electronics

MDPI

Article

Efficient Smoke Segmentation Using Multiscale Convolutions and Multiview Attention Mechanisms

Xuesong Liu and Emmett J. Ientilucci *

Chester F. Carlson Center for Imaging Science, Rochester Institute of Technology, Rochester, NY 14623, USA; xl2088@rit.edu

* Correspondence: ejipci@rit.edu

Abstract

Efficient segmentation of smoke plumes is crucial for environmental monitoring and industrial safety. Existing models often face high computational demands and limited adaptability to diverse smoke appearances. To address these issues, we propose SmokeNet, a deep learning architecture integrating multiscale convolutions, multiview linear attention, and layer-specific loss functions. Specifically, multiscale convolutions capture diverse smoke shapes by employing varying kernel sizes optimized for different plume orientations. Subsequently, multiview linear attention emphasizes spatial and channel-wise features relevant to smoke segmentation tasks. Additionally, layer-specific loss functions promote consistent feature refinement across network layers, facilitating accurate and robust segmentation. SmokeNet achieves a segmentation accuracy of 72.74% mean Intersection over Union (mIoU) on our newly introduced quarry blast smoke dataset and maintains comparable performance on three benchmark smoke datasets, reaching up to 76.45% mIoU on the Smoke100k dataset. With a computational complexity of only 0.34 M parameters and 0.07 Giga Floating Point Operations (GFLOPs), SmokeNet is suitable for real-time applications. Evaluations conducted across these datasets demonstrate SmokeNet's effectiveness and versatility in handling complex real-world scenarios.

Keywords: deep learning; semantic segmentation; smoke segmentation; quarry blast smoke

check for updates

Academic Editor: Maciej Ławryńczuk

Received: 19 May 2025
Revised: 24 June 2025
Accepted: 25 June 2025
Published: 27 June 2025

Citation: Liu, X.; Ientilucci, E.J. Efficient Smoke Segmentation Using Multiscale Convolutions and Multiview Attention Mechanisms. *Electronics* **2025**, *14*, 2593. https://doi.org/10.3390/electronics14132593

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Accurate smoke segmentation serves as an essential preliminary step for downstream applications, such as 3D smoke reconstruction, volume estimation, and quantification of pollutants and harmful gases, facilitating environmental monitoring and industrial safety. Specifically, quarry blasts generate nitrogen oxide (NO_x) emissions, posing substantial environmental and health concerns [1]. Quarry blasting produces smoke plumes characterized by irregular shapes, mixed textures of dust and debris, and varying opacity levels.

Smoke segmentation is a subtask within semantic segmentation, thus general semantic segmentation architectures such as UNet [2] and Fully Convolutional Networks (FCNs) [3] are frequently applied. Furthermore, efficient CNN architectures like MobileNet [4], EfficientNet [5], and ShuffleNet [6] have been developed to achieve performance through reduced computational complexity. Additionally, Vision Transformers (ViTs) [7] enhance the ability to capture global and contextual information, though they typically require higher computational resources. However, applying these general models

directly to smoke segmentation often encounters difficulties due to smoke's dynamic and variable characteristics.

To specifically address smoke segmentation challenges, several specialized methods adapted from the general segmentation approaches have been proposed to improve the segmentation performance. The Deep Smoke Segmentation (DSS) [8] model employs a two-path FCN to extract global context, enhancing segmentation accuracy but increasing computational complexity. Frizzi et al. [9] introduced a VGG16-based model that attempts to improve smoke plume segmentation performance compared to image processing techniques, yet this comes with higher model parameters and complexity. Nevertheless, real-time and resource-limited scenarios, such as quarry blast smoke segmentation, still present challenges due to persistent trade-offs between efficiency and accuracy. Recently, Yuan et al. [10] further advanced smoke segmentation by integrating attention-based modules inspired by transformer architectures. Their lightweight model incorporates spatial and channel attention, effectively reducing parameters and complexity while maintaining high segmentation accuracy.

Although Yuan's lightweight model achieves good performance on smoke segmentation tasks, it primarily focuses on chimney smoke and fire smoke, which mainly expand upward. In contrast, quarry smoke, due to its geometric characteristics, expands and spreads in multiple orientations, leading to more complicated scenarios. To further improve smoke segmentation efficiency with a low-parameter model, particularly for quarry smoke, we introduce SmokeNet, a UNet-based architecture specifically designed to meet the unique demands of smoke segmentation in both synthetic and real-world environments, with a particular focus on quarry smoke. Our contributions include the following:

- **Multiscale Convolutions with Rectangular Kernels:** Integrates a multiscale convolution module using rectangular-shaped kernels alongside standard kernels. Rectangular kernels are designed to capture spatial patterns that are suitable for the irregular and anisotropic characteristics of smoke. Specifically, vertically oriented rectangular kernels address tall, narrow smoke plumes commonly seen in wildfires or smoke directly ejected from drilled holes during quarry blasts. In contrast, horizontally oriented kernels address wide, low plumes typically observed when smoke leaks and spreads out from collapsed rock terrain after blasting.
- **Multiview Linear Attention:** Employs a multiview linear attention mechanism to efficiently enhance feature integration. Traditional attention mechanisms, such as those used in Vision Transformers, calculate attention weights through pairwise interactions, resulting in quadratic computational complexity with respect to input size. In contrast, linear attention approximates these interactions using linear projections, scaling linearly with input dimensions and reducing computational demands. The multiview design further enhances feature representation by applying attention separately across spatial (height and width) and channel dimensions through element-wise multiplication.
- **Layer-Specific Loss:** Incorporates a layer-specific loss strategy that applies additional supervision to intermediate layers, reducing feature discrepancies between convolutional and attention modules. Directly applying a dedicated loss between these two types of features ensures alignment and consistency across layers.

Compared with the recent lightweight smoke segmentation model by Yuan et al. [10], which achieved 75.57% mIoU with 0.88 M parameters and 1.15 GFLOPs, SmokeNet achieves higher segmentation accuracy (76.45% mIoU) with fewer parameters (0.34 M) and lower computational complexity (0.07 GFLOPs). These improvements demonstrate SmokeNet's suitability and enhanced efficiency, particularly for real-time segmentation in quarry blasting scenarios.

2. Related Work

2.1. Deep-Learning Methods for Smoke Segmentation

Recent deep-learning-based methods have advanced smoke detection and segmentation across diverse scenarios.

Image-Level Detection and Segmentation Datasets: Yin et al. [11] introduced a CNN for image-level smoke and fire detection, achieving strong recognition accuracy but without pixel-level outputs. Yuan et al. [12] added attention for temporal smoke localization in videos, yet segmentation evaluation was limited. Cheng et al. [13] created Smoke100k, a synthetic benchmark; however, it lacks complexity for real environments.

Outdoor Segmentation Frameworks: DeepSmoke [14] combined EfficientNet backbone with DeepLabv3+ to segment smoke in clear and hazy outdoor scenes, improving IoU and reducing false alarms. Frizzi et al. [9] proposed a CNN-based segmentation model trained on augmented in-vehicle and wildfire frames, showing improved boundary accuracy and reduced misclassification against clouds.

Early-Stage and Active Learning Approaches: DSS [8] fused coarse and fine branch FCNs for blurry smoke, performing well on synthetic and real data. FoSp network [15] introduced focus–separation modules for early-stage smoke, achieving high F-beta scores on SmokeSeg dataset. Pereira et al. [16] used active learning to reduce labeling needs while maintaining segmentation accuracy in smoke videos.

Wildland and UAV Monitoring: SmokeyNet [17] and its FIgLib dataset provided spatiotemporal models and nearly 25k images for wildfire smoke detection. A recent UAV wildfire segmentation study [18] applied transformer-based models on aerial smoke datasets, improving generalization across environments.

Transformer-Based and Real-Time Detection: Smoke detection [19] applied an RT-DETR real-time detector with triplet attention and pyramid fusion for low-resolution video smoke detection. For industrial alarm uses, a novel CNN architecture [9] trained on smoke data showed robust segmentation in real time.

These studies highlight advancements in smoke segmentation across diverse conditions, yet challenges remain in handling transparent smoke, variable lighting, and limited labeled data. Attention mechanisms and multiscale feature representations offer promising solutions to enhance model robustness and adaptability.

2.2. Advances in Attention Mechanisms and Multiscale Feature Representation

Attention mechanisms and multiscale feature representation have advanced semantic segmentation performance, each providing distinct advantages and facing unique challenges.

U-Net Series: Ronneberger et al. [2] introduced the original U-Net, effective in capturing local features through encoder-decoder structures with skip connections but limited in global context modeling. Subsequent works, such as UNet++ by Zhou et al. [20], enhanced multilevel feature integration but at increased computational complexity. Attention U-Net by Oktay et al. [21] integrated attention gates, enhancing the model's capability to suppress irrelevant regions but at the cost of additional computations. MA-Unet [22] further improved this architecture through multiscale attention mechanisms, offering a balance between computational cost and segmentation accuracy. Recent developments such as TransUNet [23] and Swin-Unet [24] combined Transformer structures with U-Net, boosting global context capabilities while introducing greater computational overhead.

Vision Transformers (ViTs): Pure Transformer models, such as ViT [7] and DeiT [25], showed strong global context capture abilities but often required large datasets for effective training and struggled with fine-scale segmentation tasks. Hierarchical Vision Transformers, including Swin Transformer [26] and Pyramid Vision Transformer (PVT) [27], addressed these issues by efficiently capturing multiscale contexts with hierarchical archi-

tectures. SegFormer [28] integrated hierarchical Transformers and CNN features, balancing segmentation accuracy with computational efficiency.

Multiscale and Attention Hybrids: Attention-Augmented UNet (AA-UNet) [29] integrated Transformer-like attention within CNN-based segmentation models, effectively capturing contextual details but with increased computational demand. The Lawin Transformer [30] applied large-window attention mechanisms for multiscale feature aggregation, yielding strong segmentation performance but high computational complexity. Focal Transformer [31] dynamically adjusted attention granularity, achieving fine-grained detail capture with efficiency challenges. MERIT [32] utilized multiscale cascaded attention decoding, improving segmentation results at the expense of decoder complexity. The Twins-SVT [33] introduced spatially separable attention, reducing computational requirements while maintaining accuracy.

These approaches reveal inherent trade-offs between detail accuracy, global context, and computational cost, motivating further research into efficient, robust models for resource-constrained scenarios.

2.3. Enhancing Model Robustness and Computational Efficiency

Semantic segmentation in resource-constrained settings depends on models that balance accuracy and computational efficiency.

Mobile and Separable Convolution Architectures: MobileNet [4] utilizes depthwise separable convolutions to reduce parameters but may compromise fine details. ShuffleNet [6] introduces channel shuffling and grouped convolutions, effectively enhancing computational efficiency with modest accuracy trade-offs. EfficientNet [5] proposes a balanced scaling method across depth, width, and resolution, optimizing accuracy and efficiency.

Lightweight Semantic Segmentation Models: Fast-SCNN [34] achieves efficiency through efficient downsampling operations but may sacrifice detailed segmentation accuracy. LEDNet [35] applies asymmetric encoder–decoder architectures with attention mechanisms, offering a good balance between efficiency and accuracy for real-time tasks. BiSeNet and BiSeNet V2 [36,37] separate spatial and contextual branches, effectively handling real-time segmentation but increasing model complexity. MSGU-Net [38] integrates ghost modules within a multiscale U-Net design, achieving parameter reductions without compromising segmentation quality. LMSC-UNet [39] employs modified skip connections and MobileNetV2 modules for efficient segmentation.

Asymmetric and Attention-Augmented Architectures: FASSD-Net [40] utilizes dilated asymmetric convolutions for efficient segmentation, enhancing accuracy with minimal computational overhead. Improved Fast-SCNN variants [41] incorporate attention mechanisms to enhance segmentation performance. CA+ECA-PSPNet [42] integrates coordinate and channel attention within PSPNet, optimizing performance with low computational cost. HARD [43] targets hardware-constrained environments with efficient attention and dilation mechanisms, suitable for real-time deployment.

Recent efforts have specifically focused on smoke segmentation, prioritizing lightweight designs optimized for parameter efficiency. For instance, Yuan et al. [10] proposed an efficient model employing attention mechanisms tailored for smoke feature extraction. However, limited diversity in existing datasets restricts comprehensive validation, particularly for challenging scenarios like quarry smoke.

3. Methodology

SmokeNet is designed for smoke segmentation in complex environments, particularly addressing the dynamic characteristics of quarry smoke. Inspired by U-Net, SmokeNet

employs an encoder–decoder structure organized into six distinct stages, systematically reducing spatial resolution while increasing channel depth in the encoder and conversely reconstructing spatial resolution while reducing channel depth in the decoder. Specifically, the encoder begins with an initial spatial resolution of $W \times H$ with 4 channels at Stage 1, progressively reducing to spatial dimensions of $\frac{W}{32} \times \frac{H}{32}$ with 128 channels at Stage 6. Conversely, the decoder reconstructs the segmentation map by sequentially increasing spatial dimensions from $\frac{W}{32} \times \frac{H}{32}$ at Stage 6 to $W \times H$ at Stage 1, progressively reducing the number of channels. The encoder (Stages 1–6) focuses on extracting hierarchical features through multiscale convolutions and attention mechanisms, while the decoder (Stages 1–6) incorporates multiview linear attention mechanisms and skip connections to integrate detailed features from corresponding encoder stages. As illustrated in Figure 1, input tensor dimensions for each encoder stage and output tensor dimensions for each decoder stage are annotated. At each stage, feature tensors are divided into four equally sized channel chunks for individual processing. The initial input is a 3-channel RGB image, and the final output is a single-channel binary segmentation mask.

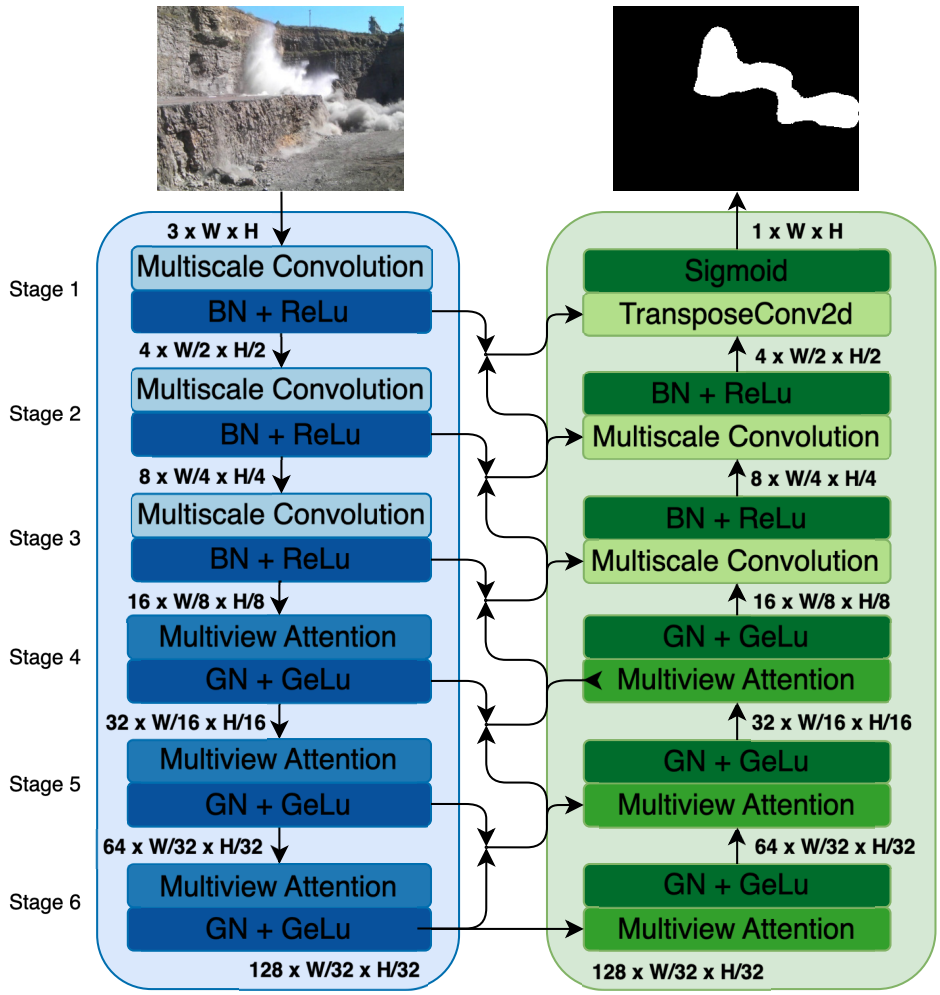

Figure 1. Overview of SmokeNet's encoder-decoder architecture. The encoder (**left**, blue) extracts hierarchical features via multiscale convolutions and multiview linear attention mechanisms. The decoder (**right**, green) reconstructs spatial resolution through transposed convolutions and skip connections, producing a binary segmentation mask.

3.1. Encoder

3.1.1. Multiscale Feature Extraction

The encoder stages (Stages 1–3) are responsible for extracting features across multiple spatial scales, as shown in Figures 1 and 2. At Stage 1, the original input tensor (3 × 256 × 256) is directly processed without channel-wise splitting. For subsequent stages (Stage 2 onward), input tensors are evenly divided into four chunks along the channel dimension before processing. Intermediate tensor dimensions at each step (Channels × Width × Height) are shown in Figure 2. This multiscale extraction is facilitated by a dedicated Multiscale Module, which processes input feature tensors through a series of convolutional operations with varying kernel sizes, followed by batch normalization and activation functions.

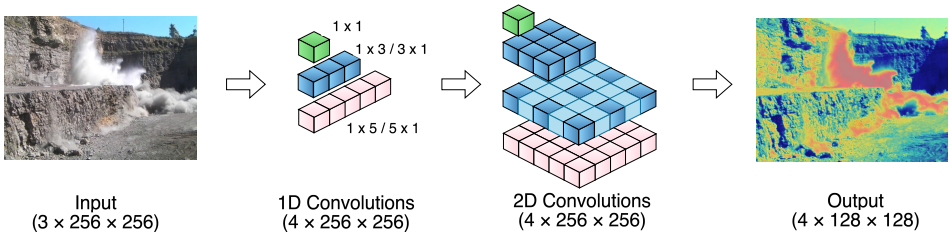

Figure 2. Detailed illustration of the Multiscale Module (Stage 1). Colors indicate different kernel sizes in 1D convolutions, which combine into corresponding rectangular kernels in 2D convolutions. The two shades of blue represent dilated convolutions.

The Multiscale Module employs 1D convolutional layers with diverse kernel sizes, including 1×1, 1×3, 3×1, 1×5, and 5×1. These convolutions are applied sequentially to the input tensor, resulting in multiple feature maps that capture different spatial extents and orientations of smoke plumes.

Rectangular kernels (1×3, 3×1, 1×5, and 5×1) were selected to specifically capture elongated and anisotropic features commonly observed in smoke plume structures. Smoke plumes from quarry blasts typically exhibit varying shapes, including vertically elongated plumes ejected directly from drilled holes and horizontally extended plumes spreading from collapsed terrain. Rectangular kernels target these directional characteristics directly, providing spatial feature extraction adapted to these anisotropic smoke shapes, compared to standard square kernels.

For instance, the 1×3 and 3×1 convolutions are adept at capturing elongated features in horizontal and vertical directions, respectively, while the 1×5 and 5×1 convolutions capture broader spatial contexts.

To construct larger and more complex 2D convolution operators efficiently, sequential 1D convolutions are employed. The equivalent 2D convolution operators, such as 3×3, 3×5, 5×3, and 5×5, are defined by applying two 1D convolutions in orthogonal directions:

$$\mathrm{Conv}_{3\times3} = \mathrm{Conv}_{3\times1} \circ \mathrm{Conv}_{1\times3}, \tag{1}$$

$$\mathrm{Conv}_{3\times5} = \mathrm{Conv}_{3\times1} \circ \mathrm{Conv}_{1\times5}, \tag{2}$$

$$\mathrm{Conv}_{5\times3} = \mathrm{Conv}_{5\times1} \circ \mathrm{Conv}_{1\times3}, \tag{3}$$

$$\mathrm{Conv}_{5\times5} = \mathrm{Conv}_{5\times1} \circ \mathrm{Conv}_{1\times5}. \tag{4}$$

As shown in Equations (1)–(4), each operator is constructed by first applying a 1D convolution along one axis, followed by another 1D convolution along the orthogonal axis. For instance, $\mathrm{Conv}_{3\times3}$ is achieved by applying $\mathrm{Conv}_{1\times3}$ (horizontal) followed by $\mathrm{Conv}_{3\times1}$ (vertical).

These operations, especially the rectangular kernel sizes targeting horizontal or vertical features, enable the module to capture a wide range of smoke shapes, from narrow and tall plumes common in campfires to wide and elongated patterns resulting from quarry blasts.

Let us denote the input tensor to the encoder as $\mathbf{F} \in \mathbb{R}^{N \times C \times H \times W}$, where N represents the batch size, C is the number of channels, and H and W denote the spatial height and width of the feature maps, respectively.

For each stage, including the Multiscale Module, the input—either from the original image (Stage 1) or the output of the previous stage—is split into four chunks along the channel dimension. Each chunk $\mathbf{F}_i \in \mathbb{R}^{N \times \frac{C}{4} \times H \times W}$ is processed independently using specific operations as follows:

$$\mathbf{F}_1' = \text{RELU}(\mathbf{F}_1), \quad \text{(identity mapping)} \tag{5}$$

$$\mathbf{F}_2' = \text{RELU}(\text{Conv}_{1\times1}(\mathbf{F}_2)), \quad \text{(1 } \times \text{ 1 convolution)} \tag{6}$$

$$\mathbf{F}_3' = \text{RELU}(\text{Conv}_{\text{selected}}(\mathbf{F}_3)), \quad \text{(non-dilated convolution)} \tag{7}$$

$$\mathbf{F}_4' = \text{RELU}\Big(\text{Conv}^{\text{dilated}}_{\text{selected}}(\mathbf{F}_4)\Big), \quad \text{(dilated convolution)} \tag{8}$$

The convolutional operations in $\text{Conv}_{\text{selected}}$ include a range of kernel sizes and their sequential combinations to emulate 2D convolutions:

$$\begin{aligned}\text{Convs} = \Big[&\text{Conv}_{1\times3},\ \text{Conv}_{3\times1},\ \text{Conv}_{1\times5},\ \text{Conv}_{5\times1},\\ &\text{Conv}_{3\times3},\ \text{Conv}_{3\times5},\ \text{Conv}_{5\times3},\ \text{Conv}_{5\times5}\Big].\end{aligned} \tag{9}$$

To address dimensional alignment, all outputs from the selected kernel operations are normalized to consistent dimensions ($N \times \frac{C}{4} \times H \times W$) before concatenation. This is achieved by applying a 1×1 convolution to adjust the channel count and using appropriate padding or cropping to match the spatial dimensions. These steps ensure compatibility during feature integration, preventing dimensional mismatches and enabling stable multiscale feature fusion, while maintaining efficiency across stages.

After processing, each chunk undergoes batch normalization to stabilize the learning process:

$$\mathbf{F}_i'' = \text{BatchNorm}\big(\mathbf{F}_i'\big), \quad \text{for } i = 1, 2, 3, 4. \tag{10}$$

The outputs from all four chunks are concatenated along the channel dimension to form the combined feature map:

$$\mathbf{F}' = \text{Concat}\big(\mathbf{F}_1'',\ \mathbf{F}_2'',\ \mathbf{F}_3'',\ \mathbf{F}_4''\big). \tag{11}$$

An identity mapping with activation is applied to the concatenated feature map to produce the final feature map:

$$\mathbf{F}_{\text{final}} = \text{RELU}\big(\text{Proj}\big(\mathbf{F}'\big)\big), \tag{12}$$

where $\text{Proj}(\cdot)$ is a 1×1 convolution if the channel dimensions of $\mathbf{F}$ and $\mathbf{F}'$ differ; otherwise, it is the identity function.

Finally, a 2×2 Max Pooling operation reduces the spatial dimensions:

$$\mathbf{F}_{\text{pooled}} = \text{MaxPool}(\mathbf{F}_{\text{final}},\ 2 \times 2). \tag{13}$$

By integrating batch normalization and RELU activation functions throughout the module, the encoder effectively captures both local and global contextual features while ensuring stable and efficient learning. The use of larger kernels and dilated convolutions in $\mathbf{F}_4'$ captures broader contextual information, and the inclusion of batch normalization after each path's output normalizes feature distributions, facilitating deeper network training.

As shown in Figure 2, the encoder of SmokeNet systematically extracts features through a combination of multiscale convolutions, dilated convolutions, activation functions, batch normalization, and strategic feature fusion. The incorporation of these elements enhances the model's ability to capture the variability and structural complexity of smoke plumes in various environments.

3.1.2. Multiview Linear Attention Mechanism

In encoder Stages 4–6, multiview linear attention mechanisms are integrated to refine the encoded features, as shown in Figure 3, focusing on essential elements while maintaining computational efficiency. In Stage 4, the input feature map (16 × 32 × 32), produced by encoder Stage 3, is processed by the multiview attention module, resulting in an output feature map of (32 × 16 × 16). Intermediate tensor dimensions at each step (Channels × Width × Height) are illustrated in Figure 3.

Multiview linear attention was selected due to its computational efficiency and ability to emphasize spatial and channel-wise features separately. Unlike traditional quadratic attention mechanisms, linear attention scales linearly with feature map dimensions, reducing computational demands. Applying attention separately along spatial (height and width) and channel dimensions allows the model to distinctly emphasize features important for delineating smoke plumes, thereby enhancing segmentation accuracy while maintaining low computational complexity.

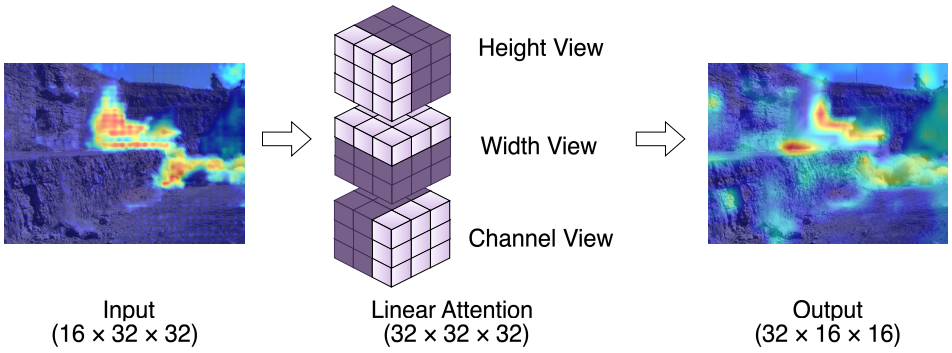

Figure 3. Detailed structure of the Multiview Attention Module (Stage 4). Attention is applied along height, width, and channel dimensions for feature refinement. The lighter-colored cubes indicate the dimension along which linear attention is computed.

Let the input tensor at stage k of the encoder be $\mathbf{F}^k \in \mathbb{R}^{N\times C\times H\times W}$. For the multiview operation, the tensor is split into four equal chunks along the channel dimension:

$$\mathbf{F}_{\text{split},i} = \{\mathbf{F}_1, \mathbf{F}_2, \mathbf{F}_3, \mathbf{F}_4\}, \tag{14}$$

where each chunk $\mathbf{F}_i \in \mathbb{R}^{N\times \frac{C}{4}\times H\times W}$.

Each chunk undergoes a distinct processing operation involving element-wise multiplication with an attention map computed via softmax activation over specific dimensions:

$$\mathbf{F}_1' = \mathbf{F}_1 \quad \text{(Identity Mapping)}, \tag{15}$$

$$\mathbf{F}_2' = \sigma_{\text{spatial}}(\mathbf{F}_2) \odot \mathbf{F}_2, \tag{16}$$

$$\mathbf{F}_3' = \sigma_{\text{height-channel}}(\mathbf{F}_3) \odot \mathbf{F}_3, \tag{17}$$

$$\mathbf{F}_4' = \sigma_{\text{width-channel}}(\mathbf{F}_4) \odot \mathbf{F}_4, \tag{18}$$

where $\sigma_{\text{dimension}}(\cdot)$ denotes the softmax activation applied over the specified dimensions, and $\odot$ represents element-wise multiplication.

The outputs from all four chunks are concatenated along the channel dimension to form the combined feature map:

$$\mathbf{F}' = \text{Concat}\left(\mathbf{F}_1',\ \mathbf{F}_2',\ \mathbf{F}_3',\ \mathbf{F}_4'\right). \tag{19}$$

This concatenated feature map is then processed by a pointwise convolution to integrate the multiscale features into a unified representation:

$$\mathbf{F}_{\text{out}} = \text{Conv}_{1\times1}\left(\mathbf{F}'\right). \tag{20}$$

Following the convolution, layer normalization and GELU activation are applied to $\mathbf{F}_{\text{out}}$. Finally, a 2×2 max pooling operation is performed to reduce the spatial dimensions, resulting in

$$\mathbf{F}_{\text{pooled}} = \text{MaxPool}(\mathbf{F}_{\text{out}},\ 2 \times 2). \tag{21}$$

As shown in Figure 3, the spatial view within the multiview attention mechanism is particularly essential for enhancing spatial feature consistency. By focusing on specific regions within the feature maps, the spatial view ensures robust encoding of intricate smoke shapes, such as those formed by narrow plumes or widespread quarry blast emissions. Similarly, the height-channel and width-channel views capture directional patterns and align encoded features with global contexts, enabling robust feature extraction for both narrow, tall plumes and wide, elongated patterns.

This strategic combination of multiview linear attention ensures that the encoder stages of SmokeNet (4–6) are adept at refining the variability and complexity of smoke patterns, delivering feature representations optimized for segmentation in the decoder.

3.2. Decoder

3.2.1. Decoder with Skip Connections

The decoder stages in SmokeNet (Stages 4–6) progressively reconstruct the spatial resolution of the feature maps using transposed convolutions. To enhance segmentation precision, skip connections are employed to transfer enriched features from the encoder to the corresponding decoder stages. These skip connections combine the encoder output at the current stage with the upsampled output of the lower-stage skip connection using element-wise addition, integrating features from multiple levels to ensure dimensional alignment and retain critical details necessary for accurate smoke segmentation.

Let the output tensor from the encoder at stage k be denoted as $\mathbf{F}^k_{\text{encoder}} \in \mathbb{R}^{N\times C_k\times H_k\times W_k}$, where C_k, H_k, and W_k represent the number of channels, height, and width, respectively, at stage k. The output of the lower-stage skip connection is denoted as

$\mathbf{F}_{\text{skip}}^{k+1} \in \mathbb{R}^{N \times C_{k+1} \times H_{k+1} \times W_{k+1}}$. The skip connection input at stage k is computed by adding the encoder output with the upsampled lower-stage skip connection as follows:

$$\mathbf{F}_{\text{skip}}^{k} = \mathbf{F}_{\text{encoder}}^{k} + \text{Up}\left(\mathbf{F}_{\text{skip}}^{k+1}\right) \in \mathbb{R}^{N \times C_k \times H_k \times W_k}, \tag{22}$$

where $\text{Up}(\cdot)$ represents an upsampling operation (e.g., transposed convolution) that aligns the spatial resolution of the lower-stage skip connection with the current stage.

3.2.2. Decoder Stage Operations

At each decoder stage, the input $\mathbf{F}_{\text{decoder}}^{k}$ is formed by adding the output of the skip connection with the upsampled output of the lower decoder stage. This is mathematically expressed as follows:

$$\mathbf{F}_{\text{decoder}}^{k} = \mathbf{F}_{\text{skip}}^{k} + \text{Up}\left(\mathbf{F}_{\text{decoder}}^{k+1}\right) \in \mathbb{R}^{N \times C_k \times H_k \times W_k}, \tag{23}$$

where $\mathbf{F}_{\text{decoder}}^{k+1} \in \mathbb{R}^{N \times C'_k \times H_{k+1} \times W_{k+1}}$ is the output from the lower decoder stage, and $\text{Up}(\cdot)$ ensures spatial alignment.

Once the decoder input is established, it undergoes a series of transposed convolutions and linear operations to reconstruct the spatial resolution while reducing the channel dimensions:

$$\mathbf{F}_{\text{output}}^{k} = \text{TransposedConv}_{3\times3}\left(\mathbf{F}_{\text{decoder}}^{k}\right) \in \mathbb{R}^{N \times C'_k \times H_k \times W_k}, \tag{24}$$

where C'_k represents the reduced channel dimension at stage k. This transposed convolution operation progressively upsamples the feature maps and diminishes the number of channels, effectively restoring spatial resolution while preserving essential details for accurate segmentation.

In the final stage of the decoder, a single-channel segmentation mask is produced through a transposed convolution followed by a sigmoid activation function:

$$\mathbf{F}_{\text{segmentation}} = \sigma\left(\text{TransposedConv}_{3\times3}\left(\mathbf{F}_{\text{output}}^{1}\right)\right) \in \mathbb{R}^{N \times 1 \times H \times W}, \tag{25}$$

where σ denotes the sigmoid activation function, ensuring that the output values are scaled between 0 and 1, suitable for binary segmentation tasks.

This structured use of skip connections and decoder operations ensures the seamless integration of multilevel features, enabling accurate smoke segmentation with fine spatial detail.

3.3. Loss Function

For training, we use a combined loss function that includes a binary cross-entropy (BCE) component and a Dice loss component, formulated to optimize segmentation accuracy by balancing region overlap and boundary alignment:

$$\mathcal{L}_{\text{combined}} = \alpha \cdot \text{BCE}(y, \hat{y}) + \beta \cdot \text{Dice}(y, \hat{y}), \tag{26}$$

where y is the ground truth, $\hat{y}$ is the predicted mask, and α and β are weights assigned to each loss component to balance their contributions. For the combined loss function, hyperparameters α and β were set empirically to balance the contributions of binary cross-entropy (BCE) and Dice loss components. BCE loss emphasizes pixel-wise accuracy and penalizes misclassification of background pixels, while Dice loss emphasizes the overlap and alignment between predicted and ground truth masks. The chosen hyper-

parameters balance pixel-level accuracy and overall mask coherence, resulting in robust segmentation performance.

The layer-specific loss strategy was introduced to address potential feature misalignment between different network stages, particularly between convolutional and attention-based encoder stages. Applying dedicated supervision at intermediate layers encourages more consistent and stable feature propagation across the network, improving the precision of intermediate feature maps and ultimately the final segmentation.

We further implement a cosine annealing learning rate schedule to modulate the learning rate during training, aiming to facilitate smoother convergence. The learning rate η_t at epoch t is given by the following:

$$\eta_t = \eta_{\min} + \frac{1}{2}(\eta_{\max} - \eta_{\min})\left(1 + \cos\left(\frac{t}{T}\pi\right)\right), \tag{27}$$

where $\eta_{\min}$ and $\eta_{\max}$ are the minimum and maximum learning rates, respectively, and T is the total number of epochs. For our model, we set $\eta_{\max} = 0.001$, $\eta_{\min} = 1 \times 10^{-6}$, and $T = 100$ epochs.

With the loss function, we incorporate layer-wise loss functions that combine the overall loss at different layers. Since the layer loss is the combined loss at different layers, we assigned different weights to each layer's loss to balance their contributions. The layer-wise loss is defined as follows:

$$\mathcal{L}_{\text{layer}} = \sum_{i=1}^{N} \gamma_i \cdot \mathcal{L}_{\text{combined},i}, \tag{28}$$

where $\mathcal{L}_{\text{combined},i}$ represents the combined loss at layer i, and γ_i is the weight assigned to layer i. For the layer-wise loss, different weights were assigned to each layer to balance their contributions, with the values assigned as follows: The progressively decreasing weights ($\gamma_1 = 0.5$, $\gamma_2 = 0.4$, $\gamma_3 = 0.3$, $\gamma_4 = 0.2$, and $\gamma_5 = 0.1$) were empirically determined to ensure that earlier stages—responsible for fundamental spatial and structural features—have relatively higher supervision, while later stages focus more on fine-grained feature refinements.

4. Experiments

The experiments aim to comprehensively evaluate SmokeNet's effectiveness in accurately segmenting smoke in complex scenarios, comparing its performance against established models across diverse datasets, and assessing the balance it strikes between computational efficiency and segmentation accuracy.

4.1. Experimental Setup

4.1.1. Deep Learning Architecture

For Model Selection Rationale, we selected CNNs combined with lightweight linear attention mechanisms to efficiently capture spatial details and contextual dependencies critical for smoke segmentation. While recent advanced architectures (transformers, diffusion models, and Mamba) offer high accuracy, their computational demands often exceed practical limits for onboard, real-time deployment. Thus, our tailored CNN-based approach enhanced by linear attention provides a balanced solution prioritizing accuracy, computational efficiency, and practical usability.

SmokeNet was implemented using the PyTorch 1.9.0 framework and trained on a single NVIDIA Tesla P40 24GB GPU. The model consists of six encoder and decoder layers with filter sizes $\in [4, 8, 16, 32, 64, 128]$. Training was conducted using the AdamW optimizer

with a learning rate of 0.001, a weight decay of 1×10^{-5}, and a cosine annealing learning rate schedule ($\eta_{\min} = 0.00001$, $T_{\max} = 50$ iterations). The model was trained for 100 epochs with a batch size of 8, using a combined loss of cross-entropy and Dice loss with layer-wise losses as the loss function.

4.1.2. Dataset

Four datasets selected to encompass both synthetic and real-world smoke variations, testing its robustness and adaptability across diverse conditions:

- Smoke100k-M [13]: A synthetic dataset comprising 25,000 training images and 15,000 test images, as illustrated in Figure 4a. It features centrally located smoke plumes of fixed size, providing a baseline for accuracy assessment in controlled conditions.
- DS01 [10]: This dataset comprises 70,632 training images and 1000 test images, featuring various smoke sizes and locations, as illustrated in Figure 4b. It challenges the model's ability to adapt to different spatial configurations, enhancing its generalization capabilities.
- Fire Smoke [44]: A real-world dataset with 3826 images, including 3060 training images and 766 test images, as illustrated in Figure 4c. It captures both outdoor wildfire smoke and indoor smoke scenarios, providing realistic environments where smoke detection is critical for early fire warning and safety monitoring.
- Quarry Smoke: An industrial dataset comprising 3703 images, including 2962 training images and 741 test images, as illustrated in Figure 4d–f. It represents dense, irregular smoke plumes mixed with dust and debris from quarry blasts, testing the model's ability to segment smoke in dynamic and high-variability environments.

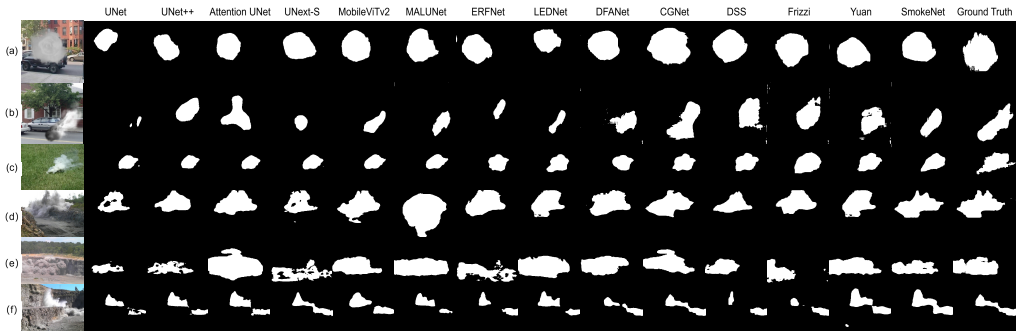

Figure 4. Segmentation results of SmokeNet and comparison models on sample images from four test datasets. (**a**) Smoke100k dataset; (**b**) DS01 dataset; (**c**) Fire Smoke dataset; (**d**–**f**) Quarry Smoke dataset.

4.1.3. Quarry Smoke Dataset Collection

The Quarry Smoke dataset was systematically collected at quarry sites during blasting operations, capturing diverse blast scenarios using multiple camera devices to accommodate varying site conditions. The collected images typically had resolutions including 640×480, 1280×720, 1920×1080, and 3840×2160 pixels, depending on the camera device used. Frame rates were 30 fps for videos with resolutions up to 1080p, and 60 fps for 4K videos, capturing clear temporal dynamics of smoke evolution during quarry blasts. Due to the rapid initial smoke expansion caused by explosive blast energy, we sampled frames at intervals of every 10 frames and selected only frames within approximately the first 10 s post-blast, ensuring noticeable smoke expansion and minimizing redundant data.

Data collection occurred under various environmental and site conditions, including diverse weather scenarios (clear skies, cloudy weather, and occasional rainfall) and quarry-specific characteristics such as rock types, bench face structures, dust levels, rock wall conditions, and blast-induced fragmentation patterns. All data collection procedures were

conducted with appropriate authorization, strictly adhering to relevant ethical guidelines and industry-standard safety regulations.

Annotations were performed using custom annotation tools specifically developed for this project to generate precise binary segmentation masks for smoke regions. The annotation team consisted of professional experts experienced in blast design, explosive loading practices, and quarry operations. Cross-validation was systematically conducted among annotators to ensure annotation consistency, quality, and reliability, reflecting consistent labeling processes.

4.1.4. Data Augmentation

To enhance SmokeNet's robustness and generalization capabilities, we employed a comprehensive data augmentation pipeline that includes both basic and enhanced augmentation techniques.

Basic augmentations, including random horizontal and vertical flips, rotations, and brightness adjustments, were applied uniformly across all datasets to introduce general variability and prevent overfitting. In addition to these common techniques, we implemented enhanced augmentations—synthetic fog and motion blur—to address domain-specific challenges in our datasets.

Synthetic fog better simulates real-world scenarios, such as blast events occurring under overcast skies, rainy showers, and high humidity in mountainous regions. Motion blur is particularly relevant in real-world situations like quarry sites, where hand-held cameras or ground cameras mounted on tripods may experience distortions due to ground vibrations caused by explosive energy. Synthetic fog and motion blur augmentations were implemented using standard libraries from the widely used `Albumentations 1.0.3` [45] data augmentation toolkit. Specifically, we employed the `RandomFog` transform for synthetic fog and the `MotionBlur` transform for motion blur augmentation, both with default parameter ranges provided by the library.

4.1.5. Performance Metrics

SmokeNet's performance was evaluated using four metrics:

- Mean Intersection over Union (mIoU): Quantifies segmentation accuracy by measuring overlap between predicted and ground-truth masks, providing a balanced evaluation that equally penalizes missed segmentation and false alarms.
- Parameter Count: Indicates model scalability. Reported in millions (M).
- Floating Point Operations (FLOPs): Measures computational complexity. Reported in gigaflops (GFLOPs), as indicated in the tables.
- Frames per Second (FPS): Reflects inference speed, critical for computationally constrained applications in dynamic environments like quarry blast monitoring.

Computational complexity (GFLOPs) and inference speed (FPS) were computed using the PyTorch-based profiling library `ptflops 0.6.9`. Specifically, GFLOPs were calculated based on a single forward pass with input resolution 256×256, while FPS was determined by averaging the inference speed over 100 forward passes, excluding initial warm-up iterations.

4.2. Results and Discussion

4.2.1. Results

As shown in Table 1, we evaluated different configurations of our model, investigating the contributions of multiscale convolution, multiview attention, and layer-specific loss functions. Each configuration was trained and evaluated on fixed, predefined training and testing splits of the datasets, with five independent runs conducted for each. The

reported results include the mean and standard deviation, demonstrating the stability and effectiveness of these design choices.

In Table 2, we compared the best-performing configuration of our model with several state-of-the-art segmentation methods. For each method, the same experimental protocol was applied, ensuring consistency in training and evaluation across the standardized dataset splits and five repetitions. This provides a rigorous and fair comparison, reflecting both segmentation accuracy and computational efficiency.

Table 1. Module evaluation across four test datasets (mean ± std).

Model Configuration	mIoU (%)				#Params (M) ↓	GFLOPs ↓	FPS ↑
	Smoke100k	DS01	Fire Smoke	Quarry Smoke			
Baseline	72.40 ± 0.08	70.83 ± 0.06	70.45 ± 0.11	69.12 ± 0.05	0.42	0.24	54.25
+ Multiscale	72.19 ± 0.10	69.78 ± 0.05	67.22 ± 0.07	63.71 ± 0.06	**0.23**	0.08	**128.65**
+ MultiviewAttn	73.81 ± 0.07	71.53 ± 0.12	69.62 ± 0.04	67.74 ± 0.10	0.71	0.12	56.03
+ LayerLoss	70.75 ± 0.09	67.45 ± 0.06	66.16 ± 0.08	63.52 ± 0.07	0.42	0.24	54.25
+ Multiscale + LayerLoss	72.24 ± 0.04	71.41 ± 0.05	68.95 ± 0.08	66.67 ± 0.07	**0.23**	0.08	**128.65**
+ MultiviewAttn + LayerLoss	74.10 ± 0.07	73.14 ± 0.06	72.24 ± 0.05	71.67 ± 0.08	0.71	0.12	56.03
+ Multiscale + MultiviewAttn	75.63 ± 0.05	73.83 ± 0.09	71.22 ± 0.06	70.34 ± 0.07	0.34	**0.07**	77.05
Full Model (SmokeNet)	**76.45 ± 0.10**	**74.43 ± 0.04**	**73.43 ± 0.03**	**72.74 ± 0.06**	0.34	**0.07**	77.05

Note: Baseline uses normal convolution without attention or layer-specific loss; Multiscale: multiscale convolution; MultiviewAttn: multiview linear attention; LayerLoss: layer-specific loss functions. **Bold** values indicate the best-performing results. Arrows indicate desirable directions for each metric.

Table 2. Detailed comparison of various methods for semantic segmentation with mean and standard deviation.

Methods	mIoU (%)				#Params (M) ↓	GFLOPs ↓	FPS ↑
	Smoke100k	DS01	Fire Smoke	Quarry Smoke			
UNet (2015)	66.13 ± 0.10	61.32 ± 0.08	60.14 ± 0.06	57.18 ± 0.05	28.24	35.24	75.58
UNet++ (2018)	69.12 ± 0.09	64.65 ± 0.04	61.77 ± 0.10	58.44 ± 0.06	9.16	10.72	91.25
AttentionUNet (2018)	69.68 ± 0.05	66.59 ± 0.12	64.15 ± 0.07	59.64 ± 0.09	31.55	37.83	46.48
UNeXt-S (2022)	72.25 ± 0.11	71.62 ± 0.07	69.59 ± 0.12	64.54 ± 0.04	0.77	0.08	**202.06**
MobileViTv2 (2022)	71.73 ± 0.10	71.54 ± 0.07	70.23 ± 0.04	69.12 ± 0.11	2.30	0.09	98.84
MALUNet (2022)	71.81 ± 0.05	70.16 ± 0.10	69.42 ± 0.04	67.64 ± 0.07	**0.17**	0.09	87.72
ERFNet (2017)	71.84 ± 0.09	71.38 ± 0.10	66.59 ± 0.06	66.24 ± 0.07	2.06	3.32	61.22
LEDNet (2019)	70.76 ± 0.10	71.63 ± 0.07	70.13 ± 0.08	67.74 ± 0.11	0.91	1.41	60.19
DFANet (2019)	66.91 ± 0.05	63.87 ± 0.10	62.76 ± 0.07	70.21 ± 0.09	2.18	0.44	31.05
CGNet (2020)	75.64 ± 0.07	73.76 ± 0.11	72.04 ± 0.10	71.91 ± 0.08	0.49	0.86	53.53
DSS (2019)	73.25 ± 0.05	72.17 ± 0.04	69.78 ± 0.12	69.81 ± 0.07	30.20	184.90	32.56
Frizzi (2021) [9]	73.44 ± 0.06	71.67 ± 0.09	70.51 ± 0.07	70.40 ± 0.11	20.17	27.90	60.32
Yuan (2023) [10]	75.57 ± 0.07	**74.84 ± 0.06**	71.94 ± 0.10	70.92 ± 0.08	0.88	1.15	68.81
SmokeNet	**76.45 ± 0.10**	74.43 ± 0.04	**73.43 ± 0.03**	**72.74 ± 0.06**	0.34	**0.07**	77.05

Note: Bold values indicate the best-performing results. Arrows indicate desirable directions for each metric.

Figure 4 illustrates qualitative segmentation results across diverse datasets, including synthetic smoke with uniform, circular patterns; real-world fire smoke with irregular and amorphous structures; and quarry blast smoke characterized by dense, complex plumes. These visualizations offer insights into the strengths and limitations of our model in handling varied smoke characteristics, showcasing its adaptability across synthetic and real-world scenarios.

4.2.2. Impact of Architectural Innovations

A module evaluation study was conducted to assess the contribution of SmokeNet's multiscale convolutional layers and multiview linear attention mechanisms to its segmentation performance and computational efficiency across four datasets: Smoke100k, DS01, Fire Smoke, and Quarry Smoke. Various model configurations were systematically compared, as detailed in Table 1. Models incorporating multiscale convolutions consistently outperformed those using normal convolutions; specifically, the full SmokeNet model achieved an mIoU of 76.45% on Smoke100k compared to 72.40% for the baseline configuration. Furthermore, integrating the multiview linear attention mechanism improved performance. Models including + MultiviewAttn, + MultiviewAttn + LayerLoss, + Multiscale + Mul-

tiviewAttn, and the Full Model consistently showed higher mIoU scores. Notably, the Full Model achieved the highest mIoU of 72.74% on the Quarry Smoke dataset. Additionally, incorporating layer-specific loss functions (configurations labeled + LayerLoss, + MultiviewAttn + LayerLoss, + Multiscale + LayerLoss, and the Full Model) consistently enhanced accuracy. Overall, the optimal configuration, which combines multiscale convolutions, multiview linear attention, and layer-specific losses Full Model , demonstrated performance improvements while maintaining efficient computational demands (0.34 M parameters, 0.07 GFLOPs) and high inference speed (77.05 FPS).

4.2.3. Segmentation Performance Comparison

As shown in Table 2, SmokeNet demonstrates consistent performance across multiple datasets. On the Smoke100k dataset, it achieves the highest mIoU of 76.45%, outperforming CGNet (75.64%) and Yuan (75.57%), while achieving higher scores than other models such as UNeXt-S (72.25%) and MobileViTv2 (71.73%). On the Fire Smoke dataset, SmokeNet attains an mIoU of 73.43%, exceeding CGNet's 72.04% and MobileViTv2's 70.23%, and outperforming other models like Frizzi (70.51%) and Yuan (71.94%). Similarly, in the Quarry Smoke dataset, SmokeNet achieves an mIoU of 72.74%, surpassing CGNet (71.91%), Yuan (70.92%), and Frizzi (70.40%). However, on the DS01 dataset, SmokeNet achieves an mIoU of 74.43%, slightly below Yuan's 74.84%. Despite this, it maintains competitiveness by outperforming models such as CGNet (73.76%), Frizzi (71.67%), and UNeXt-S (71.62%). SmokeNet consistently improves over lightweight models such as MALUNet (70.16%) and LEDNet (71.63%) while also surpassing computationally heavier models like AttentionUNet (66.59%) and DSS (72.17%) across most datasets. Although CGNet provides competitive accuracy across several datasets, its computational complexity (0.86 GFLOPs) exceeds SmokeNet's (0.07 GFLOPs), limiting CGNet's applicability for real-time, resource-constrained tasks. Conversely, Yuan et al.'s lightweight model achieves efficiency comparable to SmokeNet but with slightly lower segmentation accuracy, highlighting the typical trade-off between computational demands and performance in lightweight architectures.

In Figure 4, SmokeNet exhibits its ability to delineate complex smoke boundaries, capture fine-grained details, and maintain consistent segmentation across various smoke scenarios, including challenging quarry smoke environments. Traditional models like UNet and UNet++ produce simplified masks that miss intricate contours and fragmented structures. Smoke-segmentation-specific models such as DSS, Frizzi, and Yuan also generate less-precise masks compared to SmokeNet. Lightweight models such as UNeXt-S and MobileViTv2 often overlook subtle edges and fine details. For the quarry smoke case application, which is commonly used for pollutant quantification inside the smoke plume, slightly thicker masks than the ground truth are acceptable compared to missing parts, which could potentially omit noxious chemicals of the smoke, as shown in Figure 4d–f. In contrast, SmokeNet consistently generates segmentation masks aligned with the ground truth, effectively capturing irregular shapes, narrow projections, and diffused edges.

Figure 5 highlights the performance of SmokeNet across different challenging scenarios where SmokeNet does not effectively segment the entire smoke in these images. In Figure 5a, the image contains two horizontal clusters of smoke, with most of them being white and gray under bright sunlight. However, for the orange part on the left side of the bottom smoke cluster, SmokeNet recognizes the dark orange portion of the smoke but misses the faint, translucent orange smoke, which is less observable, especially against the orange soil background, due to the gradual transitions between smoke and background. The reason the model struggles here is that the scenario is not only challenging for the model but also for human naked eyes; the translucent orange smoke blends subtly with the similarly

colored orange soil background, making it hard to distinguish. Even for human observers, verifying faint smoke presence typically requires viewing the smoke progression in video form, observing the dense smoke fading gradually. This limitation might also reflect a drawback of our lightweight model, which may not have sufficient capacity to capture and store subtle, nuanced differences between smoke and similarly colored backgrounds.

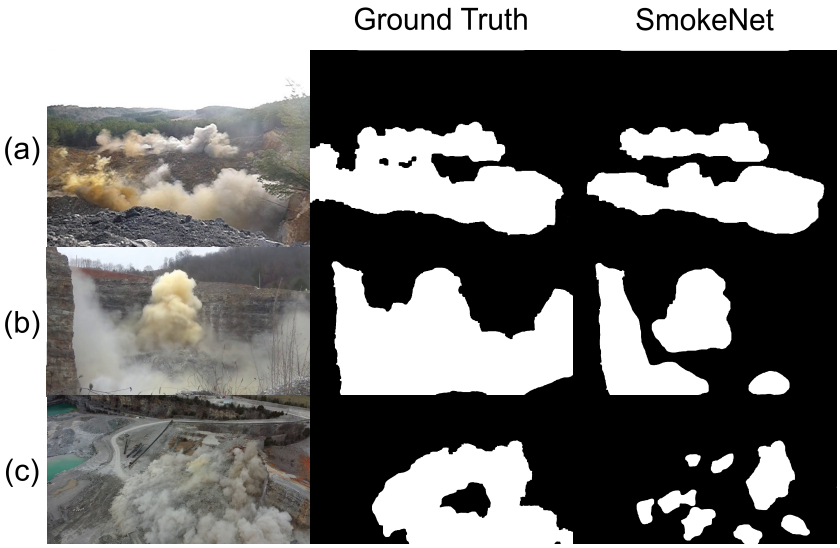

Figure 5. Segmentation results of SmokeNet in challenging quarry scenarios. (**a**) Distant view of grouped smoke plumes with faint regions; (**b**) Close-up scenario with foreground occlusions and diffused smoke boundaries; (**c**) Aerial drone view of smoke plumes horizontally dispersed across the quarry area.

In Figure 5b, the ground camera provides a closer view of the smoke plume at ground level. The complexity of the background textures and the irregularity of the smoke shapes make it difficult for SmokeNet to accurately delineate the two-part smoke composition: the yellowish smoke directly emanating from the collapsed rocks during the blast, and the white smoke surrounding it, which falls to the lower ground level first and then disperses around. Additionally, the presence of dry twigs in front of the smoke further confuses the model in recognizing the entire shape of the smoke, especially when combined with the faint smoke in the center and the rocks observed behind. This challenging situation arises because the two smoke parts result from different smoke dynamics and possibly differing depth characteristics, combined with noise from the thin foreground dried tree branches. Furthermore, since the model has learned smoke features predominantly as solid, grouped masks without internal empty areas, segmenting smoke accurately in such complex cases remains difficult.

In Figure 5c, the smoke image is captured from a drone-mounted camera, which covers the overall view of the smoke plume spread across the entire quarry site. SmokeNet can recognize the smoke near the blast spot and the rock wall but cannot accurately recognize the dispersed smaller grouped smoke clusters due to the low visibility of sparse smoke regions and the intricate gray background surface textures. This image combines challenges from the above two cases: the smoke groups are scattered widely across the image, leaving an uncommon empty central region without smoke, differing significantly from typical training dataset examples. Moreover, from the drone view perspective, smoke mixed with dust tends to exhibit higher similarity to the gray rock terrain, complicating accurate segmentation.

Despite these challenges requiring further improvement, the overall quantitative and qualitative analysis demonstrates that SmokeNet performs better than established

models in segmentation performance across various datasets. The higher mIoU scores and consistent qualitative results highlight SmokeNet's effectiveness in accurately segmenting smoke under diverse conditions. Quantitatively, SmokeNet achieves approximately 1–3% higher mIoU scores across most datasets compared to recent lightweight models (e.g., MALUNet and LEDNet). While these improvements in segmentation accuracy are notable, practical deployment in scenarios such as quarry blast monitoring also critically depends on computational efficiency and real-time inference capabilities. We, therefore, examine these efficiency aspects in greater detail in the next subsection.

4.2.4. Model Efficiency Comparison

Table 2 provides a comprehensive comparison of various semantic segmentation methods based on number of parameters (#Params), computational complexity (GFLOPs), and inference speed (FPS) while also considering their segmentation performance (mIoU).

SmokeNet demonstrates efficiency with only 0.34 M parameters and the lowest computational complexity of 0.07 GFLOPs, achieving an inference speed of 77.05 FPS. This makes SmokeNet one of the most lightweight and computationally efficient models among the compared methods. Additionally, SmokeNet maintains competitive mIoU scores, outperforming other models on Smoke100k, Fire Smoke, and Quarry Smoke datasets.

Traditional models like UNet and UNet++ have higher parameter counts (28.24 M and 9.16 M, respectively) and GFLOPs (35.24 and 10.72), which result in heavy workloads for GPU memory and computing usage. AttentionUNet offers slightly improved mIoU scores but at the cost of increased parameters (31.55 M) and GFLOPs (37.83 GFLOPs), leading to a slower inference speed of 46.48 FPS. Models such as ERFNet and DFANet have lower parameter counts (2.06 M and 2.18 M, respectively) compared to UNet and UNet++. However, they still exhibit relatively high GFLOPs (3.32 and 0.44 GFLOPs) and lower FPS (61.22 and 31.05). Thus, they impose computational demands compared to other lightweight models.

Advanced models such as UNeXt-S and MobileViTv2 achieve higher inference speeds of 202.06 FPS and 98.84 FPS with 0.77M and 2.30M parameters and 0.08 GFLOPs and 0.09 GFLOPs, respectively. However, their mIoU scores are generally lower than those of SmokeNet.

Among parameter-efficient models, MALUNet is the most lightweight with only 0.17M parameters and 0.09 GFLOPs, while CGNet and LEDNet offer a balance between low parameter counts (0.49M and 0.91M) and reasonable GFLOPs (0.86 and 1.41). Nonetheless, SmokeNet outperforms these models in terms of computational efficiency and maintains competitive inference speeds.

Models specifically designed for smoke segmentation, like DSS and Frizzi, require higher computational resources. DSS and Frizzi demand 184.90 GFLOPs and 27.90 GFLOPs, respectively, leading to slower inference speeds of 32.56 FPS and 60.32 FPS. While achieving respectable mIoU scores, their computational demands are considerably higher compared to SmokeNet. Yuan achieves competitive mIoU scores with 0.88 M parameters and 1.15 GFLOPs, but SmokeNet generally offers a better efficiency–performance balance across most datasets, except for the DS01 dataset where Yuan achieves the highest mIoU. Overall, SmokeNet provides a balance between computational demand and segmentation performance, making it highly suitable for real-time smoke segmentation tasks.

5. Conclusions

This study introduces SmokeNet, a lightweight model for smoke plume segmentation across diverse scenarios, including quarry blast smoke. By integrating multiscale convolutions and multiview linear attention within a lightweight framework, SmokeNet

segments dynamic smoke plumes with varying opacity and shapes. The experimental results demonstrate high segmentation accuracy on both synthetic and real-world datasets, such as campfire, wildfire, and quarry blast smoke. Its low parameter count, reduced computational demands, and high inference speed make it support its application in environmental monitoring and industrial safety.

However, the failure case study revealed limitations in detecting sparse smoke regions against complex backgrounds and segmenting smoke with irregular shapes and low-visibility areas. Addressing these challenges will involve enhancing feature extraction techniques, improving background differentiation, utilizing augmented and synthetic datasets for greater robustness, and optimizing SmokeNet's architecture for real-time processing. Exploring dynamic kernel shapes may also improve generalizability for irregular objects like smoke plumes. Overall, SmokeNet offers a balanced trade-off between performance and computational efficiency, making it a valuable tool for real-time smoke detection and monitoring in various applications.

Author Contributions: Conceptualization, X.L. and E.J.I.; methodology, X.L.; validation, X.L.; formal analysis, X.L.; investigation, X.L.; resources, E.J.I.; data curation, X.L.; writing—original draft preparation, X.L.; writing—review and editing, X.L. and E.J.I.; visualization, X.L.; supervision, E.J.I.; project administration, E.J.I.; funding acquisition, E.J.I. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Austin Powder through a contractual agreement.

Data Availability Statement: The dataset supporting the findings of this study was provided by Austin Powder. Due to contractual and confidentiality restrictions, the data are not publicly available; however, they can be made available from the corresponding author on reasonable request.

Acknowledgments: This work was supported by Austin Powder. We gratefully acknowledge Austin Powder for providing the essential dataset that enabled the development and evaluation of our approach.

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

References

1. Oluwoye, I.; Dlugogorski, B.Z.; Gore, J.; Oskierski, H.C.; Altarawneh, M. Atmospheric emission of NO_x from mining explosives: A critical review. *Atmos. Environ.* **2017**, *167*, 81–96. [CrossRef]
2. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015, Proceedings of the 18th International Conference, Munich, Germany, 5–9 October 2015*; Springer International Publishing: New York, NY, USA, 2015; pp. 234–241.
3. Long, J.; Shelhamer, E.; Darrell, T. Fully Convolutional Networks for Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Nashville TN, USA, 11–15 June 2015; pp. 3431–3440.
4. Howard, A.G. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861.
5. Tan, M.; Le, Q. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In Proceedings of the International Conference on Machine Learning (ICML), Long Beach, CA, USA, 9–15 June 2019; pp. 6105–6114.
6. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 6848–6856.
7. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale. In Proceedings of the International Conference on Learning Representations (ICLR), Virtual Event, 3–7 May 2021.
8. Yuan, F.; Zhang, L.; Xia, X.; Wan, B.; Huang, Q.; Li, X. Deep Smoke Segmentation. *Neurocomputing* **2019**, *357*, 248–260. [CrossRef]
9. Frizzi, S.; Bouchouicha, M.; Ginoux, J.M.; Moreau, E.; Sayadi, M. Convolutional Neural Network for Smoke and Fire Semantic Segmentation. *IET Image Process.* **2021**, *15*, 634–647. [CrossRef]
10. Yuan, F.; Li, K.; Wang, C.; Fang, Z. A Lightweight Network for Smoke Semantic Segmentation. *Pattern Recognit.* **2023**, *137*, 109289. [CrossRef]

11. Yin, Z.; Wan, B.; Yuan, F.; Xia, X.; Shi, J. Deep normalized convolutional neural network for fire and smoke detection. *Multimed. Tools Appl.* **2018**, *77*, 28549–28565.
12. Yuan, F.; Zhang, L.; Xia, X.; Wan, B.; Li, Q. Video smoke detection with deep convolutional neural networks based on spatial-temporal feature extraction. *Sensors* **2019**, *19*, 666.
13. Cheng, H.Y.; Yin, J.L.; Chen, B.H.; Yu, Z.M. Smoke100k: A Database for Smoke Detection. In Proceedings of the 2019 IEEE 8th Global Conference on Consumer Electronics (GCCE), Osaka, Japan, 15–18 October 2019; pp. 596–597.
14. Khan, S.; Muhammad, K.; Hussain, T.; Ser, J.D.; Cuzzolin, F.; Bhattacharyya, S.; Akhtar, Z.; de Albuquerque, V.H.C. DeepSmoke: Deep learning model for smoke detection and segmentation in outdoor environments. *Expert Syst. Appl.* **2021**, *182*, 115125. [CrossRef]
15. Yao, L.; Zhao, H.; Peng, J.; Wang, Z.; Zhao, K. FoSp: Focus and Separation Network for Early Smoke Segmentation. *arXiv* **2023**, arXiv:2306.04474. [CrossRef]
16. Marto, T.; Bernardino, A.; Cruz, G. Fire and Smoke Segmentation Using Active Learning Methods. *Remote Sens.* **2023**, *15*, 4136. [CrossRef]
17. Dewangan, A.; Pande, Y.; Braun, H.; Vernon, F.; Perez, I.; Altintas, I.; Cottrell, G.W.; Nguyen, M.H. FIgLib & SmokeyNet: Dataset and Deep Learning Model for Real-Time Wildland Fire Smoke Detection. *arXiv* **2021**, arXiv:2112.08598.
18. Pesonen, J.; Hakala, T.; Karjalainen, V.; Koivumäki, N.; Markelin, L.; Raita-Hakola, A.M.; Honkavaara, E. Detecting Wildfires on UAVs with Real-time Segmentation by Larger Teacher Models. *arXiv* **2024**, arXiv:2408.10843.
19. Lee, e.a. Real-Time Smoke Detection in Surveillance Videos Using an RT-DETR Architecture. *Fire* **2024**, *7*, 387.
20. Zhou, Z.; Rahman Siddiquee, M.M.; Tajbakhsh, N.; Liang, J. UNet++: A Nested U-Net Architecture for Medical Image Segmentation. In Proceedings of the 4th International Workshop on Deep Learning in Medical Image Analysis (DLMIA 2018), Granada, Spain, 20 September 2018; pp. 3–11.
21. Oktay, O. Attention U-Net: Learning Where to Look for the Pancreas. *arXiv* **2018**, arXiv:1804.03999.
22. Cai, Y.; Wang, Y. MA-Unet: An improved version of U-Net based on multi-scale and attention mechanism for medical image segmentation. *arXiv* **2020**, arXiv:2012.10952.
23. Chen, J.; Lu, Y.; Yu, Q.; Luo, X.; Adeli, E.; Wang, Y.; Lu, L.; Yuille, A.L.; Zhou, Y. TransUNet: Transformers Make Strong Encoders for Medical Image Segmentation. *arXiv* **2021**, arXiv:2102.04306.
24. Cao, H.; Wang, Y.; Chen, J.; Jiang, D.; Zhang, X.; Tian, Q.; Wang, M. Swin-Unet: Unet-Like Pure Transformer for Medical Image Segmentation. In Proceedings of the ECCV 2022 Workshops, Tel Aviv, Israel, 23–27 October 2022; Volume 13803, pp. 205–218.
25. Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; Jégou, H. Training Data-Efficient Image Transformers & Distillation Through Attention. In Proceedings of the 38th International Conference on Machine Learning (ICML), Virtual, 18–24 July 2021; pp. 10347–10357.
26. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 11–17 October 2021; pp. 10012–10022.
27. Wang, W.; Xie, E.; Li, X.; Fan, D.P.; Song, K.; Liang, D.; Lu, T.; Luo, P.; Shao, L. Pyramid Vision Transformer: A Versatile Backbone for Dense Prediction without Convolutions. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 11–17 October 2021; pp. 568–578.
28. Xie, E.; Wang, W.; Yu, Z.; Anandkumar, A.; Alvarez, J.M.; Luo, P. SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 12077–12090.
29. Rajamani, K.T.; Rani, P.; Siebert, H.; ElagiriRamalingam, R.; Heinrich, M.P. Attention-Augmented U-Net (AA-U-Net) for Semantic Segmentation. *Signal Image Video Process.* **2023**, *17*, 981–989. [CrossRef]
30. Yan, H.; Zhang, C.; Wu, M. Lawin Transformer: Improving Semantic Segmentation Transformer with Multi-Scale Representations via Large Window Attention. *arXiv* **2022**, arXiv:2201.01615.
31. Yang, J.; Li, C.; Dai, X.; Gao, J. Focal Self-attention for Local-Global Interactions in Vision Transformers. *arXiv* **2021**, arXiv:2107.00641.
32. Rahman, M.M.; Marculescu, R. MERIT: Multi-scale Hierarchical Vision Transformer with Cascaded Attention Decoding for Medical Image Segmentation. In Proceedings of the Medical Imaging with Deep Learning (MIDL), Paris, France, 3–5 July 2024.
33. Chu, X.; Tian, Z.; Wang, Y.; Zhang, B.; Ren, H.; Wei, X.; Xia, H.; Shen, C. Twins: Revisiting the Design of Spatial Attention in Vision Transformers. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 9355–9366.
34. Poudel, R.P.; Liwicki, S.; Cipolla, R. Fast-SCNN: Fast semantic segmentation network. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 1308–1314.
35. Wang, Y.; Zhou, Q.; Liu, J.; Xiong, J.; Gao, G.; Wu, X.; Latecki, L.J. LEDNet: A Lightweight Encoder-Decoder Network for Real-Time Semantic Segmentation. In Proceedings of the 2019 IEEE International Conference on Image Processing (ICIP), Taipei, Taiwan, 22–25 September 2019; pp. 1860–1864.

36. Yu, C.; Wang, J.; Peng, C.; Gao, C.; Yu, G.; Sang, N. BiSeNet: Bilateral segmentation network for real-time semantic segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 325–341.
37. Yu, C.; Gao, C.; Wang, J.; Yu, G.; Shen, C.; Sang, N. BiSeNet V2: Bilateral Network with Guided Aggregation for Real-time Semantic Segmentation. *arXiv* **2020**, arXiv:2004.02147. [CrossRef]
38. Cheng, H.; Zhang, Y.; Xu, H.; Li, D.; Zhong, Z.; Zhao, Y.; Yan, Z. MSGU-Net: A lightweight multi-scale Ghost U-Net for image segmentation. *Front. Neurorobotics* **2025**, *18*, 1480055. [CrossRef]
39. Sawant, S.S.; Medgyesy, A.; Raghunandan, S.; Götz, T. LMSC-UNet: A Lightweight U-Net with Modified Skip Connections for Semantic Segmentation. In Proceedings of the 17th International Conference on Agents and Artificial Intelligence, Porto, Portugal, 23–25 February 2025; pp. 726–734. [CrossRef]
40. Rosas-Arias, L.; Benitez-Garcia, G.; Portillo-Portillo, J.; Sánchez-Pérez, G.; Yanai, K. Fast and Accurate Real-Time Semantic Segmentation with Dilated Asymmetric Convolutions. In Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 10–15 January 2021.
41. Wu, B.; Xiong, X.; Wang, Y. Real-Time Semantic Segmentation Algorithm for Street Scenes Based on Attention Mechanism and Feature Fusion. *Electronics* **2024**, *13*, 3699. [CrossRef]
42. Guo, Z.; Ma, D.; Luo, X. Lightweight Semantic Segmentation Algorithm Integrating Coordinate and ECA-Net Modules. *Optoelectron. Lett.* **2024**, *20*, 568. [CrossRef]
43. Kwon, Y.; Kim, W.; Kim, H. HARD: Hardware-Aware Lightweight Real-Time Semantic Segmentation Model Deployable from Edge to GPU. In Proceedings of the 17th Asian Conference on Computer Vision (ACCV), Hanoi, Vietnam, 8–12 December 2024; pp. 252–269. [CrossRef]
44. Kaabi, R.; Bouchouicha, M.; Mouelhi, A.; Sayadi, M.; Moreau, E. An Efficient Smoke Detection Algorithm Based on Deep Belief Network Classifier using Energy and Intensity Features. *Electronics* **2020**, *9*, 1390. [CrossRef]
45. Buslaev, A.; Iglovikov, V.I.; Khvedchenya, E.; Parinov, A.; Druzhinin, M.; Kalinin, A.A. Albumentations: Fast and flexible image augmentations. *Information* **2020**, *11*, 125. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.