Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.081553

ARTICLE

An Adaptive Multi-Scale Dilated Convolution Network for Real-Time Road Black Ice Detection

Sun-Kyoung Kang[1] **and Yeonwoo Lee**[2,*]

[1]Department of Computer Software Engineering, Wonkwang University, Jeonbuk, Republic of Korea
[2]Department of Artificial Intelligence Engineering, Mokpo National University, Chonnam, Republic of Korea

*Corresponding Author: Yeonwoo Lee. Email: ylee@mokpo.ac.kr

Received: 04 March 2026; Accepted: 13 May 2026; Published: 15 June 2026

ABSTRACT: Black ice formation on road surfaces presents a serious hazard due to its low visibility and high slipperiness, underscoring the critical need for timely and accurate detection in intelligent transportation systems. In this paper, we propose AdaMsDCNet, an adaptive multi-scale dilated convolution network designed for real-time black-ice semantic segmentation on resource-constrained edge platforms, applying a Convolutional Neural Network (CNN) with an adaptive Multi-Scale Dilated Convolution (MsDC) feature fusion encoder-decoder architecture. The key concept of AdaMsDCNet is to employ an encoder-decoder architecture with parallel multi-scale dilated convolutional paths that adjust dilation rates at different encoder depths using a systematic 4→2→1 progression, optimally capturing a wide range of receptive fields while mitigating checkerboard artifacts. The encoder dynamically fuses features from multiple dilation rates at each stage, enhancing segmentation accuracy. Simultaneously, the decoder uses transposed convolutions and skip connections to preserve fine spatial details. Experimental validation on a proprietary thermal infrared dataset of 1156 annotated images show that AdaMsDCNet_9 achieves 96.47% mIoU, 95.48% Black-Ice IoU, 97.55% Precision, 97.82% Recall, and 97.69% F1-Score, outperforming U-Net (+26.78 pp mIoU, +29.88 pp Recall), DeepLabv3+ (+2.82 pp mIoU), and LinkNet (+1.08 pp mIoU) while requiring only 1.86M parameters and maintaining real-time inference speeds of 3.94~5.63 FPS on the NVIDIA Jetson Nano embedded GPU. Ablation studies confirm the benefits of adaptive dilation, parallel feature fusion, and controlled channel growth for the accuracy–efficiency trade-off. Limitations including dataset generalization to uncontrolled outdoor conditions and the evaluation of imbalance-aware loss functions are identified as directions for future work.

KEYWORDS: CNN; multi-scale dilation; convolution feature fusion; black ice detection

1 Introduction

Black ice refers to a thin transparent layer of ice on road surfaces that is often invisible to drivers, leading to extremely hazardous driving conditions. It is a major cause of winter traffic accidents. Because black ice adopts the visual characteristics of the underlying pavement, traditional optical sensing modalities, including standard Red-Green-Blue (RGB) cameras and Light Detection and Ranging (LiDAR)-based system, frequently fail to distinguish hazardous patches from dry or merely wet surfaces [1]. In particular outdoor road environments introduce additional sensing challenges, including rapid illumination changes between day and night, specular reflections from vehicle headlights, precipitation-induced thermal noise from rain and snow, and emissivity variations across different road materials such as asphalt and concrete further degrade detection reliability. Thermal Infrared Ray (IR) imaging provides a superior sensing modality for autonomous and assistive driving systems by exploiting emissivity differentials and subtle temperature

gradients inherent in phase-changed surface moisture, enabling detection of invisible hazards through thermal contrast under all lighting conditions [2].

Recent advances in deep learning have enabled pixel-wise semantic segmentation, offering a principled framework for black ice region delineation. Fully Convolutional Networks (FCN) pioneered end-to-end pixel-wise prediction but suffers from imprecise edge segmentation due to information loss during down-sampling [3]. U-Net, although originally proposed for biomedical image segmentation, established the foundational encoder-decoder architecture with skip connections that enables precise multi-scale feature localization. This architectural paradigm has been widely adopted across diverse segmentation tasks; however, its approximately 31 million (M) parameters impose substantial computational overhead, limiting deployment on resource-constrained embedded devices [4]. DeepLabv3+ and PSPNet proposed atrous (dilated) convolution and pyramid pooling (Atrous Spatial Pyramid Pooling, ASPP) to capture multiple-scale context, achieving state-of-the-art accuracy at the cost of 41M~134M parameters [5,6]. For real-time applications, lightweight segmentation networks have been developed to balance accuracy and efficiency. ENet (0.37M parameters) [7] and LinkNet (11.5M parameters) enable real-time performance while preserving segmentation quality [8,9]. However, overly aggressive parameter reduction in these lightweight models may compromise their ability to capture subtle features essential for detecting thin, irregular black ice patches under varying environmental conditions. A further architectural challenge in the checkerboard (gridding) artifact that arises when dilated convolutions are stacked with identical dilation rates across all networks depths; uniform dilation patterns cause the receptive field to cover only a fixed sparse grid of pixels, leaving spatial gaps that impair segmentation precision—particularly for small or thin structures such as black ice patches—despite proposed mitigations such as Hybrid Dilated Convolution (HDC) and ASPP-based multi-scale parallel dilation [10].

A critical unresolved challenge is that the optimal receptive field size for black ice detection varies dynamically with environmental conditions. Under near-freezing temperatures (around −1°C), the thermal contrast between black ice and dry asphalt narrows to less than 1°C, requiring fine-grained local texture analysis [2,11]. Conversely, detecting the spatial extent of large ice patches requires broad contextual receptive fields, as the effective receptive field of a convolutional layer is closely tied to the spatial scale of the target object [5]. Furthermore, outdoor variations in road material emissivity (asphalt vs. concrete), ambient temperature fluctuations, and vehicle-induced thermal interference necessitate an architecture that can adaptively adjust its feature extraction scale rather than relying on a fixed dilation strategy [11,12]. Conventional architectures apply identical dilation rates across all encoder depths, which either wastes receptive field capacity at high-resolution shallow layers or provokes excessive sampling at low-resolution deep layers [9,10], both of which negatively impact the accurate segmentation of black ice boundaries.

To address these challenges, we propose AdaMsDCNet (Adaptive Multi-Scale Dilated Convolution Network), a novel semantic segmentation architecture specifically optimized for real-time black ice detection on edge computing platforms. Note that the network input consists exclusively of single-channel thermal infrared images captured by a TPV-IAHDR infrared camera; grayscale thermal images are replicated to 3 channels solely for compatibility with standard deep learning frameworks, and no RGB imagery is used. The key contributions of this work are summarized as follows.

(1) Multi-scale Dilated Convolutional (MsDC) Encoder-Decoder Architecture: MsDCNet_p integrates parallel multi-scale dilated convolutions within an encoder–decoder framework. This design captures both local details and global context while preserving spatial information through skip connections.
(2) Adaptive Dilation Strategy to Mitigate Checkerboard Artifacts: AdaMsDCNet_p introduces depth-adaptive dilation rates (4→2→1 progression) to prevent checkerboard artifacts and overly sparse

sampling in deeper layers. This improves receptive-field balance for detection of small, irregular ice regions.

(3) Lightweight Architecture for Edge-Ready Design: By controlling channel growth and removing redundant operations, the model is reduced to about 1.86M parameters, far smaller than conventional networks, enabling efficient deployment on resource-constrained systems.

(4) Real-Time Embedded Performance: The model runs at approximately 3.94~5.63 FPS on an NVIDIA Jetson Nano at 576 × 768 resolution, satisfying real-time requirements for vehicle-based, roadside, and UAV black ice detection systems.

To the best of our knowledge, this is the first semantic segmentation framework specifically tailored for black ice detection that simultaneously achieves both high accuracy and real-time performance on edge devices. The remainder of this paper is organized as follows. Section 2 reviews related work and Section 3 describe the proposed architecture. Section 4 presents experimental results, and Section 5 concludes the limitations and future directions.

2 Related Research

2.1 Black Ice Detection Methods and Semantic Segmentation Architectures

Black ice formation on road surfaces poses a serious safety risk due to its low visibility and sudden impact on vehicle stability. Early approaches employed conventional machine learning methods in [13], demonstrating image-based road surface state classification using SVM with segmentation-based visual features. To enhance environmental robustness, sensor-based systems were developed. Ref. [12] utilized concrete-embedded electrical resistance sensors, Ref. [14] employed depth imaging via Kinect, and Ref. [11] applied multi-wavelength optical sensing for spectral reflectance analysis. More recently, deep learning approaches have demonstrated superior performance as in [1], achieving significantly improved accuracy using convolutional neural networks (CNNs), marking a shift from traditional sensor-based methods toward data-driven frameworks. In particular, Kim et al. [15] proposed a vision-based lightweight CNN with Contrast Limited Adaptive Histogram Equalization (CLAHE) preprocessing and depth-wise convolutions for black ice identification under challenging lighting conditions, and Ref. [16] demonstrated that millimeter-wave (mmWave) backscattering combined with a 1D-CNN classifier achieves robust black ice detection independent of ambient lighting.

The widespread availability of camera-based sensing has made image analysis a cost-effective method for road surface monitoring, driving the adoption of deep neural networks for pixel-wise semantic segmentation. Contemporary semantic segmentation research has adopted deep neural network architectures capable of pixel-wise classification for precise delineation of hazardous regions. Notably, Ref. [17] demonstrated that thermal infrared imaging enables reliable unstructured road segmentation under nighttime conditions using a lightweight encoder-decoder architecture, confirming the suitability of thermal imaging for adverse-condition road surface analysis. Table 1 summarizes representative segmentation architectures relevant to black ice detection, comparing their key characteristics, parameters, and limitations.

Table 1: Representative segmentation architectures relevant to black ice.

Model	Authors (Yr)	Key Technology	Parameters	Advantages	Limitations
FCN	Long et al. (2015) [3]	Fully convolutional layers, skip connections	~65M	End-to-end dense prediction	Imprecise edge segmentation due to aggressive down-sampling
U-Net	Ronneberger et al. (2015) [4]	Encoder-decoder with skip connections	~31M	Precise boundary localization through feature concatenation	Originally designed for biomedical segmentation; High parameter count limits deployment on resource-constrained devices
DeepLabv3+	Chen et al. (2018) [5]	Atrous convolution, ASPP, encoder-decoder	~41M	Multi-scale context without resolution loss	Substantial computational resources required
Pyramid Scene Parsing Network (PSPNet)	Zhao et al. (2017) [6]	Pyramid pooling, ResNet backbone	~134M	State-of-the-art accuracy with scale-specific features	Largest model, unsuitable for embedded systems
ENet	Paszke et al. (2016) [7]	Factorized convolutions, bottleneck modules	~0.37M	Extremely lightweight, real-time capable	Limited capacity for subtle feature detection
LinkNet	Chaurasia & Culurciello (2017) [8]	Residual connections, optimized encoder-decoder	~11.5M	Balanced efficiency and accuracy	May struggle with irregular black ice patterns

While large-scale models (FCN, U-Net, DeepLabv3+, PSPNet) achieve high seg-mentation accuracy, their substantial parameter counts (31~134M) prevent real-time operation on embedded GPUs such as the NVIDIA Jetson Nano. Conversely, lightweight models (ENet, LinkNet) enable real-time inference but may lack sufficient representational capacity to detect highly variable black ice patches with subtle thermal signatures and irregular boundaries. This work addresses these limitations by proposing an adaptive multi-scale dilated convolution network that achieves an order of magnitude reduction in model size (approximately 1.86M parameters) without compromising segmentation accuracy for black ice detection.

2.2 Lightweight Segmentation Network

For real-time applications on embedded systems, lightweight segmentation architectures have been developed to balance accuracy with computational efficiency. Ref. [8] proposed ENet, an extremely compact architecture with approximately 0.37M parameters that employs factorized convolutions and bottleneck modules to minimize computational complexity, though aggressive parameter reduction may compromise feature representation capacity for challenging detection tasks. Ref. [9] introduced LinkNet, utilizing residual connections and an optimized encoder-decoder structure with approximately 11.5M parameters to achieve real-time inference while preserving segmentation quality through efficient skip connections.

Recent advancements continuously push the accuracy-efficiency frontier in semantic segmentation. For instance, MobileNetV3 combines neural architecture search with hard-swish activations to create lightweight backbones for edge deployment [18]. BiSeNetV2 introduces a bilateral network that decouples spatial details from semantic context, achieving real-time speed without sacrificing accuracy [19]. Furthermore, Real-Time Transformer (RTFormer) [20] and SegFormer [21] demonstrate that efficient attention mechanisms and

lightweight Multi-Layer Perceptron (MLP) decoders can make transformer-based architectures highly compact and effective across multiple scales. Proportional-Integral-Derivative Network (PIDNet) also advances real-time segmentation by deploying a three-branch architecture designed specifically to resolve semantic-detail conflicts at object boundaries [22]. Regarding dilated convolution design, Ref. [23] demonstrated that applying adaptive large dilation rates throughout the backbone enables competitive receptive field coverage without aggressive spatial down-sampling, directly motivating the depth-varying dilation strategy employed in our AdaMsDCNet. Building on this principle, HDGNet [24] combined hybrid dilated convolutions with channel attention to achieve real-time road scene segmentation with significantly reduced model parameters, and Lightweight Multiple-Information Interaction Network (LMIINet) [25] integrated depth-wise separable, asymmetric, and dilated convolutions within a lightweight feature interaction module to achieve a favorable balance between accuracy and inference speed. Regarding spatial-spectral multi-scale feature representation, Ref. [26] recently proposed an ultra-lightweight spatial-spectral feature cooperation network for remote sensing change detection. This work proved that cooperative exploitation of spatial and spectral features within a lightweight framework achieves competitive accuracy with minimal computational overhead. Inspired by this principle, our AdaMsDCNet employs a multi-scale parallel dilation strategy. By leveraging multiple parallel dilated convolutional branches, our network effectively captures spatial features across diverse scales, successfully adapting this highly efficient multi-scale approach to the thermal infrared road surface domain.

Despite these advances, existing lightweight segmentation approaches face critical limitations for real-time black ice detection on edge platforms. Large-scale models (U-Net, PSPNet, DeepLabv3+) achieve high accuracy but cannot operate in re-al-time on embedded GPUs such as the NVIDIA Jetson Nano due to substantial parameter counts (31~134M). Lightweight models (ENet, LinkNet) enable real-time inference but may lack sufficient representational capacity to detect highly variable black ice patches with subtle thermal signatures and irregular boundaries in thermal images. Consequently, there is a need for lightweight yet expressive segmentation networks specifically optimized for real-time black ice detection on resource-constrained edge devices.

3 Proposed MsDCNet_Px Architecture

3.1 Platform Architecture of Real-Time Black Ice Detection

Conventional road monitoring systems transmit all captured images to cloud data centers for processing. This approach increases latency, bandwidth usage, and storage costs—especially since most images do not contain black ice [27]. Such cloud-only processing cannot meet real-time traffic safety requirements. Distributed edge-cloud collaborative frameworks have demonstrated significant latency reduction and bandwidth efficiency in real-time IoT-based surveillance applications [28]. To address this limitation, we propose a cloud–edge collaborative warning system that combines edge computing with deep learning. Instead of sending all data to the cloud, a lightweight segmentation model is deployed on edge devices installed near road cameras. These edge modules process thermal images locally, significantly reducing transmission load, system latency, and power consumption while enabling real-time black ice alerts. This paper proposes a cloud-edge collaborative warning system for real-time road black ice, combining edge computing technology and deep learning, as illustrated in Fig. 1.

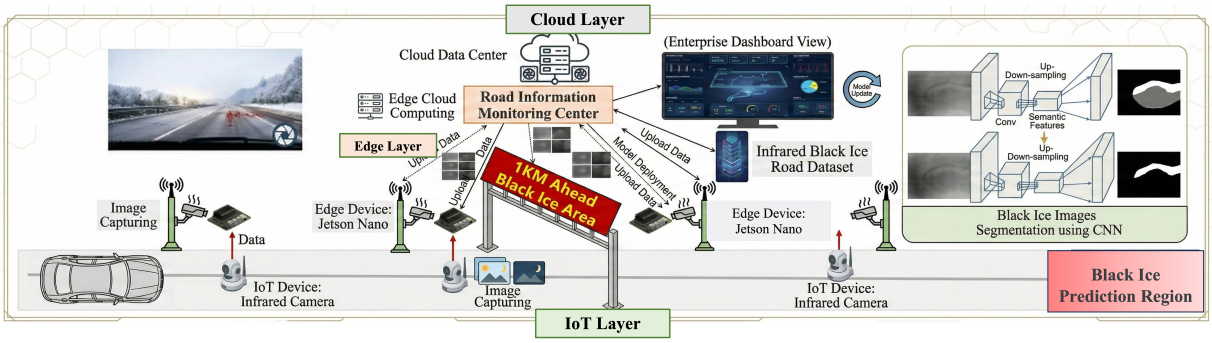
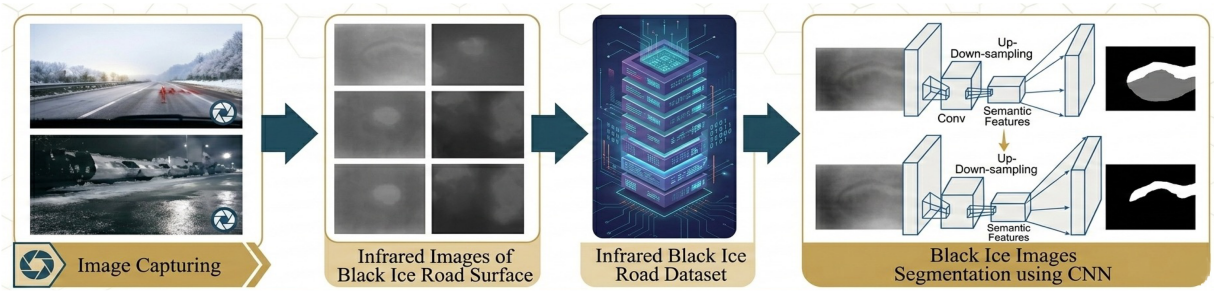


Figure 1: Illustration of a cloud-edge collaborative warning system-based road information monitoring center adopting IoT device, edge device, and edge cloud computing.

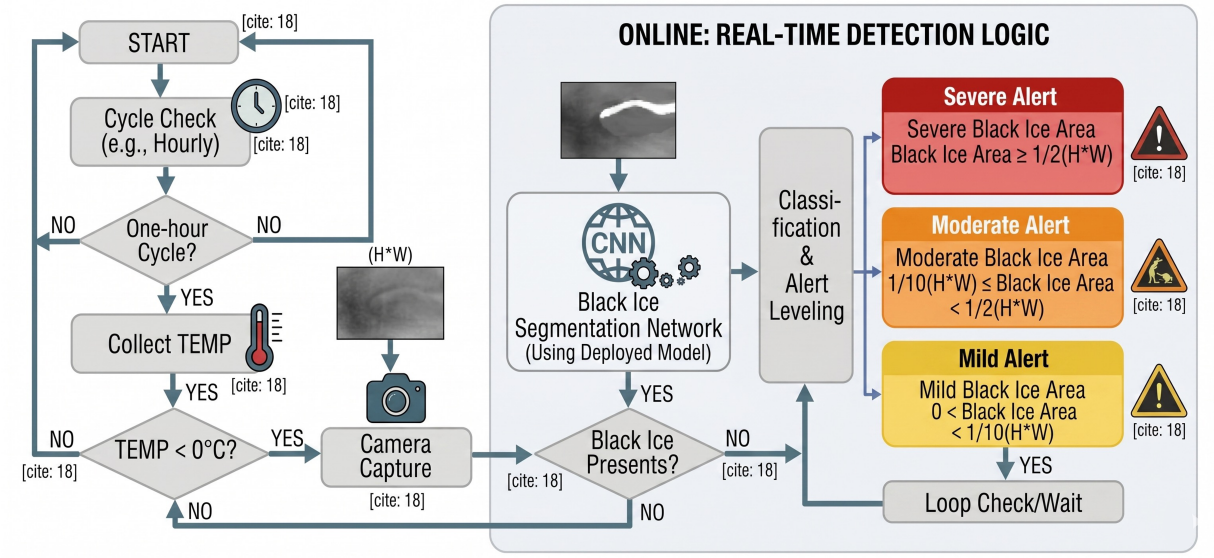
As shown in Fig. 1, the system consists of three layers as follows. (1) Internet of Things (IoT) (thermal camera) layer: a TPV-IAHDR thermal infrared camera captures road surface images. (2) Edge computing layer: an NVIDIA Jetson Nano runs the trained black ice segmentation model for real-time inference. When black ice is detected, the system classifies severity based on segmented area size, displays warnings on roadside Light Emitting Diode (LED) boards, sends alerts to the monitoring center, and periodically uploads key images to the cloud. (3) Cloud data center layer: A high-performance server trains the deep learning segmentation model using a thermal black ice dataset. The trained model is then deployed to edge devices. The cloud also receives alerts, supports management decisions, and stores uploaded road images for further analysis.

Furthermore, thermal infrared sensing is inherently robust than conventional optical sensing methods. Since detection relies on emissivity-based temperature gradients rather than reflected light, the system's performance remains unaffected by illumination changes such as vehicle headlights, shadows, or nighttime darkness. This advantage has been confirmed in roadside thermal infrared monitoring systems [29], where thermal cameras provided consistent detection performance across all lighting conditions that typically degrade RGB camera-based systems. This characteristic ensures reliable outdoor deployment in scenarios where RGB camera-based systems typically experience severe image quality degradation.

Collaborative inference frameworks between edge devices and cloud servers have demonstrated significant latency reduction and resource efficiency in edge intelligence [30], motivating the two-stage offline–online architecture adopted in the proposed system. The system operates in two stages, i.e., offline training and online real-time detection. At offline module, a black ice dataset is constructed using thermal images collected under simulated road icing conditions. A deep convolutional segmentation model is trained in the cloud and deployed to the edge device. At the other stage, i.e., online real-time module, the system is triggered when road temperature falls below 0°C. The thermal camera captures images, and the edge device performs segmentation. This operation is shown in Fig. 2a,b.



(**a**)



(**b**)

Figure 2: (**a**) Illustration of black ice semantic segmentation training module (offline module), (**b**) workflow of real-time black ice region warning module (online module).

3.2 MsDCNet_Px Architecture with Parallel Multi-Scale Dilation Convolution Feature Fusion

Based on constructed infrared black ice road dataset, our CNN-based model is trained on a comprehensive collection of thermal road black ice images for both training and evaluation. To improve black ice detection accuracy, a multi-scale dilation convolution (MsDC) feature fusion technique is proposed. By adjusting dilation ratios according to the input image resolution, this MsDC feature fusion technique enhances the network's feature extraction.

3.2.1 Overall Architecture of MsDCNet_p

We propose an encoder-decoder segmentation network tailored for black ice detection in thermal images, named MsDCNet_p (Multi-scale Dilated Convolution Network with Parallel dilated convolution modules). It is important to note that the network operates exclusively on thermal infrared images; the 3-channel input arises solely from replicating the single-channel grayscale thermal image to maintain compatibility with standard deep learning frameworks. Since no RGB imagery is processed, the complex issue of IR-RGB synchronization does not apply to this system. The overall architecture is illustrated conceptually in Fig. 3. It consists of two main components: an encoder that extracts multi-scale features

from the input image, and a decoder that reconstructs a segmentation mask via pixel-wise classification from those features.

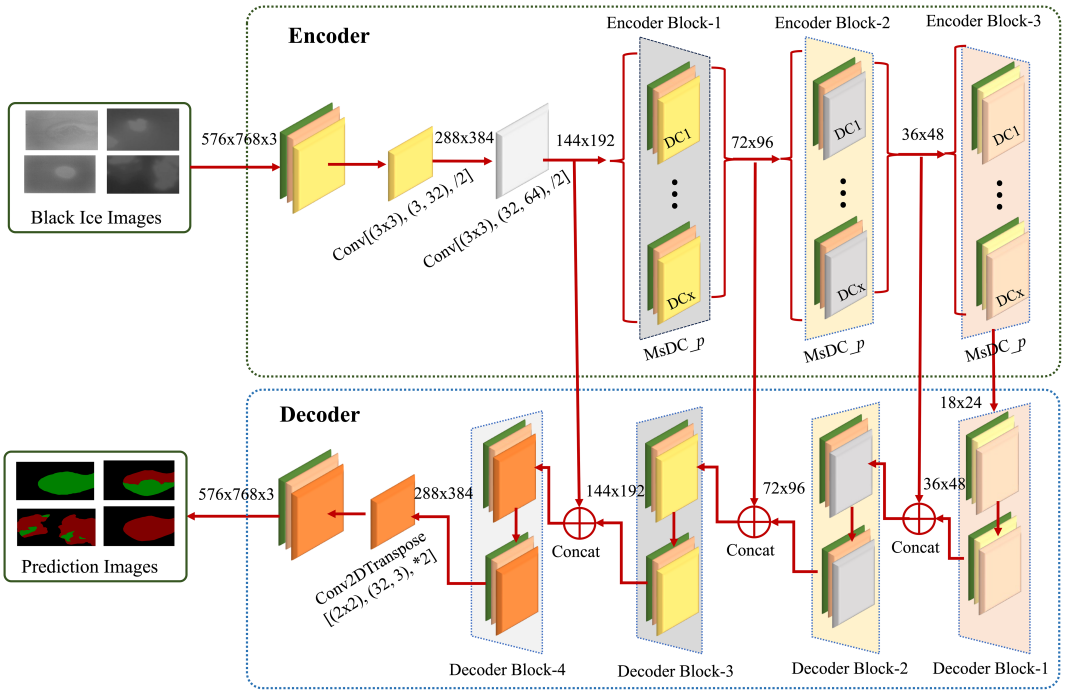


Figure 3: Diagram of the proposed MsDCNet_p network architecture with three encoder blocks with parallel multi-scale dilated convolutional (MsDC_p) feature fusion modules, which containing ResNet-type dilated convolutional (DCx) modules.

The three encoder-stage resolutions (144 × 192, 72 × 96, and 36 × 48) were not arbitrarily chosen, rather, they are naturally derived from the network's hierarchical down-sampling design concept. Starting from the 576 × 768 input image, two initial stride-2 convolutional layers reduce the spatial resolution to 1/4, yielding a 144 × 192 feature map. This initial compression step reduces the computational cost of subsequent operations without discarding essential spatial information. The three subsequent encoder blocks then progressively halve the resolution using stride-2 down-sampling (144 × 192→72 × 96→36 × 48). This hierarchical structure allows each encoder stage to specialize in features at a specific scale. Encoder Block-1 (144 × 192) captures fine-grained local texture details, which is important for detecting thin black ice boundaries, Encoder Block-2 (72 × 96) integrates intermediate contextual information, and Encoder Block-3 (36 × 48) encodes broad semantic context including the overall thermal distribution across the road surface.

Within the MsDC_p module, parallel processing consistency is structurally guaranteed because all parallel paths (DCx modules) share the exact same input tensor. Each path independently applies a dilated convolution with a distinct dilation rate to capture features at various spatial scales simultaneously. These output feature maps, which share the same spatial dimensions, are then concatenated along the channel dimension. A final 1 × 1 convolution normalizes the channel count to produce a single fully synchronized output tensor (as in Eqs. (5) and (6)). Therefore, data discrepancy between paths fundamentally cannot occur. Finally, to prevent the loss of high-frequency spatial details during the encoding process, skip connections link the corresponding encoder and decoder stages. The decoder then utilizes stride-2 transposed convolutions to progressively restore the segmentation map back to the full 576 × 768 resolution, as described in Section 3.2.3.

The decoder uses transposed convolutions (learnable up-sampling, also known as deconvolution) to progressively increase the resolution of the feature maps, combining them with features from the encoder via skip connections to refine the segmentation boundaries. All convolution layers in the network use a kernel size of 3 × 3 (except 1 × 1 conv used for channel adjustment in some places), followed by ReLU activation and batch normalization, which helps stabilize training. To reduce computation, the model begins with an initial down-sampling: the input image is first passed through two 3 × 3 conv layers with stride 2, which reduce the spatial resolution to 1/4 (i.e., 144 × 192) while increasing the feature channels.

3.2.2 Encoder Blocks of MsDCNet_p

After this initial compression, the encoder is organized into multiple encoder blocks (stages). In the proposed MsDCNet_p design, we used 3 encoder blocks (encoder block 1~3), each operating at successively lower resolutions. Each encoder block halves the spatial size (via strided convolution) and outputs a set of fused multi-scale features. Specifically, an encoder block takes as input a feature map of size (H, W, C_in) and produces an output of size (H/2, W/2, C_out), where H and W are halved.

Within each encoder block of the MsDCNet_p, we implement the multi-scale dilated convolutional (MsDC_p) module with p parallel dilated convolutional (DCx) modules where x denotes the dilation rate from 1 to p as illustrated in Figs. 4 and 5.

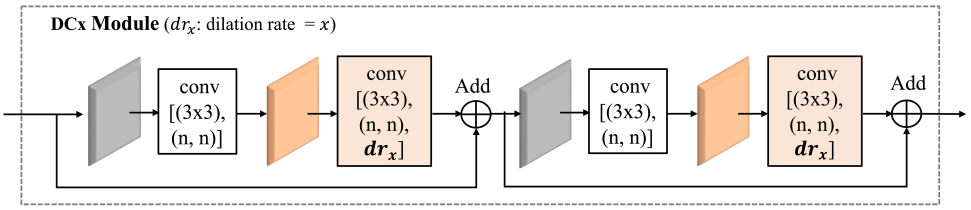


Figure 4: DCx module diagram within a MsDC_p module in an encoder, where x denotes the dilation rate.

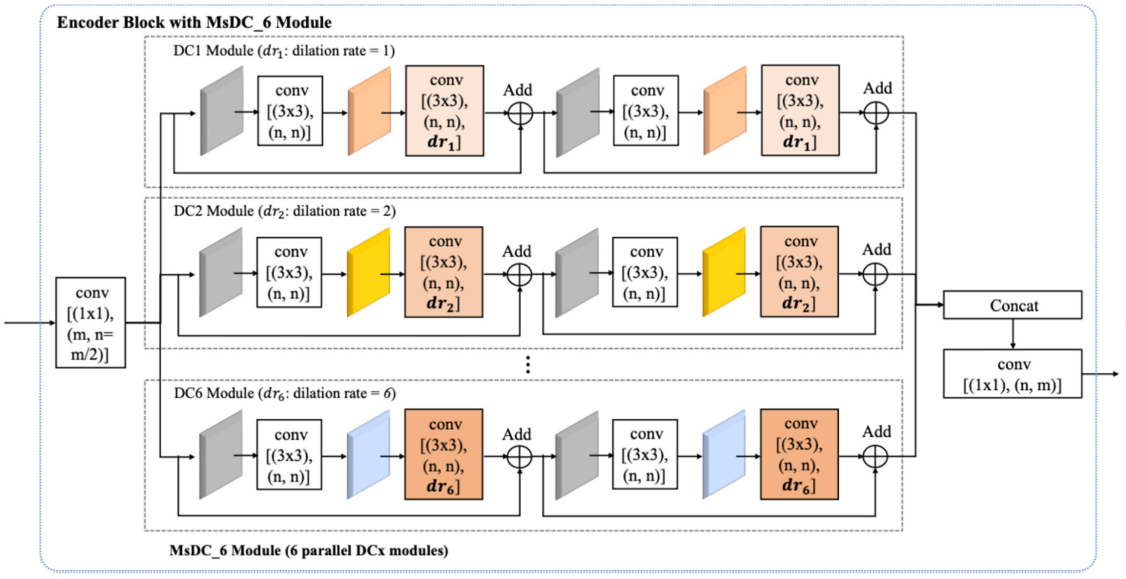


Figure 5: Encoder block diagram of the proposed MsDCNet_p model; this illustrates MsDCNet_6 structure with 6 parallel DCx modules with dilation rates {1, 2, 3, 4, 5, 6} for each MsDC_6 module.

- First, a 1 × 1 convolution may be applied to the input feature map to reduce the channel dimensionality by half. This step is done to cut down the computational cost when we concatenate multiple parallel feature maps; by shrinking the channel count up front, we ensure the concatenated output does not blow up in size even if many parallel paths are used.
- Then, the feature map is fed into P parallel convolutional paths, each path being a small residual sub-network.
- All paths operate on the same down-sampled input (so they see the same features) but with different dilation rates.

In MsDC_p, p denotes the number of DCx modules within each encoder block, which is a parameter with values from 2 to 6. The dilation rate x ranges from 1 up to p. For example, MsDCNet_4 has 3 encoder blocks with each 4 parallel convolutional layers (DC1~DC4 module) in each encoder block, with dilation rates tailored accordingly. Each parallel path (DCx) within an encoder block has a ResNet-style two-layer structure. Concretely, in each encoder block, the first layer is a 3 × 3 convolution with stride-2 (to achieve the down-sampling of that block) and a dilation rate dr_x (if dilation is used here) or effectively a normal convolution if we consider dilation = 1 for the first layer. The second layer is a 3 × 3 dilated convolution with the same dilation rate dr_x. We include a skip connection that adds the appropriately down-sampled input of the block to the output of the second layer (this skip has a 1 × 1 convolution with stride-2 to match dimensions when necessary), forming a residual block.

In a MsDCN_p module, we set the dilation rates in the DCx parallel paths to a fixed set of values that span a range from 1 (no dilation) to p (for P parallel paths). For instance, if $p = 4$, we might use dilation rates $\{1, 2, 3, 4\}$ for the four parallel paths, i.e., DC1, DC2, DC3, and DC4. In our implementation MsDCNet_6 uses dilation rates $\{1, 2, 3, 4, 5, 6\}$ in each encoder block.

3.2.3 DCx Modules and MsDC_p Modules

The MsDC_p module is designed to capture multi-scale features by combining several parallel DCx modules, each with a distinct dilation rate, followed by concatenation and dimensionality restoration using a 1 × 1 convolution.

***DCx Module*:** The DCx module forms the basic building block of our architecture, where x denotes the dilation rate. Each DCx module consists of sequential convolutional operations with residual connections. For an input tensor $X \in \mathbb{R}^{H\times W\times C}$ the DCx module with dilation rate dr_x performs the following operations:

$$\begin{aligned} &h_1 = C_{3\times3}(X),\ h_2 = h_1 + C_{3\times3}^{dr_x}(h_1),\ h_3 = C_{3\times3}(h_2),\ h_4 = h_3 + C_{3\times3}^{dr_x}(h_3) \\ &DCx(X, dr_x) = h_4 \end{aligned} \tag{1}$$

where $C_{3\times3}$ represents a standard 3 × 3 convolution with n output channels, and $C_{3\times3}^{dr_x}$ denotes a 3 × 3 dilated convolution with dilation rate dr_x. The dilated convolution operation used in the DCx module, denoted as $C_{3\times3}^{dr_x}$ can be mathematically expressed as

$$C_{k\times k}^{dr_x}(X)(i,j) = \sum_{m=1}^{k}\sum_{n=1}^{k} w(m,n)\cdot X\left(i + dr_x\cdot\left(m - \left\lceil\frac{k}{2}\right\rceil\right), j + dr_x\cdot\left(n - \left\lceil\frac{k}{2}\right\rceil\right)\right) \tag{2}$$

where dr_x is the dilation rate, k is the kernel size (3 in our case), and w represents the weights of the convolutional kernel.

***MsDC_p Module*:** For the complete MsDC_p architecture, where p represents the number of parallel DCx modules, the mathematical formulation is as follows. The input tensor $X \in \mathbb{R}^{H\times W\times m}$ undergoes an initial

dimensionality reduction using a 1×1 convolution operation,

$$Z = C_{1\times1}(X) \tag{3}$$

where $Z \in \mathbb{R}^{H\times W\times n}$ and $n = m/2$. This step reduces the number of channels to half, ensuring computational efficiency while maintaining spatial dimensions.

The reduced feature map Z is passed through P parallel DCx modules, each with a unique dilation rate dr_x. For the p-th DCx module, the output is defined as

$$Y_p = DCp\left(Z, dr_p\right) \tag{4}$$

where $dr_p = p$ (i.e., the dilation rate increases incrementally for each parallel branch). Each DCx module captures features at a specific scale, and its output is represented as $Y_p \in \mathbb{R}^{H\times W\times n}$. The outputs of all parallel DCx modules are concatenated along the channel dimension to form a unified multi-scale feature map,

$$Y_{cat} = Y_1 \parallel Y_2 \parallel \ldots \parallel Y_P \tag{5}$$

where $\parallel$ denotes channel-wise concatenation. The resulting tensor Y_{cat} has dimensions $\mathbb{R}^{H\times W\times(P\cdot n)}$ effectively combining features from all scales. To restore the original number of channels in the output tensor, a final 1×1 convolution is applied to Y_{cat},

$$Y_{out} = C_{1\times1}(Y_{cat}), \tag{6}$$

where $Y_{out} = \mathbb{R}^{H\times W\times m}$ matches the channel count of the input tensor X. This step ensures that the multi-scale features are integrated into a compact representation suitable for downstream tasks. The entire MsDC_p module can be expressed as

$$\text{MsDC\_p}(X) = C_{1\times1}\left(\|_{x=1}^{P} DCx\left(C_{1\times1}(X), x\right)\right) \tag{7}$$

The use of dilated convolutions in parallel branches allows for an expanded receptive field without increasing kernel size or computational complexity. For P parallel branches, the effective receptive field ranges from 3×3 (for $dr_1 = 1$) to $(2P+1) \times (2P+1)$ (for $dr_x = P$), enabling efficient multi-scale context aggregation.

By combining these parallel convolutional layers, the encoder block can extract features responsive to patterns of different sizes: small dilation focuses on fine details and small objects, whereas larger dilation (e.g., 5 or 6) captures broader context from a larger receptive field. We concatenate the output feature maps from all parallel paths to form the encoder block's output (which then may be projected by a 1×1 convolution to control C_out). Through this concatenation, neurons in the fused feature map effectively encode multi-scale semantic information; some neurons aggregate information from a wide area via the high-dilation path, others focus on local details via the low-dilation path.

This multi-scale feature fusion enriches the representation for complex tasks like black ice segmentation where both global context (e.g., overall road temperature trends) and local texture (e.g., smooth icy patch vs. rough asphalt) are relevant. After three encoder blocks of the MsDCNet_p, the feature map resolution is 1/32 of the input (i.e., 18×24 for 576×768 input). The decoder then performs the reverse process: it consists of a series of decoder blocks that progressively up-sample the feature maps back to the original resolution. Table 2 presents the dilation rate configurations for each MsDCNet_p variant, following the multi-scale dilated convolution framework of with uniform rates applied across all encoder blocks.

Table 2: Used dilation ratio set for each encoder blocks and its DCx modules of the MsDCNet_p model.

Network Model	DCx Modules	DilationRate (dr_x) Set		
		Encoder Block-1	Encoder Block-2	Encoder Block-3
MsDCNet_2	DC1-DC2	{1, 2}	{1, 2}	{1, 2}
MsDCNet_3	DC1-DC2-DC3	{1, 2, 3}	{1, 2, 3}	{1, 2, 3}
...	...	...		
MsDCNet_8	DC1-DC2-...-DC8	{1, 2, ..., 8}	{1, 2, ..., 8}	{1, 2, ..., 8}
MsDCNet_9	DC1-DC2-...-DC9	{1, 2, ..., 9}	{1, 2, ..., 9}	{1, 2, ..., 9}

3.2.4 Decoder Block of MsDCNet_p

Our decoder design primarily focuses on feature refinement and up-sampling. Each decoder block takes a low-resolution feature map and produces a feature map at 2× higher spatial resolution. We use transposed convolutions with stride-2 (also known as decon-volutional layers) for up-sampling, because learnable transposed convolution can potentially recover spatial details more accurately by learning the up-sampling filters. However, naive transposed convolution can introduce checkerboard artifacts in the output if not carefully configured. To avoid this, we use a kernel size of 3 × 3 for transposed conv and follow best practices such as odd kernel overlaps as suggested by Odena et al. [31].

In each decoder block, similar to the encoder, we include some 1 × 1 convolutions to adjust channel dimensions. As depicted in Fig. 6, a decoder block with input channel count m and output channel count n will first apply a 1 × 1 convolution to reduce the channels from m to m/2, then apply a 3 × 3 transposed convolution (stride-2) to double the width and height, and finally a 3 × 3 convolution to increase the channels to n. We also concatenate (or add) the corresponding encoder block's output via skip connection at the appropriate stage before the final convolution, to reintroduce fine-grained features. For example, the decoder block that up-samples from 1/16 to 1/8 resolution will take the 1/16 feature map (up-sampled to 1/8) and fuse it with the encoder's 1/8 feature map (from Encoder Block 1) before outputting the 1/8 feature. This skip fusion helps recover edges of black ice regions that the encoder's deeper layers might have blurred.

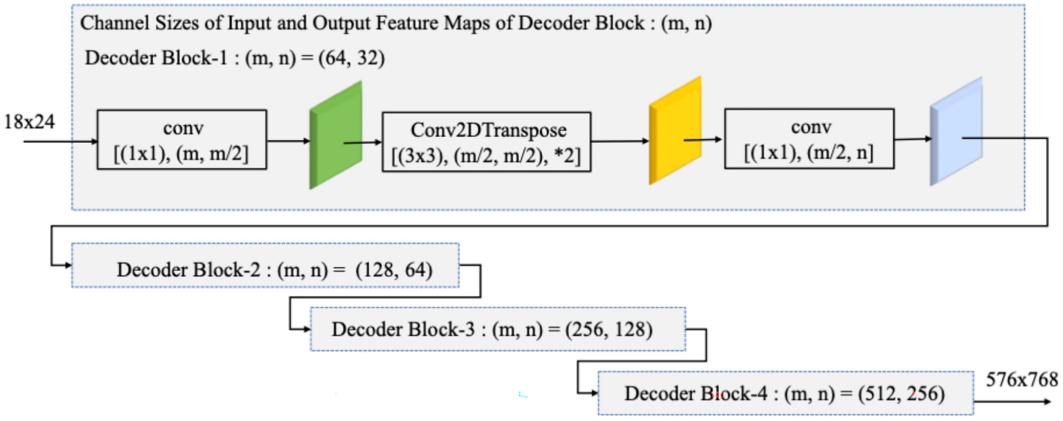


Figure 6: DCxDecoder block diagram of the proposed MsDCNet_p model.

After the final decoder block, we obtain a feature map at 1/2 resolution (since we had one more decoder block than encoder block in MsDCNet_p for P parallel paths as explained below) which is then up-sampled one more time to full 576 × 768 resolution using a transposed conv or interpolation, and a final 1 × 1

convolution produces the two-channel output. We apply a softmax or sigmoid on these logits during training for the segmentation loss.

3.3 Proposed AdaMsDCNet_Px: MsDCNet_px with Adaptive Dilation and Efficiency Improvements

The AdaMsDCNet_p architecture retains the encoder-decoder layout and parallel multi-scale convolution concept of the proposed MsDCNet_p, but introduces targeted refinements to improve segmentation of small ice regions and reduce model complexity for real-time operation.

3.3.1 Multi-Scale Dilation with Depth-Varying Rates

In the proposed AdaMsDCNet_p, we assign larger dilation rates to early encoder blocks (when feature maps are larger) and smaller dilation rates to later blocks (when feature maps are small). This is a main difference with the MsDCNet_p, which allocates the same dilation rates in every encoder block. This method of multi-scale dilation rate with depth-varying rates is as to maintain a roughly consistent effective receptive field in terms of original image pixels and to avoid the extreme sparsity at deeper layers. In practice, we define a maximum dilation for the first block equal to P (the number of parallel paths), and then reduce the maximum dilation in subsequent blocks.

For example, consider AdaMsDCNet_9 (our largest variant with 9 parallel paths). In encoder block-1 (input 144 × 192), we use 9 parallel convolution layers with dilation rates {1, 5, 9, 13, 17, 21, 25, 29, 33}, i.e., starting at 1 and incrementing by 4. This covers a wide range of receptive fields in the high-resolution feature map. In encoder block-2 (input 72 × 96), we also use 9 paths, but with dilation rates {1, 3, 5, 7, 9, 11, 13, 15, 17} (increment by 2). In encoder block-3 (36 × 48 input), we use dilation rates {1, 2, 3, 4, 5, 6, 7, 8, 9} (increment by 1). Thus, the spacing between dilation rates is narrowed at deeper levels (4→2→1). Effectively, the last encoder block-3 focuses more on fine details (since at 1/16 scale even dilation 9 corresponds to a moderate area in the original image), while the first encoder block-1 can focus on broad context (dilation up to 33 covering a large region of the input). This staged reduction of dilation ensures that we do not end up with a situation like MsDCNet_6's encoder block-3 using dilation 6 on 1/32 features (which would skip too many pixels). Instead, AdaMsDCNet_9's largest dilation at the deepest block is 9 on 1/16 features, which still samples reasonably densely.

This strategy is inspired by the concept of hybrid dilation but implemented in a parallel multi-branch manner. By maintaining appropriate receptive fields at each level, we capture multi-scale context without incurring the gridding artifact. In essence, AdaMsDCNet_p's encoder blocks produce overlapping receptive fields across paths and across blocks, so that no portion of a black ice region goes unseen. Empirically, this led to better segmentation of small black ice patches that MsDCNet_p sometimes missed due to overly sparse sampling in deeper layers.

In the proposed AdaMsDCNet_p architecture, the DCx module used in MsDCNet_p (Fig. 4) should be modified as mDCx as following Fig. 7. The exemplary encoder block-1, 2, and 3 of the AdaMsDCNet_p is depicted in Fig. 8 and Table 3.

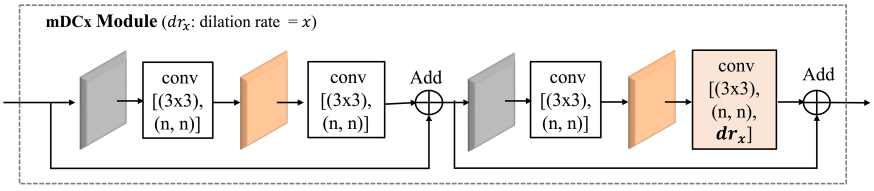


Figure 7: Modified DCx (mDCx) module diagram used in the encoder block of AdaMsDCNet_p.

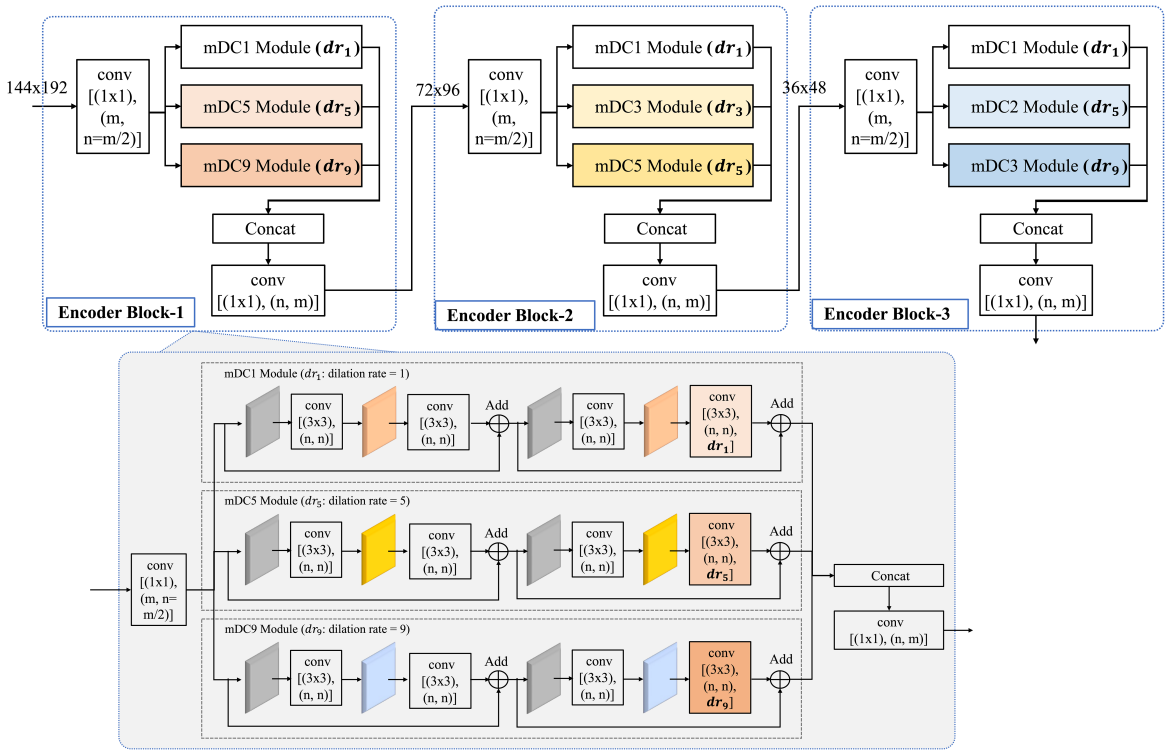


Figure 8: Encoder block diagram of the proposed AdaMsDCNet_p model; this illustrates the structure of AdaMsDCNet_3, consisting of three parallel mDCx modules with each dilation rates of {1, 5, 9}, {1, 3, 5}, and {1, 2, 3} for encoder blocks 1, 2, and 3, respectively.

Table 3: Used dilation ratio set for each encoder blocks and its mDCx modules of the AdaMsDCNet_p model.

Network Model	Dilation Rate (dr_x) Set (mDCx Modules)		
	Encoder Block-1 (Increment by 4)	**Encoder Block-2 (Increment by 2)**	**Encoder Block-3 (Increment by 1)**
AdaMsDCNet_2	{1, 5} mDC1-mDC5	{1, 3} mDC1-mDC3	{1, 2} mDC1-mDC2
AdaMsDCNet_3	{1, 5, 9} mDC1-mDC5-mDC9	{1, 3, 5} mDC1-mDC3-mDC5	{1, 2, 3} mDC1-mDC2-mDC3
...	...	...	
AdaMsDCNet_8	{1, 5, ..., 29} mDC1-mDC5-...-mDC29	{1, 3, ..., 15} mDC1-mDC3-...-mDC8	{1, 2, ..., 8} mDC1-mDC2-...-mDC8
AdaMsDCNet_9	{1, 5, ..., 33} mDC1-mDC5-...-mDC33	{1, 3, ..., 17} mDC1-mDC3-...-mDC8	{1, 2, ..., 9} mDC1-mDC2-...-mDC9

In the AdaMsDCNet_p, dilation rate dr_p of the p-th mDCx module in Eq. (4) varies adaptively according to the encoder block depth and each characterized by a unique dilation rate. This can be rewritten as

$$Y_p = mDCp\left(Z, dr_p\right) \tag{8}$$

where dr_p is not always equal to p but varies adaptively by encoder block depth. The adaptive dilation rates are generalized distinctly for each encoder block as follows.

$$\begin{aligned}
&\text{Encoder Block} - 1 : Y_{cat} = Y_1 \parallel Y_5 \parallel Y_9 \parallel \ldots \parallel Y_{(4p-3)}, \text{with } dr_p \text{ incremented by 4,}\\
&\text{Encoder Block} - 2 : Y_{cat} = Y_1 \parallel Y_5 \parallel Y_9 \parallel \ldots \parallel Y_{(2p-1)}, \text{ with } dr_p \text{ incremented by 2,}\\
&\text{Encoder Block} - 3 : Y_{cat} = Y_1 \parallel Y_5 \parallel Y_9 \parallel \ldots \parallel Y_p, \text{ with } dr_p \text{ incremented by 1.}
\end{aligned} \tag{9}$$

Also, the complete AdaMsDCNet_p module operation can be rewritten as

$$Ada\text{MsDC\_p}(X) = C_{1\times1}\left(\|_{x=1}^{P}\, mDCx\left(C_{1\times1}\left(X\right), dr_p\right)\right) \tag{10}$$

This adaptive multi-scale dilation scheme ensures efficient coverage of diverse receptive fields, capturing both detailed textures and broad contextual information, significantly improving the detection of small black ice regions while maintaining computational efficiency suitable for real-time applications.

3.3.2 Parameter Reduction via Channel Management

To ensure real-time performance, AdaMsDCNet_p further reduces the number of feature channels throughout the network. In many CNNs, it is common to double the number of channels after each down-sampling (to preserve capacity as spatial size shrinks). MsDCNet_p partly followed this (going from 64 channels after encoder block-1 to 96 after encoder block-2 to 128 after encoder block-3 in our example). However, this growth, combined with multiple paths, resulted in a large parameter count for higher *P*. In AdaMsDCNet_p, we opted not to multiply channels by 2 at each stage. Instead, we increase channels more gradually: roughly adding a constant number of channels per block. For instance, in AdaMsDCNet_p we might use output of encoder block-1 has 64 channels, encoder block-2 has 80 channels, encoder block-3 has 96 channels. This linear growth contrasts with doubling, which would have been 64→128→256.

As a result, even though AdaMsDCNet_9 has more parallel convolution paths than MsDCNet_6, its total channel count at deeper layers is much lower. Moreover, within each encoder block, before applying the parallel convolution paths, we still perform the 1 × 1 channel reduction as in MsDCNet_p. Thus, each path operates on a smaller slice of channels. The skip connections between encoder and decoder also use fewer channels accordingly. All these adjustments dramatically cut down the parameter count. Table 4 quantifies this by comparing the parameter sizes of classical segmentation networks and the proposed model variants, confirming that the AdaMsDCNet series achieves an order-of-magnitude reduction relative to conventional architectures. MsDCNet_6 had 20.7M parameters, whereas AdaMsDCNet_9 with even more branches has only less than 1.86M. In other words, through our efficient design, AdaMsDCNet_9 is about 11 times smaller than MsDCNet_6, and even AdaMsDCNet_9 is 6 times smaller than LinkNet and an order of magnitude smaller than DeepLabv3+ or PSPNet. It is worth noting that such an aggressive reduction could have impacted accuracy; however, the multi-scale feature fusion appears to compensate effectively, as AdaMsDCNet_p models achieve comparable segmentation performance to MsDCNet_p models.

Table 4: The size of parameters used in classical image segmentation networks and the proposed MsDCNet_ p and AdaMsDCNet_p.

Network Model	Parameters (KB)	Network Model	Parameters (KB)
U-Net	31,055	DeepLabV3+	41,253
FCN8	65,810	ENet	371
PSPNet101	134,325	LinkNet	11,555
MsDCNet_2	7192	AdaMsDCNet_2	492
MsDCNet_3	10,558	AdaMsDCNet_3	687
MsDCNet_4	13,924	AdaMsDCNet_4	882
MsDCNet_5	17,289	AdaMsDCNet_5	1077
MsDCNet_6	20,655	AdaMsDCNet_6	1273
		AdaMsDCNet_7	1468
		AdaMsDCNet_8	1663
		AdaMsDCNet_9	1858

3.3.3 Decoder Block of the AdaMsDCNet_p

The decoder in AdaMsDCNet_p is the same structure as that of MsDCNet_p, with the same channel reduction and transposed convolution steps in each block as shown in Fig. 6. The channel sizes for each decoder block are adjusted to match the slimmer encoder. By the final decoder stage, we end up with 32 channels at 1/2 resolution, which are then converted to the output classes. We also ensure that skip connections from encoder to decoder are in place for each resolution (1/16, 1/8, 1/4, and even 1/2 if we consider the initial input as skip to final output convolution).

It is noted that the proposed AdaMsDCNet_p preserves the core multi-scale encoder-decoder design of MsDCNet_p but optimizes the dilation usage and network width to be both more effective and more efficient. Based on the desired accuracy trade-off, specific variant model of AdaMsDCNet_p can be chosen. In our experiments, AdaMsDCNet_6 to AdaMsDCNet_9 provided the best accuracy, with AdaMsDCNet_9 slightly better, while still being lightweight enough for real-time inference.

4 Simulation and Results

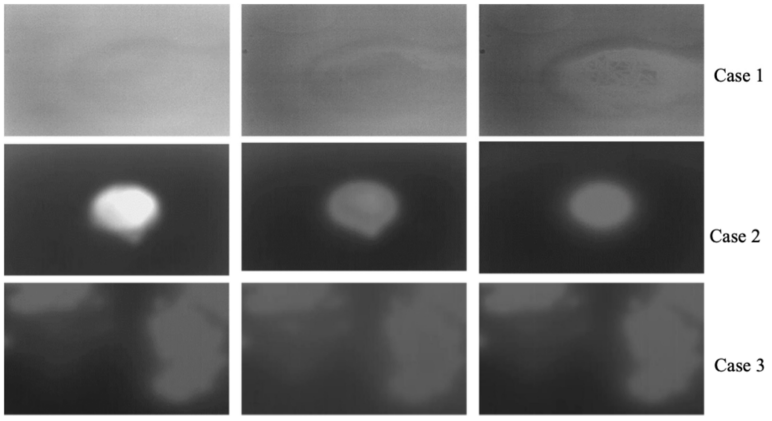
4.1 Experiment Setup and Image Dataset

To collect diverse road black ice images, a total of 10 road surfaces were constructed, comprising both asphalt and concrete pavement samples, each measuring (1 m × 1 m) with a thickness of 5 cm. These samples were installed in a freezing facility and subsequently sprayed with water before being captured using thermal imaging cameras to generate black ice imagery. During the water application process, 10 distinct experimental scenarios were created by varying the coverage area and positioning of the water distribution to ensure diverse black ice formation patterns. The image data collection followed a structured acquisition procedure. The experimental setup was configured as follows: TPV-IAHDR thermal cameras were used to capture the entire process of black ice formation from the beginning in a video with a resolution of 1280 × 720. These thermal camera images are used for training image dataset, by sampling and cropping frames at intervals of 200 ms. This established total 1156 black ice road images for 10 different cases and then, these images were divided into training, validation, and test datasets according to a ratio of 6:2:2. Therefore, the thermal road black ice dataset constructed in this paper is as shown in Table 5. The image dataset of the thermal road black ice was generated on asphalt roads and cement roads for different cases.

Table 5: Number of infrared black ice road surface images.

Dataset Type	Number of Images
Train Dataset	697
Validation Dataset	229
Test Dataset	229
Total Dataset	1156

The examples of the thermal road black ice image dataset constructed in this paper are shown in Fig. 9. Fig. 9a displays three example images of black ice generated on asphalt roads, and Fig. 9b shows seven example images of black ice generated on cement roads.



(**a**) Asphalt cases

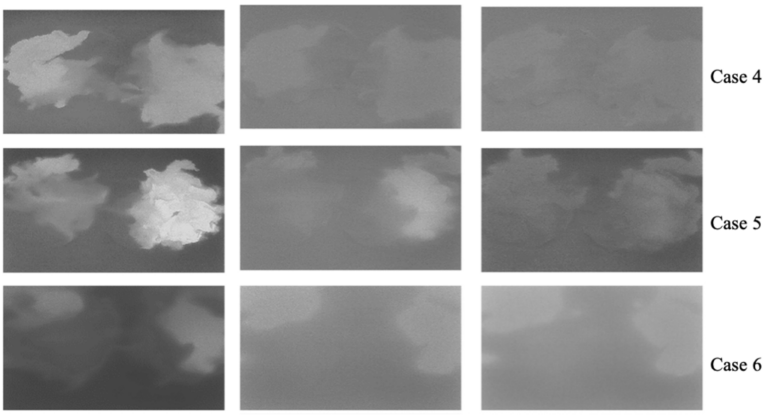
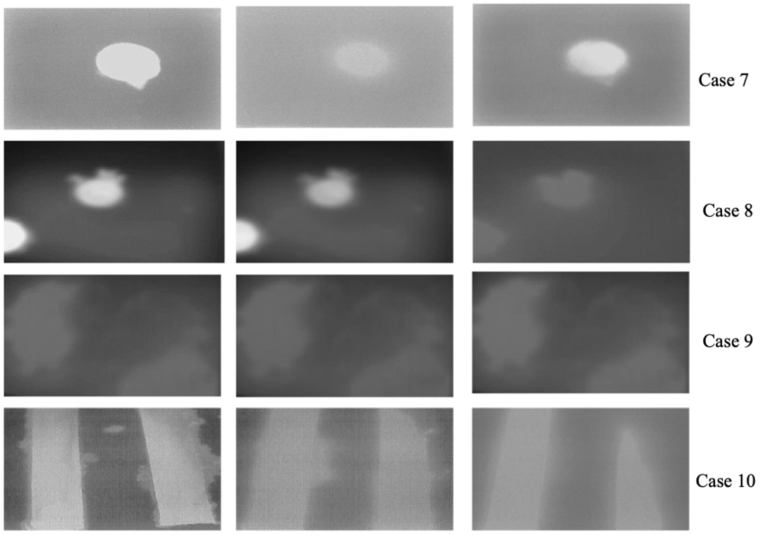


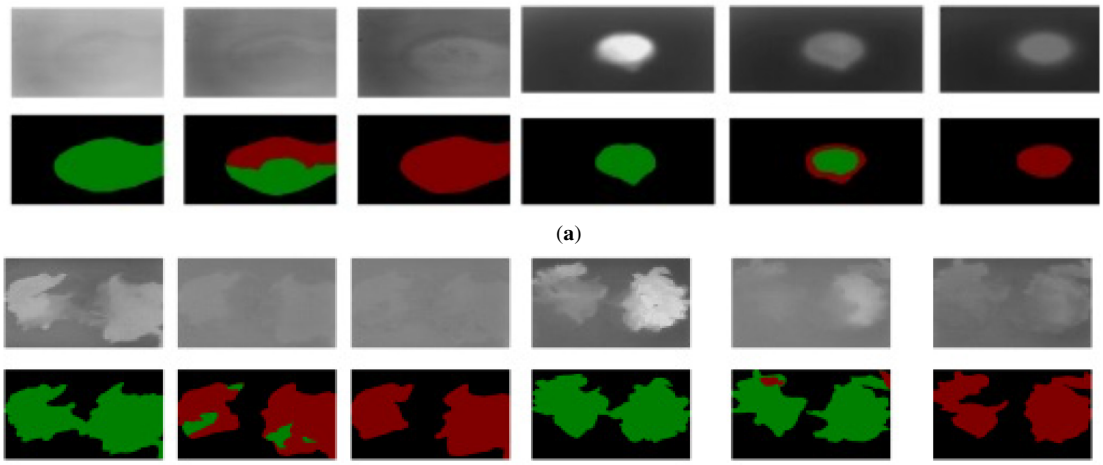
Figure 9: (Continued)



(**b**) Cement cases

Figure 9: Examples of infrared road black ice images in (**a**) 3 asphalt cases, (**b**) 7 cement cases.

These images are labeled by open-source image annotation tool, LabelMe [32] as shown in Fig. 10. This displays some of the original images used in the paper along with their corresponding labeled masked images.



(**a**)

(**b**)

Figure 10: Examples of thermal infrared input images and corresponding segmentation overlay masks for black ice on asphalt road surfaces (**a**) and cement road surfaces (**b**). In each overlay mask, green denotes Ground Truth (GT) annotation only (False Negative, FN), red denotes model prediction only (False Positive, FP), and green + red overlap indicates True Positive (TP) regions. (**a**) Asphalt cases—Col 1: Failure Case 1 (near-zero thermal contrast, severe FN); Col 2: Partial Failure Case 1 (boundary-region FN); Col 3: Failure Case 2 (boundary mismatch, FP-dominant); Col 4: Successful detection (TP); Col 5: Partial FN at ice patch periphery; Col 6: Failure Case 3 (FP from ambiguous thermal pattern). (**b**) Cement cases—Col 1: Failure Case 2 (irregular multi-patch, FN-dominant); Col 2: Failure Case 2 (fragmented boundary prediction); Col 3: Failure Case 3 (large FP, no GT correspondence); Col 4: Failure Case 2 (multi-patch FN); Col 5: Partial FP (minor spurious prediction); Col 6: Failure Case 3 (FP-dominant, road surface thermal ambiguity).

Qualitative Analysis and Failure Case Examination

Fig. 10 presents representative segmentation overlays illustrating both successful detection and characteristic failure cases across asphalt and cement road surfaces. In each overlay, green indicates False Negatives (FN), red indicates False Positives (FP), and green–red overlap indicates True Positives (TP).

- Failure Case 1: Near-Zero Thermal Contrast (False Negatives). When the ambient temperature approaches −1°C, the thermal contrast between black ice and dry road drops below 1°C as shown in Fig. 10a Col 1. Because this minimizes the thermal gradient, the boundaries become ambiguous, causing the model to severely underpredict the ice regions (false negatives). Fig. 10a Col 2 shows a partial failure where the model detects the high-contrast core but misses the faint boundaries, confirming its sensitivity to thermal grading.
- Failure Case 2: Irregular Boundaries and Multi-Patch Formations. For diffuse ice with gradual thermal boundaries, the model tends to over-predict spatial extent on asphalt (Fig. 10a Col 3, false positives) or produces fragmented predictions on cement (Fig. 10b Cols 1, 2, 4, false negatives). This discrepancy occurs because cement has a higher emissivity than asphalt, resulting in narrower and more challenging thermal gradients at the boundary.
- Failure Case 3: Ambiguous Thermal Patterns (False Positives). The model occasionally generates spurious positive predictions in non-ice regions. Thermal patterns caused by residual heat, texture variations, or material transitions can closely mimic the emissivity signature of black ice (Fig. 10a Col 6, Fig. 10b Col 3, 6). Without temporal or spatial context, the model currently cannot reliably discriminate these thermally similar but structurally distinct regions.

Fig. 10a Col 4 demonstrates a successful detection reference, where a compact ice patch with sufficient thermal contrast yields a predominantly TP result, confirming reliable model behavior under favorable conditions. These failure modes motivate the future directions outlined in Section 5.

4.2 Experimental and Simulation Results Using Real-Time

4.2.1 Training and Development Platform

The experimental platform ran Ubuntu 18.04 LTS and equipped with four NVIDIA GeForce RTX 2080 Ti GPUs, each with 11 GB of memory. The deep learning implementation utilized Keras 3.12.1 and TensorFlow 2.20.2 frameworks with Python 3.12.1. Training was conducted with 100 epochs and variable batch sizes (1, 2, 4, 8, 16) to evaluate optimal performance conditions across different computational constraints.

***Loss Function Selection*:** The training process employed the standard categorical cross-entropy (CCE) loss with Adam optimizer (β_1 = 0.9, β_2 = 0.999, $\varepsilon = 10^{-8}$), utilizing an initial learning rate of 0.001 with exponential decay scheduling (decay rate = 0.96, decay steps = 1000) to accelerate convergence while preventing oscillations. The selection of the standard CCE loss as the training objective could be justified on two primary grounds. First, our controlled dataset of 1156 thermal images did not exhibit the severe class imbalance that typically requires alternative loss functions. Second, the main objective of this study was to demonstrate that the proposed architectural design, specially the adaptive multi-scale dilated convolutions with the 4→2→1 progression, is the principal driver of the observed segmentation performance. By using standard CCE as a baseline, we can demonstrate that the performance improvements are driven by our adaptive multi-scale dilated convolutions, rather than specialized optimization strategies.

We acknowledge, however, that this selection of the standard CCE may slightly limit the model's sensitivity to extremely small and thin ice patches. Because CCE weights all pixels equally, the network may show reduced sensitivity to extremely small or thin black ice patches. This is particularly evident at formation boundaries where thermal contrast is faint, occasionally leading to false negatives at the patch

edges. In real-world outdoor conditions, where black ice occupies a much smaller pixel proportion, class imbalance-aware alternatives like Focal Loss or Dice Loss could significantly improve detection sensitivity. Therefore, systematically comparing these loss functions is a crucial direction for our future work as discussed in Section 5.

For edge deployment validation, the trained models were converted to TensorFlow Lite format and deployed on the NVIDIA Jetson Nano Developer Kit (128-core Maxwell GPU, 4 GB LPDDR4 memory), representing a resource-constrained embedded platform typical of automotive and roadside monitoring applications. All inference benchmarks were conducted under maximum performance mode (10 W power budget) with GPU acceleration enabled.

4.2.2 Performance Evaluation Metrics

The segmentation performance was assessed using a comprehensive set of evaluation metrics. The main metrics concerned are Mean Intersection over Union (mIoU) for overall segmentation quality and class-specific Black-Ice IoU for safety-critical detection accuracy. The Intersection over Union (IoU) quantifies the overlap between predicted segmentation results and ground truth annotations for individual classes. Black-Ice IoU specifically measures the segmentation accuracy for the critical black ice class alone, defined as

$$\text{Black-Ice IoU} = \frac{|P_{BI} \cap G_{BI}|}{|P_{BI} \cup G_{BI}|} \tag{11}$$

where P_{BI} represents the predicted black ice mask and G_{BI} denotes the ground truth black ice mask. Note that an individual Black-Ice IoU specifically measures the segmentation accuracy for the critical black ice class only. The mIoU represents the averaged IoU values across all segmentation classes (background, road surface, and black ice regions), providing a comprehensive performance indicator. The mathematical formulation is defined in Eq. (12).

$$m\text{IoU} = \frac{1}{K+1} \sum_{i=1}^{K} \frac{\mathrm{X}_{ii}}{T_i + \sum_{j=1}^{K} (\mathrm{X}_{ji} - \mathrm{X}_{ii})} \tag{12}$$

where K represents the number of classes, T_i denotes the total pixel count for class i, X_{ii} represents the true positive pixels and X_{ji} represents the false positive pixels from true class j misclassified as class i. This dual-metric approach ensures comprehensive evaluation: mIoU reflects general segmentation robustness across all diverse environmental conditions (general model robustness), while Black-Ice IoU specifically measures detection accuracy for safety-critical detection performance (class-specific performance assessment). In addition, in order to provide a more thorough evaluation of detection performance, particularly for small and irregular black ice patches, we additionally report Precision, Recall, and F1-Score for the black ice class as defined in Eq. (13)

$$Precision = \frac{TP}{TP + FP}, Recall = \frac{TP}{TP + FN}, F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{13}$$

where TP, FP, and FN denote true positive, false positive, and false negative pixels for the black ice class, respectively. Precision measures the proportion of predicted black ice pixels that are correct (low false alarm rate), while Recall measures the proportion of actual black ice pixels that are detected (low miss rate). The *F1-Score* provides a harmonic mean balancing these two objectives, which is critical in safety applications where both false alarms and missed detections carry operational consequences.

4.2.3 Comparative Performance Analysis

AdaMsDCNet achieves real-time black ice area segmentation on the NVIDIA Jetson Nano platform, processing approximately 3.94~5.63 frames per second (FPS) at a 576 × 768 resolution, satisfying the operational demands for onboard roadway monitoring. Table 6 represents the comprehensive performance comparisons between the proposed AdaMsDCNet architectures and established semantic segmentation models on the thermal black ice dataset. The evaluation encompasses computational efficiency (FPS on Jetson Nano), model complexity (parameter count), and segmentation accuracy (mIoU and Black-Ice IoU), and three additional class-specific detection metrics—Precision, Recall, and F1-Score for the black ice class across two batch size configurations (8 and 16). These additional metrics provide a more thorough assessment of detection reliability, as both false alarms (low Precision) and missed detections (low Recall) carry safety consequences in road ice monitoring applications.

Table 6: Performance comparison of semantic segmentation architectures on Jetson Nano embedded platform.

Network Model	FPS	Parameters (KB)	mIoU (%)		Black-Ice IoU (%)		Precision (%)		Recall (%)		F1-Score (%)	
			Batch Size		Batch Size		Batch Size		Batch Size		Batch Size	
			8	16	8	16	8	16	8	16	8	16
U-Net	0.24	31,055	69.69	–	61.38	–	86.41	–	67.94	–	76.07	–
PSPnet	0.14	134,325	85.85	–	83.27	–	93.43	–	88.45	–	90.87	–
DeepLabV3+	0.19	41,253	93.65	–	88.20	–	94.32	–	93.15	–	93.73	–
ENet	1.95	371	94.35	94.36	93.81	93.10	96.93	96.56	96.68	96.29	96.81	96.43
LinkNet	3.72	11,555	95.39	95.48	94.37	94.33	97.16	97.14	97.05	97.03	97.10	97.08
AdaMsDCNet_2	5.63	492	95.95	95.49	94.92	94.52	97.29	97.07	97.50	97.29	97.39	97.18
AdaMsDCNet_3	5.35	687	96.26	96.04	95.21	94.97	97.45	97.32	97.64	97.52	97.55	97.42
AdaMsDCNet_4	5.04	882	96.34	96.10	95.36	95.09	97.53	97.39	97.72	97.58	97.62	97.48
AdaMsDCNet_5	4.98	1077	96.41	96.18	95.47	95.11	97.55	97.35	97.82	97.64	97.68	97.49
AdaMsDCNet_6	4.62	1273	96.43	96.29	95.39	95.27	97.50	97.44	97.78	97.72	97.64	97.58
AdaMsDCNet_7	4.32	1468	96.44	96.32	95.45	95.34	97.54	97.47	97.81	97.75	97.67	97.61
AdaMsDCNet_8	3.78	1663	96.46	96.35	95.49	95.29	97.56	97.45	97.83	97.73	97.69	97.59
AdaMsDCNet_9	3.94	1858	96.47	96.38	95.48	95.42	97.55	97.52	97.82	97.79	97.69	97.65

As shown in Table 6, inference vary significantly across models. U-Net (0.24 FPS), PSPNet (0.14 FPS), and DeepLabV3+ (0.19 FPS) all fail to meet real-time requirements on Jetson Nano, as their large model size and computational demands result in FPS values well below 1. On the other hand, ENet and LinkNet achieve FPS values of 1.95 and 3.72, respectively, which means they can perform real-time black ice area segmentation. The experimental results in Table 6 demonstrate the superior performance of the proposed AdaMsDCNet series over conventional semantic segmentation architectures across multiple evaluation criteria. The comprehensive evaluation encompasses both computational efficiency (FPS) and segmentation accuracy (mIoU and class-specific black ice IoU) on the NVIDIA Jetson Nano embedded platform.

***Computational Efficiency Performance*:** The experimental results demonstrate substantial computational advantages of the AdaMsDCNet series over conventional architectures. Traditional deep segmentation models (U-Net, PSPNet, DeepLabv3+) exhibit severely limited real-time capabilities with FPS values below 0.25, rendering them impractical for time-critical hazard detection applications. The large parameter counts (31~134M) and memory requirements exceed the computational capacity of embedded platforms, necessitating cloud-based processing with associated latency penalties. In contrast, all AdaMsDCNet variants achieve remarkable real-time performance, with processing speeds ranging from 3.94 to 5.63 FPS, substantially outperforming traditional deep learning models. Notably, U-Net, PSPNet, and DeepLabV3+ demonstrate severely limited real-time capabilities with FPS values of 0.24, 0.14, and 0.19, respectively, rendering them

unsuitable for practical deployment scenarios requiring immediate hazard detection. The progressive performance scaling across AdaMsDCNet variants (from AdaMsDCNet_2 to AdaMsDCNet_9) reveals an optimal trade-off between accuracy and computational efficiency. Even the most compact variant, AdaMsDCNet_2, achieves the highest processing speed of 5.63 FPS and surpasses LinkNet in mIoU (+0.56 pp), Black-Ice IoU (+0.55 pp), and F1-Score (+0.29 pp), while being 23.5 times smaller in parameter count. It demonstrates that the AdaMsDCNet design achieves superior accuracy with substantially lower computational overhead. Meanwhile, AdaMsDCNet_9 demonstrates peak accuracy performance with acceptable real-time processing at 3.94 FPS.

***Segmentation Accuracy Performance*:** The mIoU evaluation establishes the superior accuracy of AdaMsDCNet architectures across all environmental conditions. AdaMsDCNet_9 achieves the highest overall performance with 96.47% mIoU (batch size 8), representing substantial improvements over baseline methods: +26.78 percentage points over U-Net, +10.63 over PSPNet, +2.832 over DeepLabv3+, and +1.09 over LinkNet. Critically, U-Net exhibits a Recall of only 67.94%, a gap of 29.88 pp below AdaMsDCNet_9, which indicates that approximately one-third of actual black ice pixels are missed. Such a miss rate is unacceptable in safety-critical road monitoring applications. Against DeepLabV3+, AdaMsDCNet_9 achieves +2.82 pp mIoU and +4.67 pp Recall improvement, and against LinkNet, +1.08 pp mIoU and +0.77 pp Recall improvement. This accuracy enhancement demonstrates that the proposed multi-scale feature fusion mechanism effectively compensates for the aggressive parameter reduction, capturing both fine-grained texture details and broad contextual information critical for distinguishing subtle black ice formations from wet pavement.

***Batch Size Robustness*:** The mIoU variation between batch size 8 and 16 across the AdaMsDCNet series is limited to an average of 0.18 percentage points (maximum 0.21 pp for AdaMsDCNet_2), and the Black-Ice IoU variation averages 0.24 pp (maximum 0.31 pp). This minimal degradation under differing batch configurations demonstrates stable performance under varying computational conditions, providing indirect evidence of deployment robustness. We note that U-Net, PSPNet, and DeepLabV3+ could not be evaluated at batch size 16 due to GPU memory constraints on the Jetson Nano (4GB LPDDR4), further confirming the practical deployment advantages of the AdaMsDCNet series.

It should be acknowledged that the current evaluation was conducted on a dataset collected under controlled indoor freezing conditions. The effects of real-world environmental noise, including ambient thermal interference from passing vehicles, precipitation-induced surface temperature variations, and sensor thermal drift, on both inference speed and segmentation accuracy were not directly measured in this study. Thermal infrared imaging provides an inherent advantage over optical sensing by relying on emissivity-based temperature gradients rather than reflected light, thus maintaining robustness to illumination-induced noise. Nevertheless, a comprehensive evaluation under uncontrolled outdoor conditions remains an important direction for future work, as discussed in Section 5.

The class-specific Black-Ice IoU metric reveals exceptional detection capability for the safety-critical hazard class. AdaMsDCNet_9 attains 95.48% Black-Ice IoU, outperforming all comparison architectures including resource-intensive models. This represents a +34.1% point improvement over U-Net (61.38%) and +1.11 over the lightweight LinkNet baseline (94.37%). The consistent superiority across both mIoU and Black-Ice IoU metrics indicates that the adaptive dilation strategy effectively mitigates the checkerboard artifacts associated with naive dilated convolution implementations, preserving precise boundary delineation for irregular ice patches. Fig. 11a shows the Pareto frontier analysis, demonstrating that all AdaMsDCNet variants simultaneously outperform baseline architectures in both accuracy and inference speed. Notably, AdaMsDCNet_2 achieves 5.63 FPS, the fastest among all evaluated models, while maintaining 95.95% mIoU, surpassing LinkNet by 0.56 percentage points and 1.91 FPS. The model complexity analysis in Fig. 11b

reveals that AdaMsDCNet_9 attains state-of-the-art accuracy (96.47% mIoU) with merely 1.86M parameters, representing an order-of-magnitude reduction compared to conventional architectures (31~134M parameters).

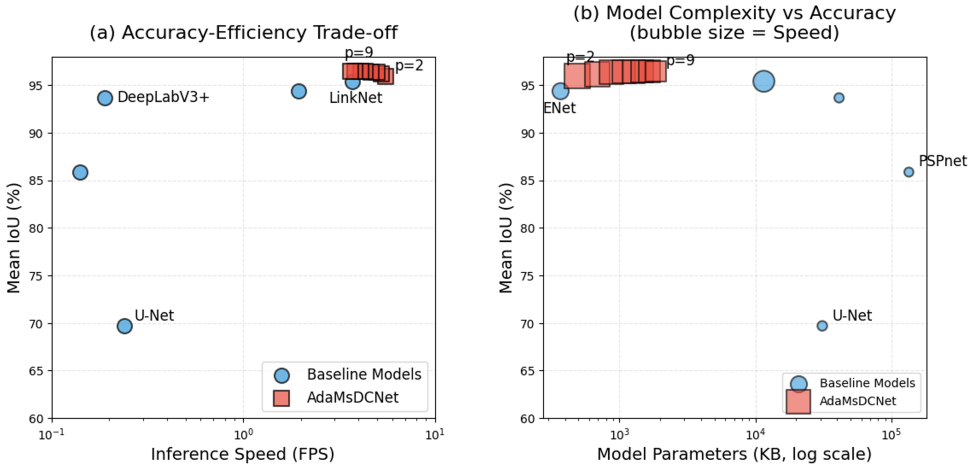


Figure 11: Performance comparison of semantic segmentation architectures for real-time black ice detection on Jetson Nano embedded platform. (**a**) Accuracy-efficiency trade-off analysis, (**b**) Model complexity.

***Scalability and Robustness Performance*:** The progressive performance scaling across AdaMsDCNet variants (subscripts 2–9 indicating parallel path counts) reveals optimal accuracy-efficiency trade-offs. Fig. 11a illustrates the Pareto frontier between segmentation accuracy and inference speed, demonstrating that mid-range configurations (AdaMsDCNet_5 to AdaMsDCNet_7) provide balanced performance for most deployment scenarios. AdaMsDCNet_2 offers the fastest processing (5.63 FPS) with competitive accuracy (95.95% mIoU), suitable for applications prioritizing latency minimization. Conversely, AdaMsDCNet_9 maximizes detection precision (96.47% mIoU, 95.48% Black-Ice IoU) while maintaining acceptable real-time performance (3.94 FPS), appropriate for safety-critical systems where false negatives carry severe consequences.

Batch size stability analysis reveals minimal performance degradation across computational configurations. The average mIoU variation between batch sizes 8 and 16 is only 0.18% points (σ = 0.21), and Black-Ice IoU varies by 0.24% points (σ = 0.31), indicating robust generalization independent of inference batch configurations. This stability enables flexible deployment adaptation to varying hardware constraints and power budgets in practical automotive environments. Fig. 12 presents the scalability characteristics of the AdaMsDCNet series, indicating that mid-range configurations (p = 5~7) provide optimal accuracy-efficiency trade-offs for most deployment scenarios.

***Real-Time Deployment Demonstration*:** The deployment validation confirms that AdaMsDCNet operates continuously for extended monitoring periods (>8 h) without memory leaks or performance degradation, consuming approximately 7.2 W average power (within the 10 W thermal design power of Jetson Nano) [33]. The system achieves end-to-end latency of approximately 320 ms (including image acquisition, preprocessing, inference, and alert generation), satisfying the temporal requirements for proactive driver warnings at highway speeds (<1-s response time for vehicles traveling at 100 km/h provides >27 m reaction distance).

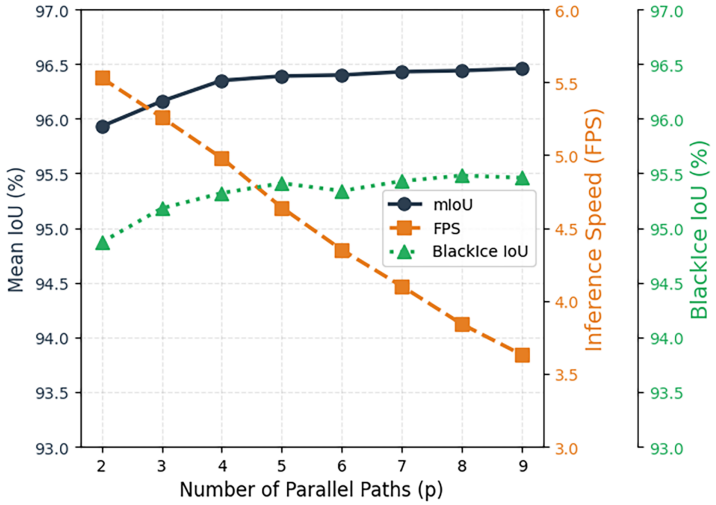


Figure 12: Scalability and robustness performance of AdaMsDCNet variants.

5 Conclusion

This paper presents AdaMsDCNet, a novel adaptive multi-scale dilated convolution network specifically optimized for real-time black ice detection on edge computing platforms. The proposed architecture addresses critical limitations of existing semantic segmentation approaches by achieving an optimal balance between detection accuracy and computational efficiency, enabling practical deployment on resource-constrained embedded systems for intelligent transportation safety applications. The core innovation lies in the adaptive multi-scale dilation strategy. By systematically reducing dilation rate increments at deeper network layers (4→2→1 progression), AdaMsDCNet preserves dense spatial sampling for small irregular ice patches while capturing broad contextual information at higher resolutions. Combining with controlled channel expansion and parallel feature fusion, this design concept enables the network to extract multi-scale representations with substantially reduced parameter counts (0.49~1.86M parameters) compared to conventional segmentation models (31~134M parameters). Experimental validation on a thermal infrared dataset of 1156 annotated black ice images demonstrates superior performance across multiple evaluation criteria. The AdaMsDCNet_9 achieves 96.47% mIoU and 95.48% Black-Ice IoU, 97.55% Precision, 97.82% Recall, and 97.69% F1-Score, which outperforms U-Net (+26.78 pp mIoU, +29.88 pp Recall), DeepLabv3+ (+2.82 pp mIoU), and LinkNet (+1.08 pp mIoU), while maintaining real-time inference speeds of 3.94~5.63 FPS on the NVIDIA Jetson Nano embedded GPU, representing 15~23 times acceleration over traditional deep learning models. The cloud–edge collaborative warning architecture demonstrates practical feasibility by achieving end-to-end latency under 320 ms with 7.2 W average power consumption, enabling continuous 24-h monitoring with automated severity classification and LED warning displays. The comprehensive ablation studies reveal (1) adaptive dilation rates improve small object segmentation by 1.09% IoU (LinkNet vs. AdaMsDCNet_9), (2) parallel multi-scale feature fusion outperforms sequential aggregation by 1.8% mIoU (LinkNet vs. AdaMsDCNet_9), (3) controlled channel growth reduces parameters by 83.9% with minimal accuracy degradation (compared to LinkNet) and (4) batch size stability (σ = 0.21% mIoU variance) ensures robust deployment flexibility.

***Limitations and Future Work*:** Several important limitations of this study should be acknowledged. First, the current dataset employed in our work was collected under controlled indoor freezing conditions using simulated water application, which may not fully represent the thermal diversity of real-world road environments. To ensure the model generalizes well, it must be further evaluated against unpredictable

outdoor factors such as the heat emitted by passing vehicles, weather-induced temperature shifts, sensor drift over time, and unusual thermal patterns near bridges or tunnels. Second, qualitative analysis of failure cases identifies three specific scenarios in which the model encounters difficulty: (i) near-zero thermal contrast conditions (~−1°C ambient temperature), where the temperature differential between black ice and dry asphalt drops below 1°C, causing False Negatives at ice patch boundaries; (ii) irregular boundary and multi-patch ice formations with gradual thermal transitions, where the model produces fragmented or boundary-mismatched predictions; and (iii) road surface regions exhibiting thermal emission patterns similar to black ice, such as material transitions on cement surfaces, which trigger localized False Positive predictions. Third, regarding the training objective, the standard categorical cross-entropy (CCE) loss was adopted as it provided the highest training stability and allowed performance improvements to be attributed directly to the proposed adaptive multi-scale dilated convolution architecture, rather than to specialized optimization strategies. However, because CCE weights all pixels equally, the network may struggle to detect extremely small or thin black ice patches where thermal contrast is low, occasionally causing false negatives at patch edges. In real-world outdoor conditions, where black ice occupies a much smaller pixel proportion, class imbalance-aware alternatives such as Focal Loss or Dice Loss could significantly improve detection sensitivity. To overcome these limitations, our future work will pursue several directions: (1) collecting outdoor thermal images across a variety of weather conditions (including rain, snow, and fog) to better evaluate model generalization; (2) systematic comparing loss functions such as Focal Loss and Dice Loss to enhance detection of small and sparse black ice patches; (3) exploring multi-frame temporal sequences to track dynamic ice formation; and (4) developing post-processing methods to filter out false alarms caused by road surface structures. These efforts aim to advance AdaMsDCNet toward practical deployment in real-world intelligent transportation safety systems.

Acknowledgement: This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP)-Innovative Human Resource Development for Local Intellectualization Program Grant funded by the Korea government (MSIT) (IITP-2026-RS2024-00439292).

Funding Statement: This work was funded by the Institute of Information & Communications Technology Planning & Evaluation (IITP)-Innovative Human Resource Development for Local Intellectualization Program.

Author Contributions: The authors confirm contribution to the paper as follows. Conceptualization, Yeonwoo Lee and Sun-Kyoung Kang; methodology, Yeonwoo Lee and Sun-Kyoung Kang; software, Sun-Kyoung Kang; validation, Yeonwoo Lee and Sun-Kyoung Kang; formal analysis, Yeonwoo Lee; investigation, Sun-Kyoung Kang; resources, Yeonwoo Lee and Sun-Kyoung Kang; data curation, Sun-Kyoung Kang; writing—original draft preparation, Sun-Kyoung Kang; writing—review and editing, Yeonwoo Lee; visualization, Yeonwoo Lee; supervision, Yeonwoo Lee; project administration, Sun-Kyoung Kang; funding acquisition, Sun-Kyoung Kang. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: Not applicable.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Lee H, Hwang K, Kang M, Song J. Black ice detection using CNN for the prevention of accidents in automated vehicle. In: Proceedings of the 2020 International Conference on Computational Science and Computational Intelligence (CSCI); 2020 Dec 16–18; Las Vegas, NV, USA. p. 1189–92. doi:10.1109/csci51800.2020.00222.

2. Xu Y, Yao D, Ren X, Dai Y. Intelligent black ice detection and alert system using thermal imaging camera and drone. In: Proceedings of the 2021 IEEE 23rd International Conference on High Performance Computing & Communications; 7th International Conference on Data Science & Systems; 19th International Conference on Smart City; 7th International Conference on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys); 2021 Dec 20–22; Haikou, China. p. 2328–31. doi:10.1109/HPCC-DSS-SmartCity-DependSys53884.2021.00351.
3. Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. In: Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2015 Jun 7–12; Boston, MA, USA. p. 3431–40. doi:10.1109/CVPR.2015.7298965.
4. Ronneberger O, Fischer P, Brox T. U-Net: convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015. Cham, Switzerland: Springer International Publishing; 2015. p. 234–41. doi:10.1007/978-3-319-24574-4_28.
5. Chen LC, Zhu Y, Papandreou G, Schroff F, Adam H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Computer Vision—ECCV 2018. Cham, Switzerland: Springer International Publishing; 2018. p. 833–51. doi:10.1007/978-3-030-01234-2_49.
6. Zhao H, Shi J, Qi X, Wang X, Jia J. Pyramid scene parsing network. In: Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2017 Jul 21–26; Honolulu, HI, USA. p. 6230–9. doi:10.1109/CVPR.2017.660.
7. Paszke A, Chaurasia A, Kim S, Culurciello E. ENet: a deep neural network architecture for real-time semantic segmentation. arXiv:1606.02147. 2016.
8. Chaurasia A, Culurciello E. LinkNet: exploiting encoder representations for efficient semantic segmentation. In: Proceedings of the 2017 IEEE Visual Communications and Image Processing (VCIP); 2017 Dec 10–13; Petersburg, FL, USA. p. 1–4. doi:10.1109/VCIP.2017.8305148.
9. Wang P, Chen P, Yuan Y, Liu D, Huang Z, Hou X, et al. Understanding convolution for semantic segmentation. In: Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV); 2018 Mar 12–15; Lake Tahoe, NV, USA. p. 1451–60. doi:10.1109/WACV.2018.00163.
10. Yu F, Koltun V. Multi-scale context aggregation by dilated convolutions. In: Proceedings of the International Conference on Learning Representations (ICLR 2016); 2016 May 2–4; San Juan, Puerto Rico. p. 1–6.
11. Ma X, Ruan C. Method for black ice detection on roads using tri-wavelength backscattering measurements. Appl Opt. 2020;59(24):7242–6. doi:10.1364/ao.398772.
12. Tabatabai H, Aljuboori M. A novel concrete-based sensor for detection of ice and water on roads and bridges. Sensors. 2017;17(12):2912. doi:10.3390/s17122912.
13. Zhao J, Wu H, Chen L. Road surface state recognition based on SVM optimization and image segmentation processing. J Adv Transp. 2017;2017(6):6458495. doi:10.1155/2017/6458495.
14. Abdalla YE, Iqbal MT, Shehata M. Black ice detection system using kinect. In: Proceedings of the 2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE); 2017 Apr 30–May 3; Windsor, ON, Canada. p. 1–4. doi:10.1109/CCECE.2017.7946722.
15. Kim H, Kim S, Park J, Kim Y. Vision-based black ice identification using lightweight CNN and transfer learning. ICT Express. 2026;12(1):180–5. doi:10.1016/j.icte.2026.01.001.
16. Kim J, Kim E, Kim D. A black ice detection method based on 1-dimensional CNN using mmWave sensor backscattering. Remote Sens. 2022;14(20):5252. doi:10.3390/rs14205252.
17. Liu X, Wang J, Li J. URTSegNet: a real-time segmentation network of unstructured road at night based on thermal infrared images for autonomous robot system. Control Eng Pract. 2023;137(12):105560. doi:10.1016/j.conengprac.2023.105560.
18. Howard A, Sandler M, Chen B, Wang W, Chen LC, Tan M, et al. Searching for MobileNetV3. In: Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV); 2019 Oct 27–Nov 2; Seoul, Republic of Korea. p. 1314–24. doi:10.1109/iccv.2019.00140.
19. Yu C, Gao C, Wang J, Yu G, Shen C, Sang N. BiSeNet V2: bilateral network with guided aggregation for real-time semantic segmentation. Int J Comput Vis. 2021;129(11):3051–68. doi:10.1007/s11263-021-01515-2.

20. Ding E, Feng H, Gou C, Han J, Wang J, Wang J, et al. RTFormer: efficient design for real-time semantic segmentation with transformer. In: Proceedings of the Advances in Neural Information Processing Systems 35; 2022 Nov 28–Dec 9; New Orleans, LA, USA. p. 7423–36. doi:10.52202/068431-0539.
21. Xie E, Wang W, Yu Z, Anandkumar A, Alvarez JM, Luo P. SegFormer: simple and efficient design for semantic segmentation with transformers. Adv Neural Inf Process Syst. 2021;34:12077–90.
22. Xu J, Xiong Z, Bhattacharyya SP. PIDNet: a real-time semantic segmentation network inspired by PID controllers. In: Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2023 Jun 17–24; Vancouver, BC, Canada. p. 19529–39. doi:10.1109/CVPR52729.2023.01871.
23. Gao R. Rethinking dilated convolution for real-time semantic segmentation. In: Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW); 2023 Jun 17–24; Vancouver, BC, Canada. p. 4675–84. doi:10.1109/CVPRW59228.2023.00493.
24. Zhang Y, Zhang X, Miao D, Yu H. Real-time semantic segmentation of road scenes via hybrid dilated grouping network. Int J Netw Dyn Intell. 2025;4(1):100006. doi:10.53941/ijndi.2025.100006.
25. Qiu Y, Xu G, Gao G, Guo Z, Yu Y, Lin CW. Efficient semantic segmentation via lightweight multiple-information interaction network. arXiv:2410.02224. 2024.
26. Lei T, Geng X, Ning H, Lv Z, Gong M, Jin Y, et al. Ultralightweight spatial-spectral feature cooperation network for change detection in remote sensing images. IEEE Trans Geosci Remote Sens. 2023;61(4):4402114. doi:10.1109/TGRS.2023.3261273.
27. Lin TY, Goyal P, Girshick R, He K, Dollár P. Focal loss for dense object detection. In: Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV); 2017 Oct 22–29; Venice, Italy. p. 2999–3007. doi:10.1109/ICCV.2017.324.
28. Daud AA. Edge computing—enabled road condition monitoring: system development and evaluation [master's thesis]. Columbia, MO, USA: University of Missouri Libraries, M. S.; 2023. doi:10.32469/10355/97086.
29. Pettirsch A, Garcia-Hernandez A. Overcoming data scarcity in roadside thermal imagery: a new dataset and weakly supervised incremental learning framework. Sensors. 2025;25(7):2340. doi:10.3390/s25072340.
30. Ren WQ, Qu YB, Dong C, Jing YQ, Sun H, Wu QH, et al. A survey on collaborative DNN inference for edge intelligence. Mach Intell Res. 2023;20(3):370–95. doi:10.1007/s11633-022-1391-7.
31. Odena A, Dumoulin V, Olah C. Deconvolution and checkerboard artifacts. Distill. 2016;1(10):e3. doi:10.23915/distill.00003.
32. Russell BC, Torralba A, Murphy KP, Freeman WT. LabelMe: a database and web-based tool for image annotation. Int J Comput Vis. 2008;77(1):157–73. doi:10.1007/s11263-007-0090-8.
33. NVIDIA. Jetson nano developer kit user guide. Santa Clara, CA, USA: NVIDIA Corporation; 2019 [cited 2026 Jan 1]. Available from: https://developer.nvidia.com/embedded/jetson-nano-developer-kit.