engineering proceedings

MDPI

*Proceeding Paper*

# Forest Fire Monitoring from Unmanned Aerial Vehicles Using Deep Learning †

**Christophe Graveline [1] and Pierre Payeur [2]**

[1] University of Ottawa; cgrav064@uottawa.ca
[2] University of Ottawa; ppayeur@uottawa.ca
* Correspondence:
† Presented at the 12th International Electronic Conference on Sensors and Applications (ECSA-12), 12–14 November 2025; Available online: https://sciforum.net/event/ECSA-12.

**Abstract**

Forest fires pose a serious threat to the environment with the potential of causing ecological harm, financial losses, and human casualties. While research suggests that climate change will increase the frequency and severity of these fires, recent developments in deep learning and convolutional neural networks (CNN) have greatly enhanced fire detection techniques and capability. These models can be leveraged by unmanned aerial vehicles (UAVs) to automatically monitor burning areas. However, drones can carry only limited computational and power resources, therefore on-board computing capabilities are constrained by hardware limitations. This work focuses on the design of segmentation models to identify and localize active burning areas from aerial RGB images processed on limited computing resources. To achieve this goal, the research compares the performance of different variants of the DeepLabv3 neural network model for fire segmentation when trained and tested with the FLAME dataset using a k-fold cross validation approach. Experimental results are compared with U-Net, a benchmark model used with the FLAME dataset, by implementing this model in the same codebase as the DeepLabv3 model. This work demonstrates that a refined version of DeepLabv3, with a MobileNetv2 backbone using pretrained layers and a simplified atrous spatial pyramid pooling (ASPP) module, yields a similar performance to U-Net with a precision of 87.8% and a recall of 83.2% while only requiring 20% of the number of parameters involved with the U-Net topology. This significantly reduces memory and power consumption, enabling longer UAV flight duration and reducing the processing overhead associated with sensor input, making it more suitable for deployment on unmanned aerial vehicles. The model's compact architecture implemented using TensorFlow and Keras for model design and training, along with OpenCV for image preprocessing, makes it portable and easy to integrate with edge devices such as NVIDIA Jetson boards.

**Keywords:** image segmentation; aerial image processing; deep learning; forest fire detection

Academic Editor(s): Name

Published: date

**Citation:** Graveline, C.; Payeur, P. Forest Fire Monitoring from Unmanned Aerial Vehicles Using Deep Learning. *Eng. Proc.* **2025**, *5*, x. https://doi.org/10.3390/xxxxx



## 1. Introduction

Forest fires pose a serious threat to the environment with the potential of causing harm to biodiversity, soil erosion, and air pollution, as well as result in human casualties [1]. In recent years, climate change has made the issue worse with research suggesting

that forest fires will increase in frequency and severity [2]. They can also have a negative impact on the economy by endangering local businesses, tourism, and agriculture, leading to financial losses. According to the National Interagency Fire Center (NIFC), forest fires have burned on average 4,287,522 acres annually over the past ten years [3], and annually costing $3 billion to fight in the US [4]. Similarly, the Canadian National Forestry Database (NFD) annually reports over 8000 fires, burning on average over 2.1 million hectares [5]. According to Canadian wildland fire management agencies, these fires cost between $800 million and $1.4 billion annually, and the impact of climate change is expected to drive up costs [6]. Forest fires not only pose a threat to those who are caught in the burning area. Because they release a lot of smoke they can lead to respiratory illnesses and long-term health issues to nearby communities. Moreover, forest fires burn massive amounts of biomass and release significant volumes of carbon monoxide and carbon dioxide into the atmosphere, further exacerbating the effects of climate change.

For these reasons researchers have created early detection techniques to better control forest fires. Fire detection has historically depended on human observation from lookout towers, which is subject to human error and limits coverage. Another approach is the use of electronic sensors, which have response delays because they need a high concentration of heat or smoke to sound an alarm [7]. Likewise, satellites are used to cover large areas, but they also need human monitoring and are prone to data latency. Recent developments in deep learning and artificial intelligence (AI) have greatly enhanced fire detection techniques. Convolutional neural networks (CNNs), a type of AI-based computer vision technique, has had great success in detecting forest fires early on. There are four fundamental approaches to applying computer vision techniques to fire detection. Those include classification, object detection, semantic segmentation, and instance segmentation. Image classification aims to find out if an image's content falls into a particular class. In this paper, a classifier model's function is to determine whether a given image contains fire or not. Object detection aims to not only find if an image belongs to a particular category but also to locate the burning area using a bounding box. Finally, there is segmentation, which evolves tracing a pixel-level outline of an object known as a mask. There are two types of segmentation: instance segmentation, which distinguish between different instances of an object, and semantic segmentation, which is used in this paper to locate and identify fire regions at the pixel level. However, semantic segmentation cannot differentiate between separate instances of fire regions.

This paper explores the development and application of CNN-based methods for forest fire monitoring. It begins by reviewing the state-of-the-art in the field of forest fire monitoring, including a survey of available datasets and techniques used for fire monitoring, while highlighting their advantages and limitations. Next, we discuss the constraints that come with deploying these models on an unmanned aerial vehicle (UAV). Finally, we propose and integrate our own deep learning-based approach to fire monitoring. The design of this deep learning model explicitly considers that it is intended for implementation on an embedded system on board an unmanned aerial vehicle with limited computational and power resources.

## 2. State of the Art

### *2.1. Image-Based Fire Detection*

This section surveys the literature on computer vision-based forest fire monitoring. There has been extensive research conducted on the subject, and many researchers have implemented and tested models for tasks such as fire detection, classification, and segmentation. Among detection models considered in the literature, YOLO (You Only Look Once) [8] has emerged as a popular choice due to its speed and accuracy. Jiao et al.

employed YOLOv3 for wildfire detection [9]. They proposed a lightweight variant to YOLOv3, called Tiny-YOLOv3, which is able to process more frames per second, making it a good choice for computationally restricted environments such as UAVs. Their model was tested on 60 images but they did not specify the dataset utilized. They reported an 82% precision and 79% recall on their testing set. Recent studies explored newer versions of YOLO. Examples include Tahir et al. [10] who implemented YOLOv5 for fire detection. They used the FLAME and FireNet datasets to train and test their model. They also proposed a method to reduce the computational cost of their model by integrating CSPNet [11] and Darknet [12] into their base YOLOv5 model. Their model resulted in precision and recall scores of 97% and 92% respectively, and an F1 score of 94%. Li et al. [13] introduced a fire recognition model based on ShuffleNetv2 called R-ShuffleNetv2 which they train and evaluate on the FLAME dataset. Their findings indicate that R-ShuffleNetv2 performed better than ShuffleNetv2, achieving a processing rate of 31 frames per second while maintaining an F1 score of 89.09%. Other methods, worth noting even though they are not strictly used for fire detection, include that of Chiang et al. [14]. They developed a method for dead tree detection, which is crucial in preventing forest fires. Their approach used a Mask R-CNN [15] model with transfer learning. A notable element of their approach is that they used data augmentation to expand their dataset. This approach achieved an average precision score of 54% in detecting dead trees from aerial images. Sridar et al. [16] employed DenseNet [17] for fire detection. They included images without fire to reduce false positives. Their model demonstrated 90% accuracy in classifying images containing forest fires. Alternatively, segmentation-based models remain relatively underexplored in comparison with forest fire detection. This highlights the need for further research in segmentation-based models. In this category, the authors of the FLAME dataset [18] propose the use of U-Net [19] for fire segmentation with a precision of 92% and recall of 84%. A summary of the different models in the literature is presented in Table 1.

**Table 1.** Summary of different models in the literature for forest fire detection or segmentation including the dataset that they were tested on and a summary of their performance.

| Model Used | Dataset | Performance |
|---|---|---|
| Tiny-YOLOv3 [9] | Unspecified | Precision: 82%, Recall: 79% |
| YOLOv5 [10] | FLAME + FireNet | Precision: 97%, Recall: 92%, F1 Score 94% |
| ShuffleNetv2 [13] | FLAME | Acc: 82.12%, F1 Score: 85.44%, 34FPS |
| R-ShuffleNetv2 [13] | FLAME | Acc: 86.33%, F1 Score: 89.08%, 31FPS |
| U-Net [18] | FLAME | Precision: 92%, Recall: 84%. |
| DenseNet [16] | Custom | Acc: 90% |

### *2.2. Unmanned Aerial Vehicles*

Unmanned aerial vehicles (UAV) or drones became popular for the monitoring of forest fires because of their ability to swiftly navigate large and dense areas without a human pilot involved, which reduces risk to human lives and deployment cost. Many different sensors can be mounted on UAVs including RGB cameras, thermal cameras, and gas sensors. UAVs are also capable of processing their surroundings in real time. One method used to process the data captured by the UAV is by relaying images to a ground station and processing the data there. However, the UAV must be connected to a broadband network, which might not be available when working in remote areas. Therefore, a commonly used alternative has been to execute image processing with onboard edge

computing and relaying only the location of detected burning areas. It is important to note that UAVs have limited computational and power resources given their size and reliance on batteries. This must be considered when designing new computational models. Another critical aspect is the type of data used. UAVs can be equipped with RGB cameras and thermal cameras, and both can be used for fire detection. However, in this work we opted to eliminate thermal images because in real life scenarios heat-emitting objects that are not fires may be present and can be perceived as false positives. Moreover, despite the fact that fusing RGB and thermal images is likely to improve accuracy, processing more complex input image data is taxing the requirements for onboard hardware and energy consumption [1].

## 3. Technical Background

The detection of forest fires leverages techniques from machine learning, most notably artificial neural networks. Although artificial neural networks come in a variety of classes, the main ones used in computer vision are convolutional neural networks (CNN) and fully connected neural networks. This section summarizes two neural model architectures explored in this research.

### *3.1. U-Net Architecture*

U-Net, a segmentation-based model, was originally proposed for medical imaging applications in [19] but has since been applied in other domains. U-Net is characterized by its U-shaped architecture, which consists of an encoder and a decoder, as shown in Figure 1.

**Figure 1.** U-Net architecture.

By feeding the data through a sequence of down-sampling steps, each comprising two convolutional layers, activation layers, and max pooling layers, the encoder is able to capture context by progressively decreasing the spatial dimensions while increasing the feature map depth. The decoder then uses the features extracted by the encoder to create a binary mask with the same resolution as the input image. It does so using a number of convolutional and upsampling layers to increase dimension and decrease the feature map depth. U-Net also includes skip connections that connect corresponding encoder and decoder layers. These skip connections help the model preserve the spatial information lost during feature extraction and increase the model's localization accuracy. U-Net's capacity to integrate fine-grained feature extraction with global context is what makes it a viable model for forest fire detection. The model also converges quickly, is lightweight, and is simple to adapt to meet the limited processing capability requirements.

### *3.2. DeepLabv3*

DeepLabv3 is a deep learning model for semantic segmentation [20] which proved successful in domains like autonomous driving, medical imagery analysis, and notably in aerial image analysis. For this reason, DeepLabv3 offers a competitive alternative for wildfire detection. A representation of DeepLabv3 is presented in Figure 2.

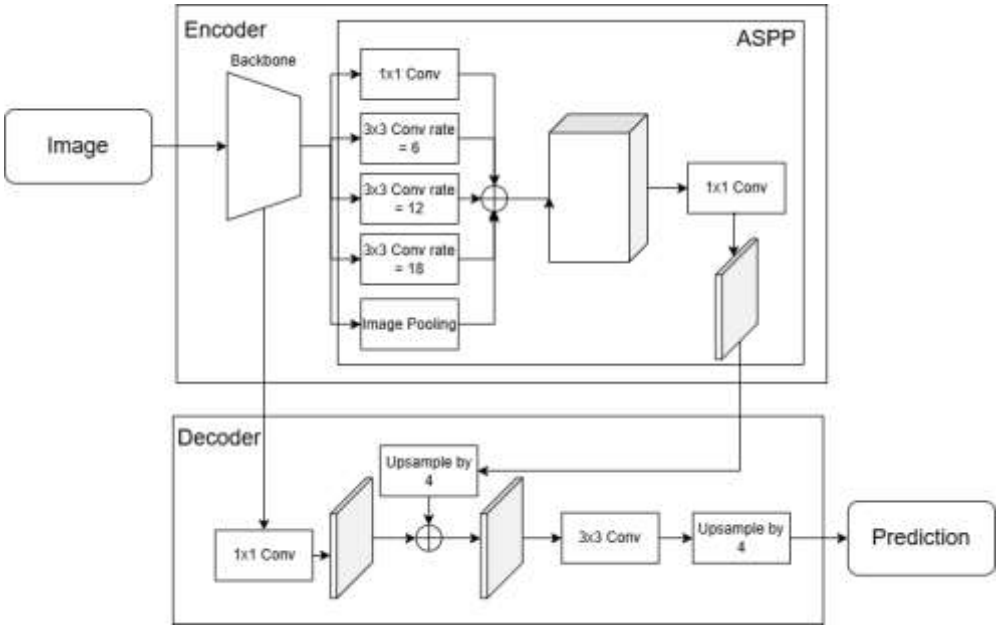



**Figure 2.** DeepLabv3 architecture.

One of the primary innovations of DeepLabv3's is the use of atrous (or dilated) convolutions. Atrous convolution applies convolutional filters at different stride rates and enables the model to contextualize information by capturing features at multiple scales, improving its ability to recognize objects of different sizes. Given the variety of shapes and sizes that fire can take, this is important for the application considered. The Atrous Spatial Pyramid Pooling (ASPP) is a module constructed by combining these atrous convolutions (each at different rate) to extract features at various scales. The model is able to extract both global context and fine details thanks to this multiscale approach. Additionally, DeepLabv3 incorporates residual connections to improve performance and stability and a backbone that is utilized to extract features on the encoder path. In this paper, we explore the use of ResNet and MobileNet as backbones. In the case of ResNet, which is a residual neural network [21] introduced to address the vanishing gradient problem, the version of ResNet50 is selected as it allows a fair compromise between size and feature extraction capability. Alternatively, MobileNet forms a family of lightweight deep convolutional neural networks [22] designed specifically for embedded computer vision applications such as computing hardware available on UAVs. The MobileNetV2 version is selected as it provides computational efficiency without compromising accuracy.

## 4. Datasets

The FLAME dataset (Fire Luminosity Airborne-based Machine learning Evaluation) [18] contains color and thermal images of burning debris in a pine forest in Observatory Mesa, Arizona. Figure 3 shows samples of color images in the top three rows and samples of thermal images in the bottom three rows.

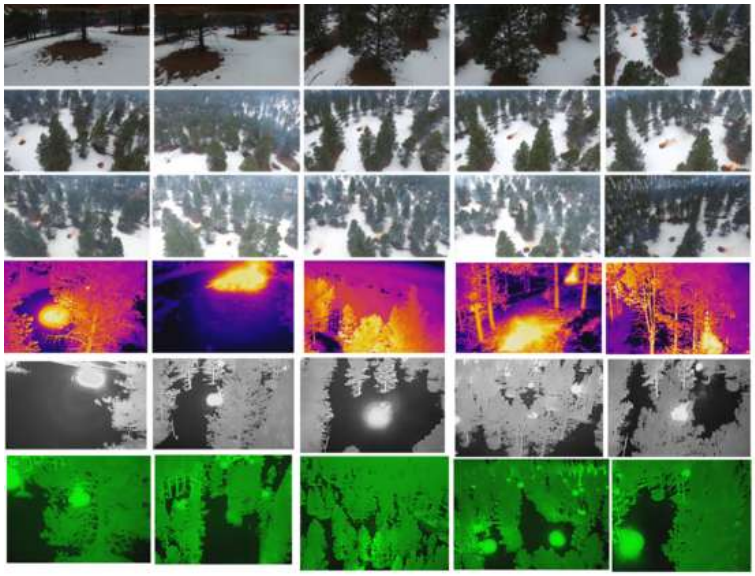


**Figure 3. S**ample images from the FLAME dataset [18] with color images in top three rows, and thermal images in bottom three rows, encoded in Fusion, WhiteHot and GreenHot palettes.

No correspondence is provided between the images captured by the RGB camera and the thermal camera. The data captured by the RGB image was compiled and manually labeled using MATLAB's Image Labeler for classification and segmentation. There are 2003 RGB images captured for segmentation at a resolution of 3480 × 2160, and 47,992 images labeled for classification at a resolution of 254 × 254. The label for segmentation consists of a mask indicating whether each pixel contains fire or not.

In this project the FLAME dataset is used because it provides segmentation labels that are particularly useful for applications aimed at locating fire regions. In contrast with the detection approach, datasets such as FireNet [23] and the Fire Detection Dataset [24,25], segmentation provides a more detailed understanding of the spread of forest fires, and enables more detailed monitoring of their progression over time.

## 5. Methodology

### *5.1. Data Preparation and Computing Resources*

This work focuses on developing segmentation models to identify and localize active burning areas from aerial color images. To achieve this, the FLAME dataset is considered. Labels are converted from their original three channel format to a single channel (gray-scale) image where pixels with a value of one indicate regions with fire, and pixels with a value of zero indicate regions without fire. A sample of converted image from the dataset with corresponding binary mask label is shown in Figure 4. The conversion is performed to simplify the output of the model to a single channel, making it compatible with the binary cross-entropy loss function considered.

**Figure 4.** Sample of the dataset with binary label: input RGB image (**left**) and binary mask (**right**).

To reduce the computational load and to ensure compatibility with the models input size, images were resized from their original 1280 × 720 pixels resolution to 256 × 256 pixels. This reduction was necessary not to exceed the computational resources imposed by hardware limitations. As a result, a compromise is made on the resolution compared to the FLAME dataset paper [18] which originally proposed the implementation of U-Net using 512 × 512 input images. Experiments are conducted on a Google Colab notebook connected to a hosted runtime (shown in Figure 5). This runtime consists of an NVIDIA T4 Tensor Core GPU with 15 GB of virtual RAM and 51 GB of system RAM. The dataset is stored in Google Drive and imported into the notebook. TensorFlow [26] and Keras [27] are used to implement the semantic segmentation models, and OpenCV [28] for image preprocessing, including resizing and Scikit-Learn (sklearn) library to implement k-fold cross-validation [29].

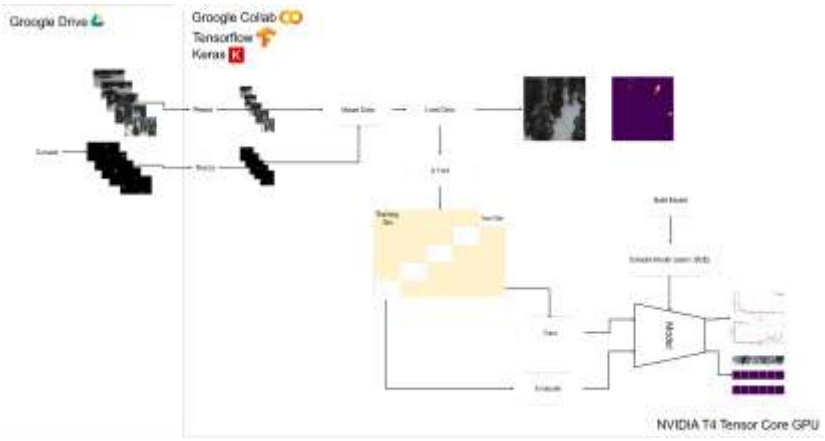



**Figure 5.** End-to-end workflow for semantic segmentation.

Experiments with the converted FLAME dataset are conducted and performance is evaluated with three segmentation models: U-Net, DeepLabv3 with ResNet50 as backbone, and DeepLabv3 with MobileNetv2 as backbone. For both U-Net and DeepLabv3, we used Adam optimizer [30] with a learning rate of 0.001 for fast and stable convergence. We used the binary cross-entropy loss function. We trained our models using a batch size of 16 for up to 30 epochs. Early stopping (with a patience of 5 epochs) is implemented to prevent overfitting.

### *5.2. Fire Detection with U-Net*

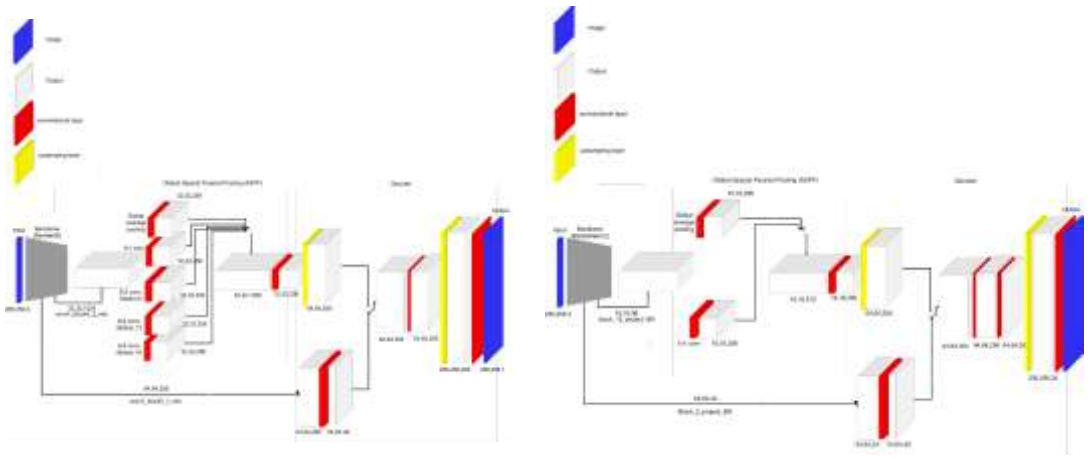
U-Net was implemented in [18] with a couple of modifications from its original design in [19]. To improve performance in small fire regions, Shamsoshoara et al. [18] replaced the standard ReLU activation function with the Exponential Linear Unit (ELU) [31], which mitigates vanishing gradients and speeds up training. Dropout layers were applied to prevent overfitting, and a sigmoid activation function was used in the final layer for binary classification. We also modified the input layer of the network which was adapted to accept 256 × 256 × 3 RGB images.

### *5.3. Fire Detection with DeepLabv3*

Alternative models are built on DeepLabv3, which is selected due to its robust semantic segmentation capabilities and ability to maintain segmentation accuracy over variable fire surface areas. The performance achieved with two different backbones, respectively ResNet50 and MobileNetV2, is compared. ResNet50 is considered for its deep feature extraction capabilities and its mechanisms to avoid vanishing gradients and increase stability, which could be an issue in these experiments because of the limited size of the

dataset. And MobileNetV2 for its lightweight design and improved computational efficiency.

In the case of DeepLabv3 with MobileNetV2 as a backbone, a unique approach is proposed to further reduce the number of parameters. It consists of simplifying the atrous spatial pyramid pooling by using fewer atrous convolutions. This is motivated by the fact that the dimension of the feature map extracted from MobileNetV2 is smaller than that extracted from RestNet50, meaning that the features do not need to be extracted at large dilation. As a result, with the proposed modification the number of parameters in the segmentation model can be reduced from a total of 933,154 parameters (3.56 MB) to a total of 401,701 parameters (1.53 MB), while maintaining similar accuracy. This can be seen by comparing the ASPP in Figures 6a,b where the ResNet50 example contains five atrous convolution layers, including global average pooling, 1 × 1 convolution layer, and three 3 × 3 convolution layers with a dilation of 6, 12 and 18, while its MobileNetV2 counterpart only contains two atrous convolution layers including global average pooling and 1 × 1 convolution layer. Next, the features extracted by the ASPP are up-sampled and fused with lower-level features from the backbone to preserve spatial details. Further convolutional layers upscale the segmentation map, the final output is generated by 1 × 1 convolution layer, producing a pixel-wise prediction of fire regions as seen in Figure 6. Unlike ResNet50 and U-Net, MobileNetV2 does not contain any mechanism known to help mitigate the vanishing gradient problem and stabilize training. Possible solutions include the introduction of skip connections, but to not increase the complexity of the model, we instead freeze a portion of the layers of MobileNetV2 which were initialized with pretrained ImageNet weight [32]. This allows it to leverage existing features from ImageNet, stabilize training and facilitate backpropagation [33].



(**a**) (**b**)

**Figure 6.** DeepLabv3 architecture with two different backbones. The dimensions of the layers are indicated along their respective feature maps. (**a**) The model is composed of a ResNet50 backbone, a dilated spatial pyramid pooling composed of five atrous convolution layers and a decoder [34]. (**b**) The model is composed of a MobileNetV2 backbone, a simplified dilated spatial pyramid pooling with only two atrous convolution layers and a decoder. The model is based on [35].

### *5.4. Performance Evaluation*

To ensure reliable performance assessment, we used k-fold cross-validation instead of the FLAME dataset paper's single 85/15% train/testing partition [18]. We applied a 5-fold split, using 80% of the data for training and 20% for testing in each fold. For each fold we evaluated metrics including intersection over union (IoU) for segmentation accuracy,

as well as precision, recall, specificity and F1-score for fire detection effectiveness. This methodology offers a systematic evaluation of fire segmentation models using the FLAME dataset. By comparing U-Net and DeepLabv3 (with two different backbones) in the same testing environment, we mitigated the issue related to the variability introduced when models are evaluated across different codebases.

## 6. Experimental Evaluation

All three models demonstrated strong performance on the FLAME dataset. A summary of the experimental results with each of the segmentation models considered for comparison is presented in Table 2, with DeepLabv3 using a ResNet50 backbone achieving the highest IoU with a score of 60.59%. This indicates that this model better localizes and accurately captures the shape of the fire regions. This can be observed by comparing the mask prediction generated by DeepLabv3 using ResNet50 in Figure 7 with those produced by U-Net (Figure 8) and DeepLabv3 using MobileNetV2 (Figure 9). Figure 7 presents a more detailed mask segmentation that matches the ground truth displayed on the middle row for each figure. The model's precision score is 88.4%, meaning that the fire regions predicted are likely to be accurate, but a much lower recall of 64.63% indicates that the model struggles to detect all the fire areas present in the ground truth. The high IoU is likely due to the model's atrous convolutions and ASPP module allowing it to better capture multi-scale contextual information, leading to more precise fire boundary delineation, while the relatively weak F1-Score is likely due to the limited training data combined with how large the model is (11,852,353 parameters), leading to some underfitting. Figure 10 illustrates the loss curve of DeepLabv3 with ResNet50 which shows that the testing loss curve is relatively unstable. This could result from poor fitting due to a small dataset size, high variance in the dataset, or a non-representative test split.

In contrast, U-Net achieved the highest F1 score of 90.84%, outperforming both DeepLabv3 variants, which suggests that it performs well both in positive and negative cases. Its precision score was particularly strong compared to other models with a score of 91.83%, indicating a reliable detection of fire regions. And a similarly strong recall score of 90.13% indicates that the model was able to detect the majority of fire regions. This makes U-Net an all-around robust model. However, the model achieved a mean IoU score of 49.71%, indicating some difficulty for the model to accurately localize and size the fire region compared to DeepLabv3 with a ResNet50 backbone. Similar to that of DeepLabv3 with ResNet50, the loss curve of U-Net, shown in Figure 11, exhibits signs of instability in the early training phase.

The MobileNetV2 variant of DeepLabv3 is far more computationally efficient than both its ResNet50 counterpart and U-Net. In terms of computational efficiency (see Table 2), DeepLabv3 with MobileNetV2 required only 20% of the memory and number of parameters compared to U-Net, and 3% of the memory footprint and number of parameters compared with the ResNet50 variant, making it a viable choice for real-time or resource constrained environments. However, the original implementation of DeepLabv3 with MobileNetV2 backbone without frozen layers and initialization on ImageNet weights (detailed in Section 5.3) provided mitigated results due to the effect of vanishing gradient, with a precision and recall of 63.4% and 92.6% respectively, and unstable training loss (see Figure 12). This is likely caused by the bottom layers' weights being poorly fitted and initialized. As a result, the noise introduced in the early layers propagates through the network and is interpreted by top layers as a fire region. This effect can be visualized in Figure 13, where we see multiple false-positive regions.

**Table 2.** Experimental results for compared models including the number of parameters/size of the model. Values provided represent the average over 5-fold cross-validation.

| Model | Precision (%) | Recall (%) | F1-Score (%) | IoU (%) | Param. |
|---|---|---|---|---|---|
| U-Net | 91.83 | 90.13 | 90.84 | 49.71 | 1,941,105 (7.40 MB) |
| DeepLabv3 w/ResNet50 | 88.4 | 64.63 | 74.83 | 60.59 | 11,852,353 (45.21 MB) |
| DeepLabv3 w/MobileNetV2 (frozen) | 87.84 | 83.22 | 84.85 | 49.71 | 401,701 (1.53 MB) (trainable: 317,301) (untrainable: 84,400) |
| DeepLabv3 w/MobileNetV2 (unfrozen) | 63.38 | 92.61 | 76.24 | 49.71 | 401,698 (1.53 MB) (trainable: 391,282) (untrainable: 10,416) |

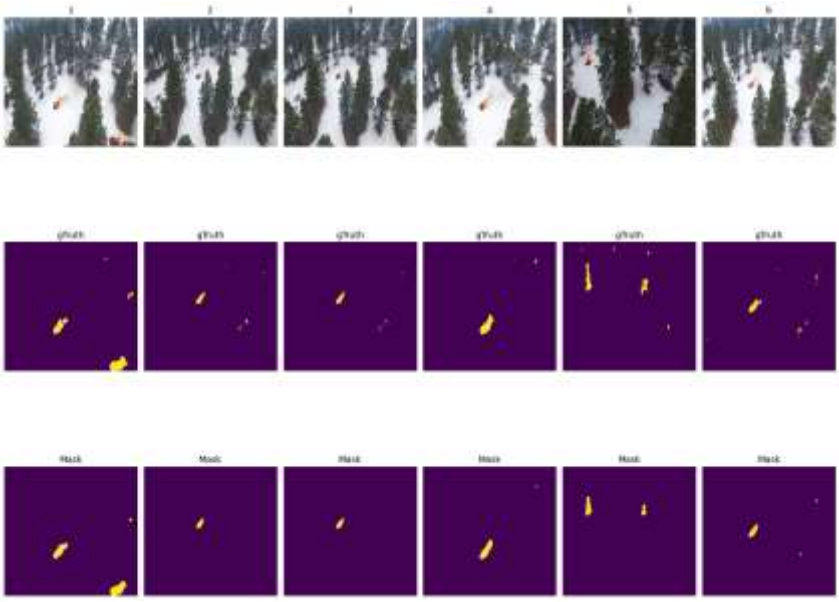


**Figure 7.** Results with DeepLabv3 using ResNet50: (**top row**) input image; (**middle row**) ground truth; (**bottom row**) generated mask segmentation.

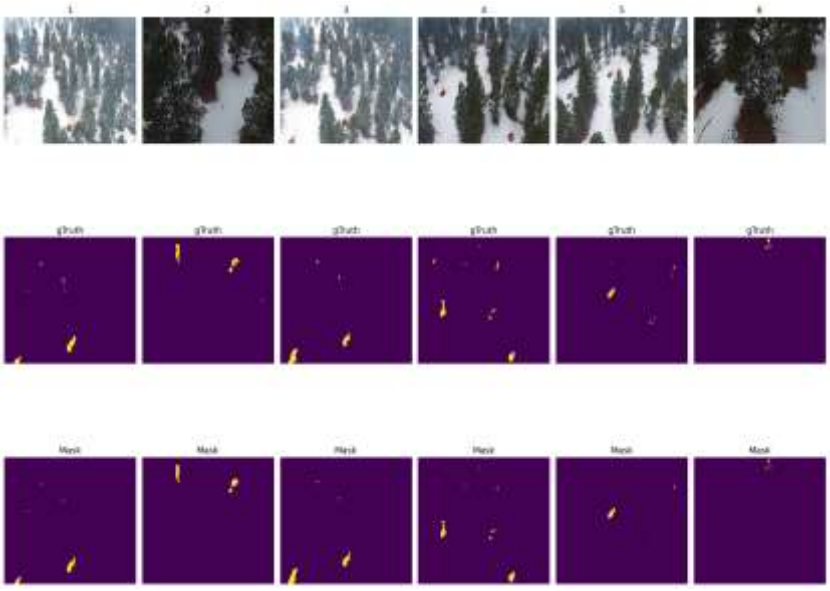


**Figure 8.** Results with U-Net: (**top row**) input image; (**middle row**) ground truth; (**bottom row**) generated mask segmentation.

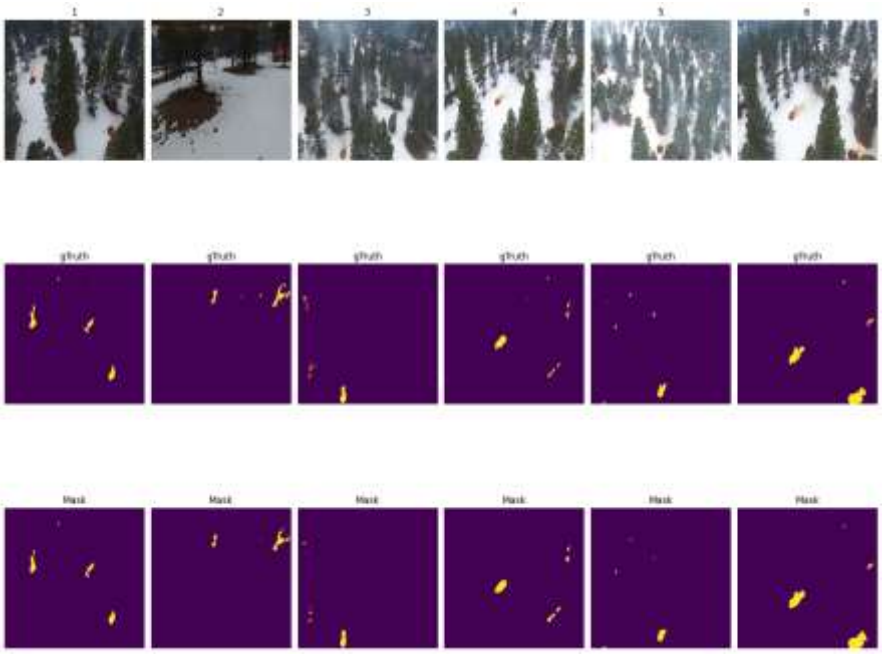


**Figure 9.** Results with DeepLabv3 using MobileNetV2 with frozen layers: (**top row**) input image; (**middle row**) ground truth; (**bottom row**) generated mask segmentation.

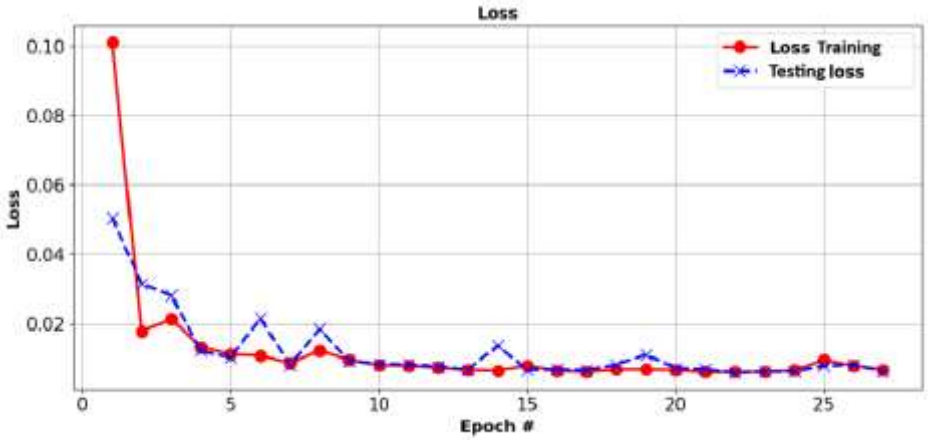



**Figure 10.** Loss curve while training and testing the DeepLabv3 model with ResNet50 backbone calculated at every epoch.

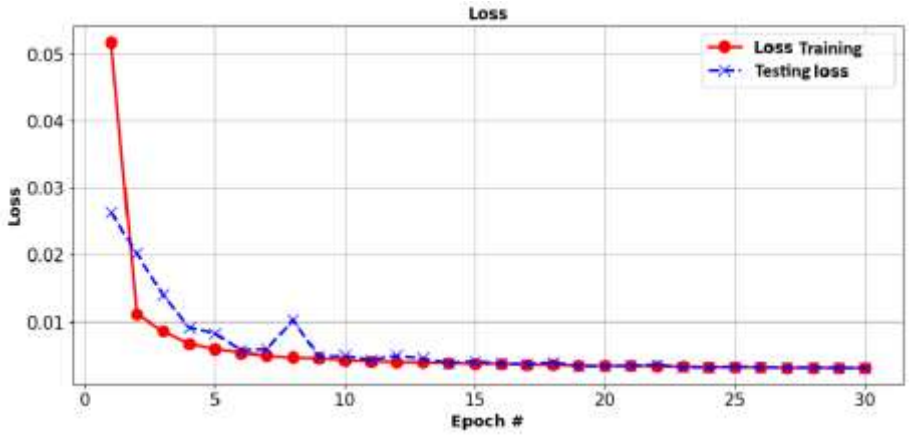



**Figure 11.** Loss curve while training and testing U-Net model calculated at every epoch.

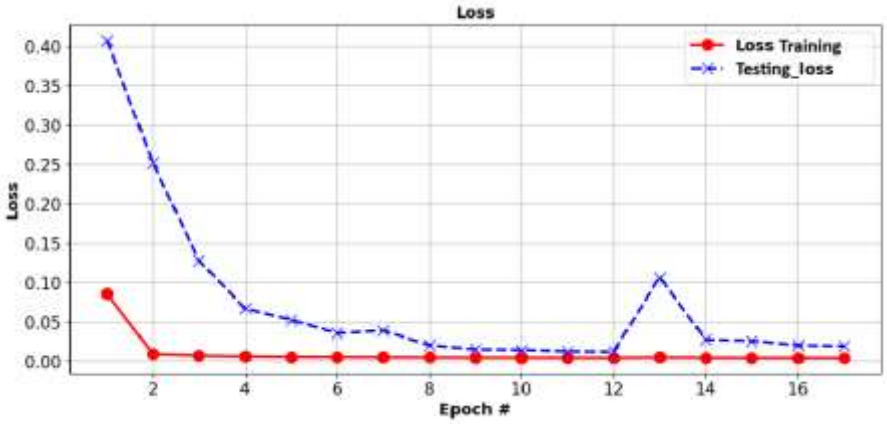



**Figure 12.** Loss curve while training and testing DeepLabv3 model using MobileNetV2 without freezing layers during training, calculated at every epoch.

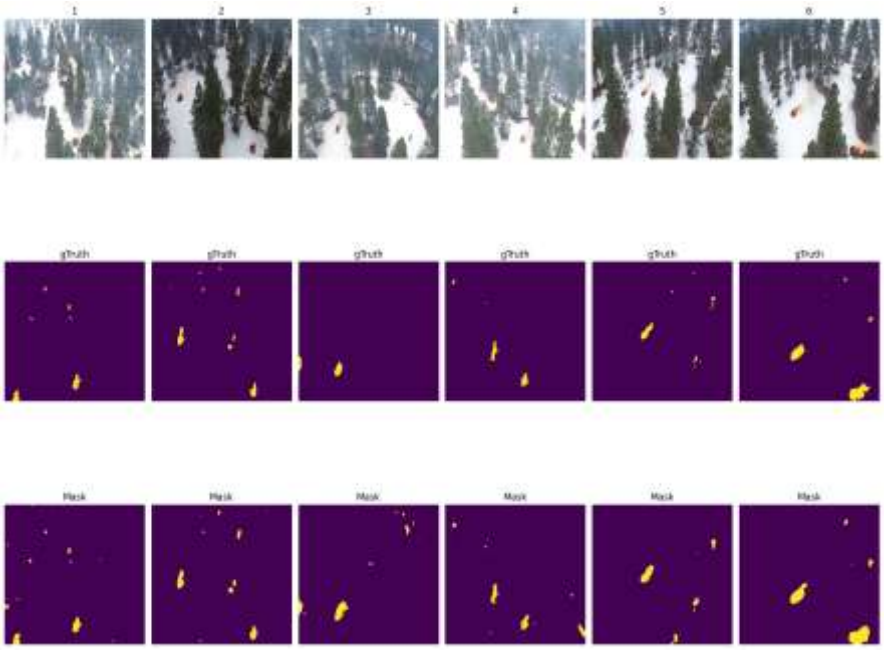


**Figure 13.** Results with DeepLabv3 using MobileNetV2 without freezing layers during training: (**top row**) input image; (**middle row**) ground truth; (**bottom row**) generated mask segmentation.

When frozen layers and initialization with ImageNet weights are implemented on DeepLabv3 model with MobileNetV2 as a backbone, the performance improved with a precision of 87.84% and recall of 83.22% (Table 2), leading to a F1-score of 84.85% and IoU of 49.71%, making it comparable to U-Net and surpassing the ResNet50 variant. As expected, freezing layers leads to an increased number of untrainable parameters but is not detrimental to performance. This strategy also helps stabilize the testing loss curve, as can be seen when comparing the loss curve of the model with frozen layers (Figure 14) with the loss curve with unfrozen layers (Figure 12). Freezing some layers during training does help prevent the effect of vanishing gradients while not increasing the overall complexity and accelerating the training process.

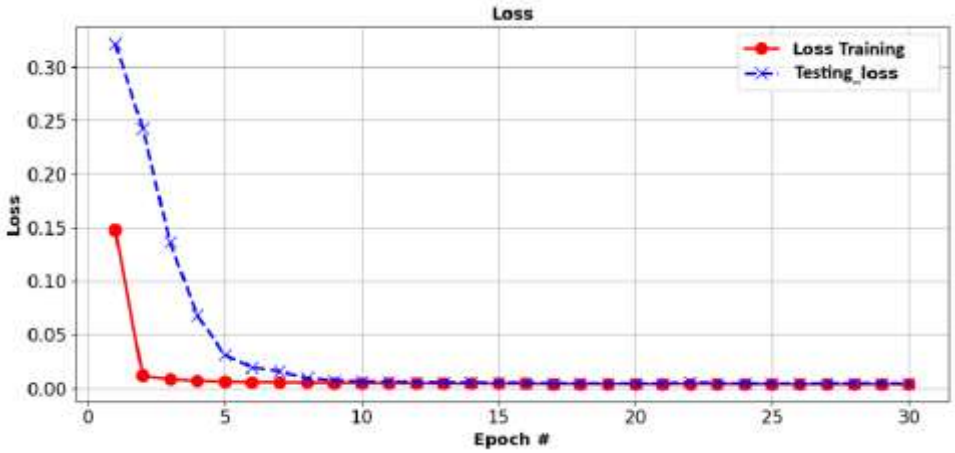



**Figure 14.** Loss curve while training and testing DeepLabv3 model using MobileNetV2 with frozen layers during training, calculated at every epoch.

The choice to downscale the input resolution to 256 × 256 also limits the model's performance, as many details are lost during resizing. Using the same U-Net architecture on downscaled input images results in a lower IoU than the 78.17% originally reported in [18] with full-resolution inputs. However, the performance of U-Net achieved in this comparative study demonstrates a similar precision and recall to the results obtained with the FLAME dataset in [18], which used a resolution of 512 × 512, and where they obtained a precision of 91.99% and a recall of 83.88%. This confirms that the proposed model is able to accurately extract features associated with the fire regions but the lower IoU may indicate that some features related to the shape and location of the fire areas might have been lost. However, the proposed simplified architecture and training strategy with frozen layers allows for better and more efficient processing, which is important in an application where computational resources are limited. Overall, this study provides evidence that the DeepLabV3 with a MobileNetV2 backbone form an effective model for wildfire segmentation on limited computational resources. Its precision and IoU scores are comparable to the state of the art but with significantly less parameters making it a significantly more compact model to implement on resource-limited hardware. Comparatively, U-Net remains a robust alternative yielding better precision and recall making it a suitable alternative in cases where robust performance is necessary. Thus, both models offer valuable tools in wildfire management systems.

## 7. Discussion

This research provides practical insight to design a deep learning-based approach to segmentation tasks for wildfire monitoring. One notable insight is the performance of DeepLabv3 model using MobileNetV2 as the backbone for segmentation tasks. The model was able to perform on par with U-Net while being significantly lighter in computational requirements and memory storage space, making it a suitable choice to deploy on resource-constrained UAVs. Interestingly, DeepLabv3 with ResNet50 as a backbone, while able to localize and accurately predict the shape of fire areas at a relatively large scale, was severely lacking in detecting smaller fire regions. U-Net has demonstrated very strong performance in this study. The inclusion of exponential linear unit (ELU) activation as proposed in [31] improved model convergence and stability. In future work, a similar approach could be taken with DeepLabv3 to enhance its performance.

To mitigate the impact of vanishing gradient in MobileNet, skip connections may be considered, although this approach risks increasing the complexity of models. Instead, the use of alternative activation function or batch normalization [36] may be investigated.

Another important consideration is the impact of the input image resolution on segmentation performance. Due to hardware limitations, in this study a reduced resolution of 256 × 256 is used. Reducing the image scale allows for faster training speed, and it makes the models less computationally taxing, but at the cost of losing some features that the model would advantageously capture to achieve more accurate predictions. Future study could focus on processing higher-resolution images for better accuracy while focusing on optimizing memory footprint to find a balance between accuracy and efficiency. Furthermore, this study excluded thermal imaging data, as we suggest that thermal imagery is more prone to generating false positives and that fusing RGB data with thermal imagery could be computationally taxing. However, thermal information could be interesting to explore future work in scenarios with low visibility due to smoke or darkness. One of the primary issues encountered during this project is the limited amount of data and the low variability in publicly available datasets on forest fire. While using a k-fold cross-validation approach has allowed to reduce the effect of a small dataset on evaluation metrics, expanding the dataset with additional aerial images of fire regions in a larger variety of contexts would further help the models to generalize. This is a problem highlighted particularly by some of the larger models like ResNet50, which require a large amount of data to perform well. This could be improved by applying data augmentation. Lastly, one implementation worth exploring in subsequent work is the implementation of techniques to further reduce the computational complexity of proposed models, such as pruning or quantization.

## 8. Conclusions

In this study we focused on the segmentation of fire regions in color images contained in images captured by the FLAME dataset. Three models are compared and strategies are formulated to further improve on their performance: U-Net as proposed in [18] and DeepLabv3 using two different backbones, ResNet50 and MobileNetV2 respectively. The experimental methodology involves data preprocessing, models implementation, and training/testing using k-fold cross-validation. Experimental results show that DeepLabv3 with MobileNetV2 as a backbone closely matches the performance of the state-of-the-art model, U-Net, while providing a significantly reduced computational footprint. Meanwhile, DeepLabv3 with a ResNet50 backbone, while better at localizing larger fire areas, tends to perform worse on cases of a smaller size and is significantly heavier computationally. Our findings suggest that this architecture would be preferable for cases where accurate prediction of the shape of the fire region is more important than computational efficiency. While U-Net remains a competitive choice due to its robust F1-score, its relative complexity compared to DeepLabv3 with MobileNetV2 backbone makes it a less desirable model for small UAVs.

A key contribution of this paper is the experimental and comparative evaluation of multiple DeepLabv3 models on the FLAME dataset. The work also proposes a method to stabilize training and minimize the vanishing gradient problem in MobileNetV2 without increasing its computational complexity, while also reducing its training time by freezing a portion of the backbone layers and initializing those weights to ImageNet. This study highlights the effect of different backbones on performance and points out the strengths and weaknesses of these different backbones depending on the application. This study highlights the potential of using DeepLabv3 and U-Net for fire region segmentation in imagery from aerial vehicles and provides an effective method to assist fire management operations by contributing a more reliable and efficient monitoring system.

Future work will focus on implementing fusion models for segmentation where thermal imaging is combined with the current model to improve segmentation accuracy.

Obtaining larger datasets with more variability will also contribute to improve the models ability to generalize.

**Author Contributions:**

**Funding:**

**Institutional Review Board Statement:**

**Informed Consent Statement:**

**Data Availability Statement:**

**Conflicts of Interest:**

## References

1. Sobha, P.; Latifi, S. A survey of the machine learning models for forest fire prediction and detection. *Int. J. Commun. Netw. Syst. Sci.* **2023**, *16*, 131–150.
2. Natural Resources Canada. Government of Canada. January 2025. Available online: https://natural-resources.canada.ca/forest-forestry/insects-disturbances/climate-change-fire (accessed on).
3. National Interagency Fire Center. 2020. Available online: https://www.nifc.gov/fire-information/nfn (accessed on).
4. National Interagency Fire Center. 2020. Available online: https://www.nifc.gov/fire-information/statistics/suppression-costs (accessed on 14 April 2025).
5. Canadian National Fire Database (cnfdb). 2023. Available online: https://cwfis.cfs.nrcan.gc.ca/ha/nfdb?type=poly&year=2023 (accessed on 14 April 2025).
6. Cost of Wildland Fire Protection. 2025. Available online: https://natural-resources.canada.ca/climate-change/climate-change-impacts-forests/cost-fire-protection (accessed on 14 April 2025).
7. Cheng, G.; Chen, X.; Wang, C.; Li, X.; Xian, B.; Yu, H. Visual fire detection using deep learning: A survey. *Neurocomputing* **2024**, *596*, 127975. Available online: https://www.sciencedirect.com/science/article/pii/S092523122400746X (accessed on).
8. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
9. Jiao, Z.; Zhang, Y.; Xin, J.; Mu, L.; Yi, Y.; Liu, H.; Liu, D. A deep learning based forest fire detection approach using uav and yolov3. In Proceedings of the 1st International Conference on Industrial Artificial Intelligence (IAI), Shenyang, China, 23–27 July 2019; pp. 1–5.
10. Ul Ain Tahir, H.; Waqar, A.; Khalid, S.; Usman, S.M. Wildfire detection in aerial images using deep learning. In Proceedings of the 2nd International Conference on Digital Futures and Transformative Technologies (ICoDT2), Rawalpindi, Pakistan, 24–26 May 2022; pp. 1–7.
11. Liao, C.-Y.W.H.-Y.; Wu, Y.-H.; Chen, P.-Y.; Hsieh, J.-W.; Yeh, I.-H. CSPNet: A New Backbone that can Enhance Learning Capability of CNN. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, WA, USA, 14–19 June 2020; pp. 1571–1580.
12. Redmon, J. Darknet: Open Source Neural Networks in C. 2013–2016. Available online: https://pjreddie.com/darknet/ (accessed on).
13. Li, M.; Zhang, Y.; Mu, L.; Xin, J.; Xue, X.; Jiao, S.; Liu, H.; Xie, G.; Yi, Y. A real-time forest fire recognition method based on r-shufflenetv2. In Proceedings of the 5th International Symposium on Autonomous Systems (ISAS), Online, 9–10 April 2022; pp. 1–5.
14. Chiang, C.-Y.; Barnes, C.; Angelov, P.; Jiang, R. Deep Learning-Based Automated Forest Health Diagnosis From Aerial Images." *IEEE Access* **2020**, *8*, 144064–144076.
15. He, K.; Gkioxari, G.; Doll'ar, P.; Girshick, R.B. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017.
16. Sridhar, P.; Devi, N.R.; Samyuktha, S.; Sanjeev, A.; Srinivasan, C. Wildfire detection and avoidance of false alarm using densenet. In Proceedings of the 13th International Conference on Computing Communication and Networking Technologies (ICCCNT), Virtual, 3–5 October 2022; pp. 1–4.
17. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 22–25 July 2017; pp. 2261–2269.

18. Shamsoshoara, A.; Afghah, F.; Razi, A.; Zheng, L.; Ful'e, P.; Blasch, E. The flame dataset: Aerial imagery pile burn detection using drones (uavs). 2020.
19. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. *International Conference on Medical Image Computing and Computer-Assisted Intervention*; Springer International Publishing: Cham, Switzerland, 2015.
20. Chen, L.-C.; Papandreou, G.; Schroff, F.; Adam, H. Rethinking atrous convolution for semantic image segmentation. *arXiv* **2017**, arXiv:1706.05587.
21. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
22. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
23. FireSmokeCustom. Firenet Dataset. January 2024. Available online: https://universe.roboflow.com/firesmokecustom/firenet-j1bfm (accessed on 19 April 2025).
24. Foggia, P.; Saggese, A.; Vento, M. Real-time fire detection for video surveillance applications using a combination of experts based on color, shape and motion. *IEEE Trans. Circuits Syst. Video Technol.* **2015**, *25*, 1545–1556.
25. Lascio, R.D.; Greco, A.; Saggese, A.; Vento, M. Improving fire detection reliability by a combination of videoanalytics. In Proceedings of the International Conference on Image Analysis and Recognition (ICIAR), Vilamoura, Portugal, 22–24 October 2014.
26. Abadi, M.; et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. 2015, Software Available from tensorflow.org. Available online: https://www.tensorflow.org/ (accessed on).
27. Chollet, F.; et al. Keras. 2015. Available online: https://keras.io (accessed on).
28. Bradski, G. The OpenCV Library. Dr. Dobb's Journal of Software Tools, 2000.
29. Kfold. 2025. Available online: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.KFold.html (accessed on 14 April 2025).
30. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* 2014, arXiv:1412.6980. Available online: https://arxiv.org/abs/1412.6980 (accessed on).
31. Clevert, D.-A.; Unterthiner, T.; Hochreiter, S. Fast and accurate deep network learning by exponential linear units (elus). *arXiv* 2015, arXiv:1511.07289. Available online: https://arxiv.org/abs/1511.07289 (accessed on).
32. Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
33. Xiao, X.; Mudiyanselage, T.B.; Ji, C.; Hu, J.; Pan, Y. Fast deep learning training through intelligently freezing layers. In Proceedings of the International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Atlanta, GA, USA, 14–17 July 2019; pp. 1225–1232.
34. Multiclass Semantic Segmentation Using Deeplabv3+. 2024. Available online: https://keras.io/examples/vision/deeplabv3 plus/. (accessed on 14 April 2025).
35. Keras Implementation of Deeplabv3+ with Mobilenetv2 Backbone for ifb Undegraduate Thesis. 2023. Available online: https://github.com/RWaiti/Keras-DeeplabV3Plus-MobilenetV2/tree/main (accessed on 14 April 2025).
36. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the 32nd International Conference on International Conference on Machine Learning (ICML), Lille, France, 6–11 July 2015; Volume 37, pp. 448–456.