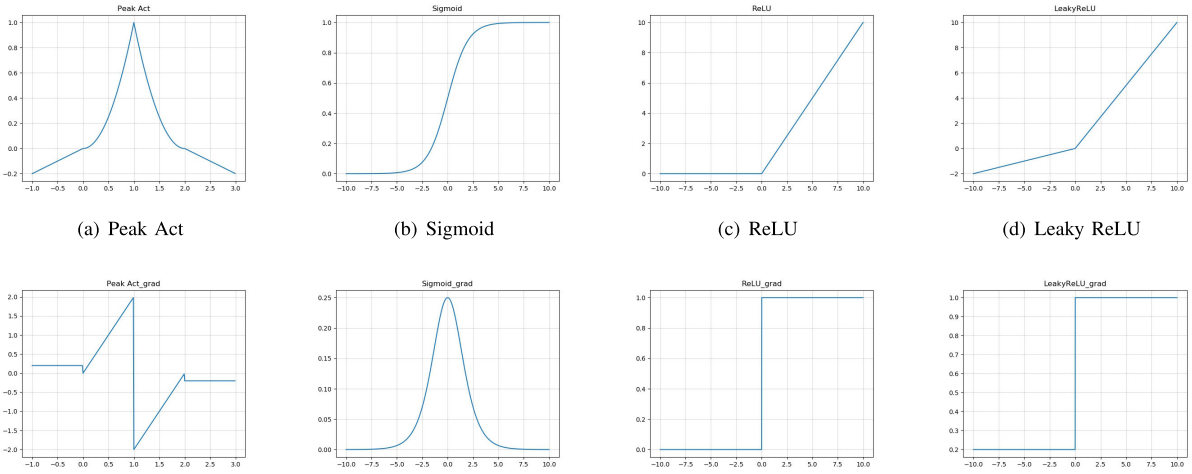


(a) Best feature map using Sigmoid (b) Best feature map using ReLU (c) Best feature map using Leaky ReLU (d) Best feature map using Leaky ReLU and Peak Act in the last layer (e) Best feature map using Peak Act



(f) 3D surface of best channel using Sigmoid (g) 3D surface of best channel using ReLU (h) 3D surface of best channel using Leaky ReLU (i) 3D surface of best channel using Leaky ReLU and Peak Act in the last layer (j) 3D surface of best channel using Peak Act

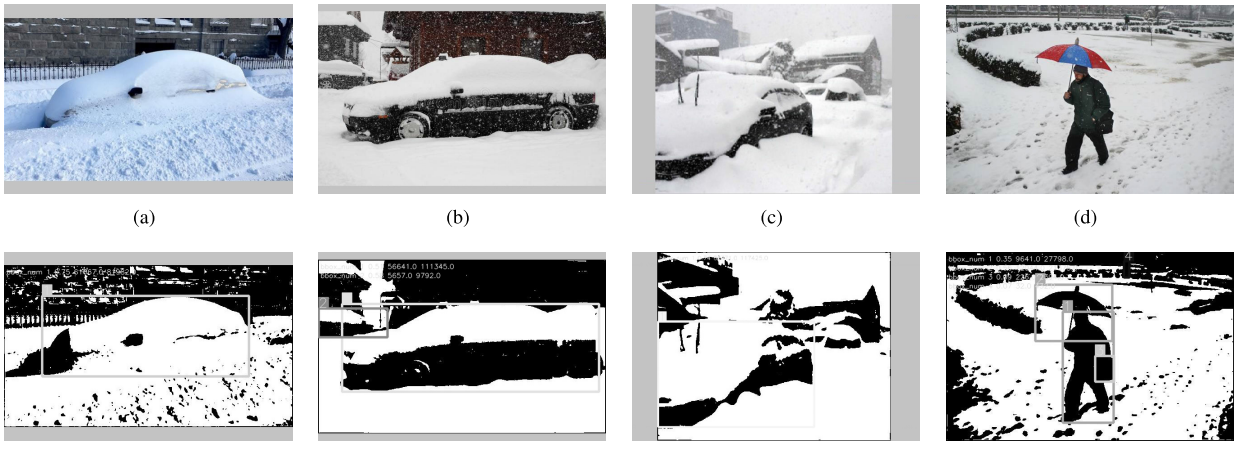
Fig. 6. The visualized output channels of the CNN models with different activation functions. The model with Peak Act (e) generates the clearest feature map, which is highly appropriate for calculating the SCR.



(a) Peak Act (b) Sigmoid (c) ReLU (d) Leaky ReLU

(e) Peak Act gradient (f) Sigmoid gradient (g) ReLU gradient (h) Leaky ReLU gradient

Fig. 7. Different activation functions and their gradients.



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 8. The visualized feature maps used to compute the SCR. The SCR results are shown in the upper left corner of the images in the second row, where the format is "bbox_num X\SCR\A1\A2". bbox_num X indicates the object id in the image, A1 is the area of snow, A2 is the area of the bounding box, and SCR = A1/A2. For instance, the result of feature map (f) is "bbox_num 1 0.51 56641.0 111345.0; bbox_num 2 0.58 5657.0 9792.0".

most of the object, thus, Fig. 8 (c) should be graded into the particularly difficult level. The SCR provides a quantitative metric to avoid bias from different observers and allows fair comparisons when humans are confused.

TABLE I

COMPARISON WITH EXISTING METHODS ON RSOD. RSOD WITH THE FOUR-DIFFICULTY LEVELS PROVIDES A QUANTITATIVE WAY TO STUDY THE DEGRADATION OF DETECTORS IN REAL-WORLD SNOW

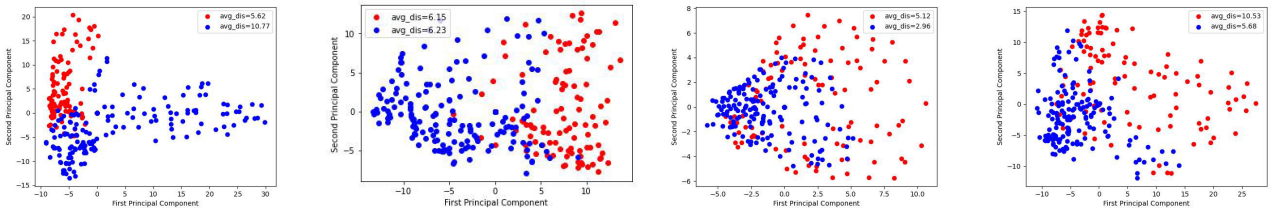
Method	Easy (600)		Normal (1000)		Difficult (400)		Particularly Difficult (100)		All Levels (2100)	
	AP	AP_{50}	AP	AP_{50}	AP	AP_{50}	AP	AP_{50}	AP	AP_{50}
YOLOv5s	41.1	61.3	**36.1**	55.5	26.4	41.3	25.2	39.2	34.3	52.0
YOLOv6-nano (2022)	39.9	57.5	35.7	51.7	**32.0**	**44.9**	29.1	45.3	33.6	49.1
YOLOv7-tiny (2022)	41.5	61.9	**38.7**	**57.7**	27.8	43.9	27.3	**52.4**	**35.9**	**53.9**
EfficientDet D0 (CVPR'20)	20.1	28.6	22.9	34.4	17.1	27.1	29.1	41.0	18.5	27.2
EfficientDet D1 (CVPR'20)	22.4	30.1	26.6	37.6	18.7	29.5	26.6	45.5	21.1	29.9
SSD300 (ECCV'16)	26.1	44.2	27.1	46.8	16.8	29.8	20.5	37.4	23.1	40.2
CF-YOLO (K=1)	**45.6**	**67.5**	34.9	55.0	**27.7**	**44.4**	**30.7**	**47.1**	**34.5**	**53.4**
CF-YOLO (K=3)	**58.0**	**63.8**	35.5	**56.0**	26.3	41.5	**31.9**	44.2	32.7	50.2

(1) Predictions of CF-YOLO

(2) Predictions of YOLOv5s

Fig. 9. The predicted results of CF-YOLO and YOLOv5s trained on MSCOCO only. Compared with YOLOv5s, our method 1) has higher confidence in detection results (columns 1 and 2); 2) can reduce missed detections (columns 3 and 4); and 3) can reduce false detections (columns 5 and 6).

(a) Original prediction of CF-YOLO (b) Original prediction of YOLOv5s (c) Original prediction of YOLOv6n (d) Original prediction of YOLOv7t



(e) PCA for CF-YOLO (f) PCA for YOLOv5s (g) PCA for YOLOv6n (h) PCA for YOLOv7t

Fig. 10. The original predictions and PCA results of different detectors when the detection confidence thresholds are set to 0.01. Only CF-YOLO and YOLOv6n identify the car successfully, whereas YOLOv5s and YOLOv7t fail to pass the non-maximum suppression (NMS) operation owing to their inadequate confidence in the car. Compared with the PCA results of other detectors, CF-YOLO exhibits a superior ability to cluster the object pixels and distinguish the object pixels from the background pixels.

We classify the difficulty level of the images in RSOD mainly based on human observations, since the difficulty level of most images is easy to determine. When humans are confused, SCR can help determine the difficulty level. Generally, the SCR of particularly difficult level is larger than 0.7, and the SCR of difficult level ranges within [0.5,0.7].

B. Performance of Detectors on RSOD

Our CF-YOLO is implemented in PyTorch. All training settings are the same as YOLOv5s (e.g., the batch size=32, the SGD optimizer with the momentum of 0.937 and the weight decay of 0.0005, and the learning rate=0.01). We choose the lightweight versions of the latest cutting-edge detectors for fair comparison, which include YOLOv5s [19], YOLOv6-nano [20], YOLOv7-tiny [21], SSD300 [22], EfficientDet D0 and D1 [24]. These methods and our CF-YOLO, as lightweight networks, can be easily deployed to mobile devices, which require high efficiency under limited computation resources.

RSOD is divided into the training, validation, and test sets with 1701, 189, and 210 images respectively. To balance the difficulty of each subset, the images are randomly allocated to the subsets. In order to verify the effectiveness of CF-YOLO on the four-difficulty levels of RSOD, we test on the four levels and the whole dataset respectively. For a fair comparison, all detectors are trained on MSCOCO only. As shown in

TABLE II

EXPERIMENTAL RESULTS OF DIFFERENT METHODS BEFORE AND AFTER TRAINING ON RSOD. TRAINING ON RSOD BRINGS SIGNIFICANT IMPROVEMENTS TO THE MODELS' PERFORMANCE. SOTAS ARE ALSO MORE ROBUST IN SNOWY SCENES BY TRAINING ON RSOD

Method	AP (val)	AP_{50} (val)	AP (test)	AP_{50} (test)
	Before training / Training on RSOD (20 epochs)			
YOLOv5s	34.2 / 41.1	49.8 / 62.5	32.7 / 37.9	55.4 / 64.6
YOLOv6-nano	32.6 / 43.5	45.9 / 60.2	**36.4** / **45.6**	55.3 / 66.3
YOLOv7-tiny	**38.0** / **44.3**	**53.5** / **66.0**	**36.8** / 40.9	56.3 / 67.5
EfficientDet D0	23.0 / 28.9	33.3 / 47.4	26.3 / 33.3	40.0 / 53.1
EfficientDet D1	25.1 / 37.8	34.5 / 56.8	27.9 / 40.2	41.6 / 62.4
SSD300	23.9 / 34.7	42.4 / 59.8	25.6 / 33.1	46.8 / 59.5
CF-YOLO (K=1)	35.5 / 41.2	51.8 / 64.6	35.6 / **42.4**	**57.6** / **70.6**
CF-YOLO (K=3)	**38.9** / **47.5**	**56.5** / **71.1**	34.0 / 41.9	**58.4** / **70.6**

TABLE III

COMPARISON OF CF-YOLO WITH SOTAS ON MSCOCO. WE CONDUCT THE SPEED TEST ON A SINGLE TESLA V100 GPU WITH BATCH SIZE=1, TAKING THE AVERAGE SPEED OF 5000 IMAGES OF COCO VAL2017

Method	Params	GFLOPS	AP (val)	AP_{50} (val)	AP (test)	AP_{50} (test)	FPS
YOLOv5s	7.3M	17.3	36.3	55.3	36.6	55.3	65
YOLOv6-nano	4.3M	11.2	35.9	51.2	-	-	75
YOLOv7-tiny	6.2M	13.8	38.7	56.7	38.7	-	45
EfficientDet D0	3.9M	2.5	34.3	-	33.8	55.2	63
EfficientDet D1	6.6M	6.1	40.2	-	39.6	58.6	50
CF-YOLO (K=1)	9.2M	17.4	35.5	55.6	35.8	55.7	49
CF-YOLO (K=3)	22M	17.4	36.1	55.8	36.2	55.9	44

Table I, CF-YOLO shows steady and obvious advantages over SOTAs in each difficulty level and the whole dataset. Besides, as shown in Fig. 9, compared with YOLOv5s, CF-YOLO has higher confidence in detection results and can reduce missed and false detections. The reason is that the CF block enables the direct interaction of the features from different stages of the backbone so that the meaningful information diluted in the high-level features can be recovered. It can be observed from Table I that YOLOV7-tiny achieves a superior AP_{50} score on the particularly difficult level compared to our method, but its AP value is inferior. This difference indicates that the bounding boxes generated by YOLOv7-tiny generally have a larger offset (i.e., a low IoU value and a low AP value) while CF-YOLO can locate objects more accurately (i.e., a high IoU value and a high AP value). Furthermore, as shown in Table I, the detectors suffer severe degradation on different difficulty levels. This proves that our proposed gradation strategy can accurately grade snowy images into different levels.

Subsequently, to compare the performance of different methods after training on RSOD, we train the detectors on RSOD with MSCOCO pre-trained weights. We train the networks with only 20 epochs, which is enough since RSOD is much smaller than MSCOCO. As shown in Table II, CF-YOLO still outperforms SOTAs a lot on the validation and test sets, which confirms the advantage of CF-YOLO in snowy weather.

To further investigate how CF-YOLO works, we conduct a PCA study for the outputs of our CF block and the aggregation module of other YOLO detectors (see Fig. 10 (e)∼(h)): **The red points indicate the object pixels and the blue points represent the background pixels**. The upper corner of the figure shows the average distance of all points to their corresponding cluster centers. As shown in Fig. 10, CF-YOLO and YOLOv6n, which successfully detect the car, have smaller average distances than YOLOv5s and YOLOv7t. However, the background pixels of YOLOv6n's PCA result are clustered too close and mixed up with the object (i.e., foreground) pixels (see Fig. 10 (g)), which is one of the reasons why YOLOv6n produces abundant useless and erroneous proposals (see Fig. 10 (c)). In contrast, CF-YOLO can more accurately distinguish the object pixels from the background pixels. The comparison demonstrates that CF-YOLO is more suitable for the snowy scene.

C. Performance of Detectors on MSCOCO

To further investigate the generalization ability of CF-YOLO, we train two versions ($K = 1$ or 3) of CF-YOLO from scratch for 300 epochs on MSCOCO. Table III shows the comparison results between CF-YOLO and SOTAs on MSCOCO. We can see that CF-YOLO with the kernel size of $K = 1$ or $K = 3$ achieves similar results to SOTA detectors, which means our CF-YOLO performs well in snowy weather while still being competitive in normal weather. CF-YOLO (K=3) has much more parameters than YOLOv5s and CF-YOLO (K=1) (see Table III), since gOctConv handles the features from all stages simultaneously. As the kernel size increases from 1 to 3, the number of parameters for a single CF block rises from 2.3M to 8.7M. Although the number of parameters rises disproportionately, CF-YOLO (K=3) and CF-YOLO (K=1) consume similar hardware resources to YOLOv5s because of their similar GFLOPS. The change in the kernel size improves mAP by 0.6% and 0.4% on the validation and test sets, respectively. The FPS of CF-YOLO is lower than the other methods. This is mainly caused by the feature fusion procedure. Nonetheless, the speed of CF-YOLO is sufficient for possible application scenarios.

Although CF-YOLO does not achieve the best AP and FPS scores on MSCOCO, it can be considered as a trade-off between good weather and bad weather. In real-world applications, it is nearly impossible to train two models for good and bad weather, and then shift them according to the

weather. Our CF-YOLO is compatible with both good and bad weather, with only a little sacrifice of accuracy and speed. Considering the severe degradation of snowy images, such a little sacrifice may be tolerable.

VI. Conclusion

Adverse weather often poses the visibility problem for the sensors that power automated systems. While cutting-edge object detectors have obtained promising results on the datasets captured in normal weather, it is still non-trivial to detect objects from the low-quality images captured in adverse weather (e.g., snowy weather). They often ignore the latent information beneficial for detection. By developing an unsupervised training strategy, we establish a high-quality real-world snow dataset for object detection (RSOD). Considering the degradation of CNN-based detectors on RSOD, we propose cross fusion YOLO (CF-YOLO): a lightweight yet effective objection detector. The results show that our CF-YOLO not only achieves excellent performance on RSOD, but also is a competitive and lightweight general detector, which will facilitate the outdoor vision systems.

References

[1] M. Hnewa and H. Radha, "Multiscale domain adaptive YOLO for cross-domain object detection," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2021, pp. 3323–3327.

[2] S. Huang, T. Le, and D. Jaw, "DSNet: Joint semantic learning for object detection in inclement weather conditions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 8, pp. 2623–2633, Aug. 2021.

[3] T. Wang, X. He, Y. Cai, and G. Xiao, "Learning a layout transfer network for context aware object detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 10, pp. 4209–4224, Oct. 2020.

[4] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 1440–1448.

[5] P. W. Patil and S. Murala, "MSFgNet: A novel compact end-to-end deep network for moving object detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 11, pp. 4066–4077, Nov. 2019.

[6] Y. Qian, J. M. Dolan, and M. Yang, "DLT-Net: Joint detection of drivable areas, lane lines, and traffic objects," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 11, pp. 4670–4679, Nov. 2020.

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

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

[9] J. Wei, J. He, Y. Zhou, K. Chen, Z. Tang, and Z. Xiong, "Enhanced object detection with deep convolutional neural networks for advanced driving assistance," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 4, pp. 1572–1583, Apr. 2020.

[10] T. Ye, Zhang, Xi, Y. Zhang, and J. Liu, "Railway traffic object detection using differential feature fusion convolution neural network," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 3, pp. 1375–1387, Mar. 2021.

[11] K. Zhang, R. Li, Y. Yu, W. Luo, and C. Li, "Deep dense multi-scale network for snow removal using semantic and depth priors," *IEEE Trans. Image Process.*, vol. 30, pp. 7419–7431, 2021.

[12] T. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 936–944.

[13] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.

[14] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[15] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 379–387.

[16] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, and D. Lin, "Libra R-CNN: Towards balanced learning for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 821–830.

[17] 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. (CVPR)*, Jun. 2016, pp. 779–788.

[18] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.

[19] (2021). *YOLOv5*. [Online]. Available: https://github.com/ultralytics/yolov5

[20] C. Li et al., "YOLOv6: A single-stage object detection framework for industrial applications," 2022, *arXiv:2209.02976*.

[21] C.-Y. Wang, A. Bochkovskiy, and H.-Y. Mark Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," 2022, *arXiv:2207.02696*.

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

[23] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2999–3007.

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

[25] A. Kosuge, S. Suehiro, M. Hamada, and T. Kuroda, "mmWave-YOLO: A mmWave imaging radar-based real-time multiclass object recognition system for ADAS applications," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–10, 2022.

[26] E. Dong, Y. Zhu, Y. Ji, and S. Du, "An improved convolution neural network for object detection using YOLOv2," in *Proc. IEEE Int. Conf. Mechatronics Autom. (ICMA)*, Aug. 2018, pp. 1184–1188.

[27] L. Qin et al., "ID-YOLO: Real-time salient object detection based on the driver's fixation region," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 15898–15908, Sep. 2022.

[28] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 213–229.

[29] J. Gu et al., "Homography loss for monocular 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 1070–1079.

[30] Y. Chen, Y. Li, X. Zhang, J. Sun, and J. Jia, "Focal sparse convolutional networks for 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 5418–5427.

[31] S. Hu, C. Liu, J. Dutta, M. Chang, S. Lyu, and N. Ramakrishnan, "PseudoProp: Robust pseudo-label generation for semi-supervised object detection in autonomous driving systems," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2022, pp. 4389–4397.

[32] V. A. Sindagi, P. Oza, R. Yasarla, and V. M. Patel, "Prior-based domain adaptive object detection for hazy and rainy conditions," in *Proc. 6th Eur. Conf. Comput. Vis. (ECCV)*, in Lecture Notes in Computer Science, vol. 12359, A. Vedaldi, H. Bischof, T. Brox, and J. Frahm, Eds. Glasgow, U.K.: Springer, Aug. 2020, pp. 763–780.

[33] D. Jaw, S. Huang, and S. Kuo, "DesnowGAN: An efficient single image snow removal framework using cross-resolution lateral connection and GANs," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 4, pp. 1342–1350, Apr. 2021.

[34] D. Liang et al., "Semantically contrastive learning for low-light image enhancement," in *Proc. AAAI Conf. Artif. Intell.*, vol. 36, no. 2, 2022, pp. 1555–1563.

[35] Y. Liu, D. Jaw, S. Huang, and J. Hwang, "DesnowNet: Context-aware deep network for snow removal," *IEEE Trans. Image Process.*, vol. 27, no. 6, pp. 3064–3073, Jun. 2018.

[36] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8759–8768.

[37] G. Ghiasi, T. Lin, and Q. V. Le, "NAS-FPN: Learning scalable feature pyramid architecture for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7029–7038.

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

[39] Y. Chen et al., "Drop an octave: Reducing spatial redundancy in convolutional neural networks with octave convolution," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3434–3443.

[40] S. Gao, Y. Tan, M. Cheng, C. Lu, Y. Chen, and S. Yan, "Highly efficient salient object detection with 100K parameters," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 702–721.
[41] C. Wang, H. M. Liao, Y. Wu, P. Chen, J. Hsieh, and I. Yeh, "CSPNet: A new backbone that can enhance learning capability of CNN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 1571–1580.
[42] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. 40th Int. Conf. Artif. Intell. Statist.*, 2011, pp. 315–323.
[43] A. L. Maas et al., "Rectifier nonlinearities improve neural network acoustic models," in *Proc. ICML*, vol. 30, no. 1, 2013, p. 3.

Qiqi Ding is currently pursuing the Ph.D. degree with the School of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics (NUAA), China. Her research interests include deep learning, image processing, and computer vision, particularly in the domains of object detection and point cloud processing.

Peng Li is currently pursuing the Ph.D. degree with the School of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics (NUAA), China. His research interests include deep learning, image processing, and computer vision.

Xuefeng Yan received the Ph.D. degree from the Beijing Institute of Technology in 2005. He was a Visiting Scholar with Georgia State University in 2008 and 2012. He is currently a Professor with the School of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics (NUAA), China. His research interests include intelligent computing, MBSE/complex system modeling, simulation, and evaluation.

Ding Shi is currently pursuing the Ph.D. degree with the School of Computer Science and Technology, Nanjing University Aeronautics and Astronautics (NUAA), China. His research interests include reinforcement learning and computer vision, particularly in the domains of simulation intelligence.

Luming Liang received the Ph.D. degree in computer science with a minor in geophysics from the Colorado School of Mines in 2014. He is currently a Principal Research Manager with the Applied Sciences Group, Microsoft. His research interests include computer vision, image processing, deep learning, and 3D geometry processing.

Weiming Wang received the Ph.D. degree in computer science and engineering from The Chinese University of Hong Kong in 2014. He is currently an Assistant Professor with the School of Science and Technology, Hong Kong Metropolitan University (HKMU). Before joining HKMU, he was an Assistant Researcher with the Shenzhen Institutes of Advanced Technology from 2014 to 2015. After that, he was with high-tech companies for a few years. His research interests include image processing, deep learning, and 3D geometry processing. He is also serving as an Associate Editor for *The Visual Computer*.

Haoran Xie (Senior Member, IEEE) received the Ph.D. degree in computer science from the City University of Hong Kong, Hong Kong, SAR, China, and the Ed.D. degree in digital learning from the University of Bristol, U.K. He is currently an Associate Professor with the Department of Computing and Decision Sciences, Lingnan University, Hong Kong. He has published 350 research publications, including 192 journal articles, such as IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, IEEE TRANSACTIONS ON AFFECTIVE COMPUTING, and IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY. His research interests include artificial intelligence, big data, and educational technology. He is the Editor-in-Chief of *Natural Language Processing Journal*, *Computers & Education: Artificial Intelligence*, and *Computers & Education: X Reality*. He has been selected as the World Top 2% Scientists by Stanford University.

Jonathan Li (Fellow, IEEE) received the Ph.D. degree in geomatics engineering from the University of Cape Town, South Africa, in 2000. He is currently a Professor with the Department of Geography and Environmental Management and cross-appointed with the Department of Systems Design Engineering, University of Waterloo, Canada. He has coauthored over 500 publications, including more than 300 in refereed journals and more than 200 in conference proceedings. His main research interests include image and point cloud analytics, mobile mapping, and AI-powered information extraction from LiDAR point clouds and earth observation images. He is a fellow of the Engineering Institute of Canada. He was a recipient of the 2021 Geomatica Award, the 2020 Samuel Gamble Award, and the 2019 Outstanding Achievement Award in Mobile Mapping Technology. He is also serving as the Editor-in-Chief for *International Journal of Applied Earth Observation and Geoinformation* and an Associate Editor for IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, and *Canadian Journal of Remote Sensing*.

Mingqiang Wei (Senior Member, IEEE) received the Ph.D. degree in computer science and engineering from The Chinese University of Hong Kong (CUHK) in 2014. He is currently a Professor with the School of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics (NUAA). Before joining NUAA, he was an Assistant Professor with the Hefei University of Technology and a Post-Doctoral Fellow with CUHK. His research interests include 3D vision, computer graphics, and deep learning. He was a recipient of the CUHK Young Scholar Thesis Awards in 2014. He is also an Associate Editor of *ACM TOMM*, *The Visual Computer*, and *Journal of Electronic Imaging*, and the Guest Editor of IEEE TRANSACTIONS ON MULTIMEDIA.