remote sensing

MDPI

Article

SWIFT: Simulated Wildfire Images for Fast Training Dataset

Luiz Fernando, Rafik Ghali * and Moulay A. Akhloufi

Perception, Robotics and Intelligent Machines (PRIME), Department of Computer Science, Université de Moncton, Moncton, NB E1A 3E9, Canada; luiz.giongo@ufpr.br (L.F.); moulay.akhloufi@umoncton.ca (M.A.A.)

* Correspondence: rafik.ghali@umoncton.ca

Abstract: Wildland fires cause economic and ecological damage with devastating consequences, including loss of life. To reduce these risks, numerous fire detection and recognition systems using deep learning techniques have been developed. However, the limited availability of annotated datasets has decelerated the development of reliable deep learning techniques for detecting and monitoring fires. For such, a novel dataset, namely, SWIFT, is presented in this paper for detecting and recognizing wildland smoke and fires. SWIFT includes a large number of synthetic images and videos of smoke and wildfire with their corresponding annotations, as well as environmental data, including temperature, humidity, wind direction, and speed. It represents various wildland fire scenarios collected from multiple viewpoints, covering forest interior views, views near active fires, ground views, and aerial views. In addition, three deep learning models, namely, BoucaNet, DC-Fire, and CT-Fire, are adopted to recognize forest fires and address their related challenges. These models are trained using the SWIFT dataset and tested using real fire images. BoucaNet performed well in recognizing wildland fires and overcoming challenging limitations, including the complexity of the background, the variation in smoke and wildfire features, and the detection of small wildland fire areas. This shows the potential of sim-to-real deep learning in wildland fires.

Keywords: fire recognition; synthetic data; simulation; wildfire; 3D platform; deep learning; SWIFT

Citation: Fernando, L.; Ghali, R.; Akhloufi, M.A. SWIFT: Simulated Wildfire Images for Fast Training Dataset. *Remote Sens.* **2024**, *16*, 1627. https://doi.org/10.3390/rs16091627

Academic Editors: Paraskevas Tsangaratos, Wei Chen, Ioanna Ilia and Haoyuan Hong

Received: 7 April 2024
Revised: 29 April 2024
Accepted: 30 April 2024
Published: 2 May 2024

1. Introduction

In recent years, the rise in wildfires has been noted and partially attributed not only to an increase in climate change [1,2] but also to human activities, whether accidental or purposeful [2]. This phenomenon has been increasing in the past years, with projections for an even worse scenario over the next 40 to 50 years [3]. In 2023, in Canada, a record-breaking wildfire occurred, with 6623 fires burning a total of 18,401,197 hectares [4]. In Europe, 2022 was the worst year for forest fires, resulting in a total of approximately 900,000 hectares burned and a higher risk of economic losses [5]. In addition, in recent years, the United States has faced losses estimated between USD 63.5 and 285 billion [6]. Following economic losses, human health is also at risk when dealing with wildfires. While the loss of human lives directly associated with wildfires is usually not high [6], there are health problems related to them [7].

Consequently, as a response to these recent events, there has been a strong effort to develop ways to combat and monitor this natural disaster. Recently, DL (deep learning) methods have been used to process images and videos collected from satellites [8,9], surveillance cameras [10], and unmanned air vehicles such as drones [11,12]. These models have shown high performance, proving their effectiveness for the detection, segmentation, and classification of wildfires at different stages [13–15].

DL methods require visual and numerical data to be trained, meaning that all of these methods will require a vast amount of data collected from wildfires [16]. However, there are few datasets available, and of those, some contain as few as 226 images [17], 1135 images [18], 1900 images [19], and 2003 images with their corresponding binary

masks [20] for training purposes. Data on these events are usually collected and compiled by government agencies, according to their necessity [21]. Experimental wildfire data are also limited by missing data, notably meteorological data and fuel maps [21,22]. In addition, obtaining new fire data, such as through controlled burns, necessitates substantial backing from diverse stakeholders and might pose challenges in execution.

A simple way to generate visual data for wildland fires is by using digital twin models of real-life scenarios [23]. This digital solution allows for any project to develop a higher amount of data. As an example, Lockheed Martin's CMM (Cognitive Mission Manager) project used a digital twin and machine learning to help firefighters reduce fire damage by determining fire perimeters and simulating fire propagation [24]. By exploiting this innovative method, it is possible to generate a substantial amount of wildfire data without the constraints inherent in real-world conditions and improve the performance of deep learning models.

For such, we developed a novel dataset, namely, SWIFT (Simulated Wildfire Images for Fast Training), for wildland fire recognition, segmentation, and simulation. SWIFT includes more than 69,000 synthetic images labeled as wildfire, smoke, both smoke and fire, and no-fire, and 15 videos depicting various scenarios of complex backgrounds, wildfire, and smoke, along with their corresponding masks. It also includes data on related environmental conditions, such as temperature, humidity, wind direction, and wind speed. This dataset was generated from multiple viewpoints, offering views from inside the forest, views close to active fires, ground views, and aerial views, providing comprehensive coverage of wildland fires. Additionally, SWIFT was used for training the recent DL methods CT-Fire [25], BoucaNet [26], and DC-Fire [27], as well as for addressing the limitations of few available wildland fire data.

This paper presents the following contributions:

1. A novel dataset, namely, SWIFT, consisting of a large amount of data labeled as wildfire, smoke, both fire and smoke, and no-fire, is developed to improve wildland fire recognition, segmentation, and simulation tasks.
2. By exploiting the SWIFT dataset, DL models can be trained using diverse fire scenarios, allowing the development of accurate prediction and detection of fire models.
3. Recent DL models are explored and adopted for wildland fire recognition. These models are trained using the SWIFT dataset and show high performance when tested on real images (sim-to-real), overcoming challenging limitations, such as background complexity and detection of small wildfire areas.

The rest of this paper is organized as follows: Section 2 introduces recent developments in wildfire simulation and digital forest generation, published wildfire datasets, and wildfire recognition methods. Section 3 presents materials and methods, including DL methods (BoucaNet, CT-Fire, and DC-Fire) and the development and details of the SWIFT dataset. Section 4 illustrates the experimental results obtained using these DL methods. Section 5 discusses these results. Section 6 summarizes the paper.

2. Related Works

2.1. Three-Dimensional Simulation Platform

A 3D platform for simulating wildfires is any system with visual control in a three-dimensional environment that allows the setup of and consequently simulates fire in the scenario. The simulation of fire related here is necessary to obtain realistic visual data and not a truly realistic simulation of a wildfire-scale fire scenario. This platform can be easily used to generate digital wildland fire data, which are very important compared to real data, which are generally difficult to collect and require significant investment and risk [28].

Numerous projects have been developed to simulate data collection in digital environments and many others to digitally simulate wildland fire, with fewer works undertaken to integrate both ideas into the same platform. In 2017, Microsoft undertook the development of AirSim, a simulation interface developed for Unreal Engine (UE). Being open source, AirSim is one of the best simulation tools developed for drones and other vehicles. It is an

example of the training of DL with digital twins, with APIs exposed for external control and support for popular protocols such as MavLink [29]. With interaction between DL and simulation as its main focus, AirSim offers an easy acquisition of simulation data for images and parameters. It is greatly influential not only for its ease of use and versatility but also for being one of the first tools developed for game engines used for the acquisition of data. Bhattarai and Martínez-Ramón [30] developed a deep Q-learning agent for path planning in dynamic environments caused by fires. Using the UE platform with AirSim tools, they exploited the interactivity with the environment and the replay of experience to accelerate the training of the agent. Presented as a helpful tool for firefighters during stressful scenarios, the latter outperformed models trained with alternative path planning systems. Ma et al. [31] developed a simulation environment based on ROS (Robot Operating System), AirSim, and PX4 SITL (Software In The Loop, v1.12.3) for testing quadcopter aircraft. The matching of their simulation environment with real-life scenarios yielded promising results.

Stava et al. [32] studied the inverse procedural modeling of trees through the usage of parameters and definitions. They simulated the realistic growth of any plant, involving interactions with the environment such as walls and other plants. They also used a modeling approach that utilizes Monte Carlo Markov chains to find optimal parameters and a polygonal model as input. Makowski et al. [33] developed a project on the natural growth and simulation of a biome, from its bare land to the eventual formation of flora, following a completely procedural method. This method is capable of simulating up to 500,000 individual plants, with resource competition, diverse tropism, and other biological features. Pirk et al. [34] presented an interactive wood combustion model for procedural tree models. In this method, trees are represented as connected points for branches and polygonal surface meshes for combustion. The points are associated with physical properties that affect the fire behavior, such as moisture, fuel, and stress, all of which dictate the simulation. The model is interactive and can simulate multiple trees at the same time. Hädrich et al. [35] continued the previous research of Pirk et al. as a wildfire simulation on the scale of entire forests by simulating the combustion process of each individual tree. This wildfire simulation focuses on the fire spread of the combustion of plants, considering heat transfer, char insulation, and mass loss. It runs at an interactive pace, letting the user interact with the simulation and explore different types of actions that affect the fire, such as rain and the creation of firebreaks.

In 2021, Lockheed Martin initiated an internal project, namely, CMM, to demonstrate the usefulness of multidomain processes and machine learning to support firefighters during their wildland fire missions. This project was subsequently partnered with the Colorado Division of Fire Prevention and Control. It integrates MX-15 data to draw the fire perimeter using machine learning. Then, the fire spreading for the next 24 h was modeled using a physics-based simulation, taking into consideration topography, weather, and fuel type of the area [24].

Likewise, for wildfire simulation and prediction, some projects are being developed to generate digital twins of forests and environments. This new method employs LiDAR (light detection furthermore, ranging) technology to generate individual models of real-life environments. Nita [36] studied the use of GeoSLAM mobile LiDAR scanners and VirtSilv AI to produce digital twins of individual trees in a plot. This method was evaluated for around 1.4 thousand trees, showing a high accuracy compared to existing methods. Buonocore et al. [37] also introduced a framework for designing digital forest twins. They integrated various state variables at the forest and tree levels to generate a virtual forest copy. The tree or forest is formed by three layers: the first layer is composed of the biotic and physical state variables, the second layer contains the tools for digitizing and recording the state variables, and the third layer consists of the physiological processes.

2.2. Wildland Fire Datasets

Many projects have been developed to emulate wildland fires, greatly varying between methods and results. However, there is little undertaken in the area of data generation for training and testing wildland fire models, which is a modest area of development. In addition, there is a lack of tools to implement a scenario and simulate this environment in order to obtain new data.

Several wildland fire datasets have been developed with different resolutions and objectives, as presented in Table 1. BowFire [17] is a small dataset containing a few images ranging from urban fires, which affect artificial human constructions, to large-scale wildland fires. It also includes images of false-positive scenarios, with red and yellow objects and sunsets. The FLAME dataset [20] consists of aerial images and videos collected from RGB and thermal cameras, with high-resolution images of wildfire in snowy regions. It also contains a mask of wildfires for DL segmentation tasks. CorsicanFire [18] consists of both RGB images and near infrared; the latter were obtained with a longer exposure time. It also contains binary mask images used in the context of fire segmentation. FD-Dataset [38] combines two other datasets, BowFire [17] and dataset-1 [39], which contains wildfire and background images collected from the internet. ForestryImages [40] is a public dataset developed by the University of Georgia's Center for Invasive Species and Ecosystem Health. It includes a large number of images attributed to different categories, some of which are not of relevance to DL wildfire training, such as forest insects and pests. FiSmo [41] contains a large collection of data from the internet; these are labeled fire, non-fire, and ignore. It combines four other datasets, BowFire [17], Flickr-FireSmoke [42], Flickr-Fire [42], SmokeBlock [43]. FLAME2 [12] is also a public dataset from prescribed wildfires in 2021 in the canopy pine forest of northern Arizona. It includes a large amount of aerial images extracted from video recorded by a Mavic 2 Enterprise Advanced dual RGB/IR camera.

Table 1. Wildland fire datasets overview.

Ref.	Dataset Name	Image Type	Labeling Type	Fire Data	No-Fire Data
[17]	BowFire	Terrestrial	Classification Segmentation	119 images; 119 mask images	107 images
[20]	FLAME	Aerial	Classification Segmentation	17,855 images; 2003 mask images	30,155 images
[18]	CorsicanFire	Terrestrial	Segmentation	1135 images with their masks	None
[38]	FD-Dataset	Terrestrial	Classification	50,000 images; 14 videos	25,000 images; 17 videos
[40]	ForestryImages	Terrestrial	Classification	317,921 images	None
[41]	FiSmo	Terrestrial	Classification	9448 images; 158 videos	None
[12]	FLAME2	Aerial	Classification	39,751 image pairs; 7 video pairs	13,700 image pairs

As illustrated in Table 1, there is a clear imbalance when comparing terrestrial data to aerial. This problem is found when analyzing the actual number of ground-truth masks available in these datasets. Generating a mask is naturally difficult and time-consuming. It almost always requires the manual labor of experts to properly describe and label the fire zone in thousands of images. In addition, there is the human error risk. Machine learning methods for autolabeling have been developed, but they carry a lower-than-ideal accuracy for their labeling, triggering false alarms, especially with tricky scenarios such as sunsets [12]. This factor helps in understanding why many images are missing their fire-zone description. With no mask, a DL method is unable to learn from the dataset unless manually created. Some of them lack mask images, and others are missing their labeling.

Additionally, the quality of these datasets depends on the methodology employed to generate data, with the best-case scenario being a prescribed fire and setup of cameras and drones to record as much data as possible, similar to what was employed during the prescribed fire in Arizona to generate FLAME2 [12]. Prescribed fires are not easily arranged, requiring many aspects to be solved before the permission to start one is received. These many aspects can be separated into three categories: risk-related challenges, such as the fear

of liability or uncontrolled spread; resource-related challenges, such as limited financial resources and crew experience and availability; and regulation-related challenges, such as environmental regulation and poor meteorological conditions for burning [28]. Wildfires are the maximum extreme when referencing fire; with their heat values, cameras become fragile when trying to record their data. Even for drones far away from the center of a wildfire, the radiation heat can still be enough to cause damage to the electronics and ruin its ability to collect meaningful data.

In conclusion, the use of fire datasets is still limited; sometimes we are without the specific type on which a DL method will be trained, or the dataset is without labeling from an expert that guarantees performance. In these cases, a 3D platform would be immensely useful for allowing not only the acquisition of data but also the automatic creation of any mask at the same time. This would considerably reduce the danger and cost of generating these data, as well as the manual labor and time required to generate masks.

2.3. Deep Learning Approaches for Wildland Fire Recognition

Numerous DL models have been proposed for detecting and recognizing wildland fires, as well as for reducing their damage. Table 2 represents recent DL methods used to identify and recognize wildfires using aerial, ground, and satellite images.

During the creation of the FLAME dataset, a wildfire classification method was tested utilizing a DCNN (deep convolutional neural network) called Xception [20], which achieved an accuracy of 76.23%. Sandra and Risteska [44] studied five deep learning models, namely, VGG19, VGG16, ResNet50, Inception, and Xception, for recognizing wildfires. Based on a transfer learning technique that allows the model to learn from a different task and transfer this newfound knowledge to fire recognition, ResNet50 achieved the best result with an accuracy of 88.01% on the FLAME dataset. EfficientNet-B5 and DenseNet201 were combined to improve the performance of the wildland fire recognition task [45]. Using the FLAME dataset, they reached an accuracy of 85.12% better than state-of-the-art methods. In [46], FT-ResNet50, as a modified ResNet50 method, was also adopted to recognize forest fires using aerial images. It obtained an accuracy of 79.48% using the aerial dataset FLAME.

The VGG16 method was also employed to classify wildland fire in aerial images [12]. Using a large amount of images collected from the FLAME2 dataset, this method showed a high performance with an accuracy of 99.91%. It outperformed baseline methods, such as Xception, MobileNetV2, and ResNet18. A lightweight model, namely, FireXnet, was developed for improving the performance of forest fire classification task [47]. It integrates an explainable artificial intelligence method, SHAP (SHapley Additive exPlanations), to enhance its decision. It was tested on various datasets, achieving an accuracy of 98.42% better than the InceptionResNetV2, VGG16, MobileNetV2, InceptionV3, and DenseNet201 models. Wong et al. [48] proposed a novel wildland fire image classification method, namely, Reduce-VGGNet. This method is a modified VGG16 model, replacing the three fully connected layers with two fully connected layers and using the softmax method. It achieved a great classification performance with an accuracy of 91.20% using the FLAME dataset.

Ghali and Akhloufi [25–27] proposed three ensemble learning methods, namely, DC-Fire [27], CT-Fire [25], and BoucaNet [26], to identify and classify wildfires, as well as addressing their related challenging limitations, such as background complexity, detecting small fire and smoke areas, and fire variability in terms of size, shape, and intensity. The first method, DC-Fire, combines the two CNNs (convolutional neural networks) DenseNet and EfficientNet to extract wildfire features using infrared aerial images as input data. It achieved a high accuracy of 100%, outperforming existing methods such as Xception, VGG16, LeNet5, ResNet18, and MobileNetV2. The second method, CT-Fire, integrates the DCNN method RegNetY-16GF and the vision transformer EfficientFormerV2 to identify flames from many different datasets, including from both terrestrial and aerial images. It showed great results, with an accuracy of 87.77%, 99.62%, and 85.29% using aerial, ground, and both ground and aerial images, respectively. The third method, BoucaNet, combines the EfficientFormerV2 and EfficientNetV2 models as its backbone for recognizing smoke in

satellite images. Test results showed that BoucaNet achieved an accuracy of 93.67% better than published methods such as SmokeNet. This demonstrates its potential to distinguish similarities between smoke, dust, haze, and cloud classes. Jonnalagadda and Hashim [49] developed a novel DL method, SegNet (Segmented Neural Network), to enhance the processing time of the wildfire detection method. SegNet is a simple CNN consisting of five convolutional layers, five ReLU activation functions, and two max pooling layers. SegNet's input was a segmented fire image, produced by dividing an input image with a resolution of 1280 × 720 pixels into 12 smaller segments, each measuring 320 × 240 pixels. This method showed a high performance with an accuracy of 98.18% and fast processing speed, enabling real-time detection.

Table 2. Deep learning models for wildfire recognition.

Ref.	Methodology	Object Detected	Dataset	Image Type	Accuracy (%)
[20]	XCeption	Flame	FLAME: 47,992 images	Aerial	76.23
[44]	ResNet50	Flame	FLAME: 47,992 images	Aerial	88.01
[45]	EfficientNet-B5, DenseNet-201	Flame	FLAME: 48,010 images	Aerial	85.12
[46]	FT-ResNet50	Flame	FLAME: 47,992 images	Aerial	79.48
[12]	VGG-16,	Flame/smoke	FLAME2: 53,451 images	Aerial	99.91
[47]	FireXnet	Flame/smoke	Kaggle, DFire, FLAME2: 3800 images	Aerial Terrestrial	98.42
[48]	Reduce-VGGNet	Flame/smoke	FLAME: 1900 images	Aerial	91.20
[49]	SegNet	Flame	Custom: 10,242 images	Aerial Terrestrial	98.18
[25]	CT-Fire	Flame	FLAME, CorsicanFire, DeepFire, FIRE: 51,906 images	Aerial Terrestrial	87.77 99.62
[27]	DC-Fire	Flame/smoke	FLAME2: 53,451 images	Aerial	100.00
[26]	BoucaNet	Smoke	USTC_SmokeRS: 6225 images	Satellite	93.67

3. Materials and Methods

This section describes the development of SWIFT, briefly explaining how each aspect of the platform was determined and created using UE features. Following this, a detailed explanation of how data are generated through the use of predetermined movie renders and graphical shaders. Finally, the proposed DL methods are presented.

3.1. SWIFT Development

To develop SWIFT, various IDEs (integrated development environments) were considered. Among these, Unity and Nvidia Omniverse are the best-known development tools. Unity is in a very similar state to UE, containing very well-written documentation and countless guides made by the community, but it lacks graphical technologies that are ready to use in UE and are critical for the development of SWIFT, such as Nanite and Lumen. On the other hand, Nvidia Omniverse lacks easy-to-use documentation and guides made by the community that are expected from other IDEs like UE and Unity, therefore requiring more time spent on learning the software. Nvidia Omniverse's software is even more complicated; it requires an RTX graphics card and expensive proprietary hardware that many developers may not have for their projects, meaning that developing SWIFT in this restrictive IDE could well limit future development opportunities. Accordingly, UE was chosen to create, develop, and generate SWIFT. Developed by Epic Games, it was originally designed as a gaming engine and has quickly gained the support of scientific developers. UE has impressive graphical capabilities and technologies, facilitating easier and quicker development for SWIFT. This allows for a greater focus on developing the simulation while UE handles the graphical aspects of the dataset. UE is constantly being updated and upgraded with the industry's best graphical simulation and gaming technologies, keeping it ahead of other IDEs in graphical performance. Therefore, it has a broad lifetime for the future, with plentiful support and maintenance from Epic Games.

SWIFT underwent development during versions 5.2 and 5.3 of UE. These versions are pivotal since PCG was introduced in version 5.2 and Nanite received significant updates in version 5.3, which now includes landscape support.

Nanite and Lumen, technologies crafted by Epic Games for UE, stand as important advancements in graphical processing. Nanite reduces process time for rendering geometries by virtualizing geometries; with this, it can dynamically change the resolution of triangles to better accommodate screen space. Lumen, on the other hand, acts as a light system; with a hybrid ray-tracing technology, it achieves indirect lighting and dynamic illumination with significant speed and light-processing demand. Temporal Super Resolution (TSR) is another graphical technology developed for UE version 5, enabling the use of heavy technologies such as Lumen. TSR is an antialiasing technology, which improves image resolution through the use of algorithms by rendering parts of the screen in extreme definition and then meshing them together to form a single image. By rendering small parts per frame, TSR significantly reduces the computational load of each frame, allowing the use of heavy features like Lumen with a reduced performance.

Inspired by Makowski et al.'s work on biomes [33], three biomes that could be formed in ecological ecosystems found in Canada were developed, as shown in Figure 1: temperate and boreal forests, with their respective dense vegetation, and bare tundra for regions that suffer from extreme cold conditions. These biomes were generated using UE 5.2's new tool, PCG. A modular forest model was quickly developed for each type, with different densities, heights, and steepness tolerances as well as flora. PCG was employed to allow the project to adapt to any landscape with its forest types. This feature allowed the project to be completely automated regarding its scenarios, from landscape to forests.

Figure 1. Images of developed biomes for SWIFT, from left to right: boreal, temperate, and tundra.

Nowadays, the digital elevation model that represents the elevation of terrain and objects is an extensively developed technology, with satellite images already supporting the extraction of the mentioned maps. UE allows the importing of these maps for the generation of landscapes; therefore, any terrain from any place in the world can be imported into the project. Since PCG is used to generate vegetation, these realistic terrains are automatically covered by millions of trees and plants and can quickly be used to simulate our wildfire.

3.2. Data Simulation

The full simulation of a phenomenon such as a wildfire is still an open issue. Wildland fires at a landscape scale are so massively complex and their processing is so heavy that no proper understanding has been developed yet. In addition, with many factors affecting a wildfire, such as weather conditions, flora properties, physics simulations, and heat transfer, modeling wildfires is a challenging task [50].

SWIFT simulates weather conditions with a few parameters, i.e., wind speed, wind direction, humidity, and temperature, which dictate the simulation conditions and directly affect wildland fire behavior. Rain and snow are also simulated; when humidity is

above 85%, rain starts falling, and when the temperature drops below −4 degrees, snow completely overtakes rain and then paints everything white.

Plants are simulated individually. They contain their values and parameters that determine the fire behavior when burning and heating. Thus, they contain their own fuel, heat transfer, plant humidity, plant temperature, flash point, and a flag indicating whether they are already burning or not.

Initially, SWIFT's fire model was greatly inspired by the work of Pirk et al. [34] on burning trees, incorporating functionalities such as heat being consumed to evaporate the liquid content of plants instead of building up inside the plant, the pyrolysis of material, and the decrease in fuel efficiency. However, when testing the simulation of thousands of individual fires, the performance of the simulation decreased significantly. Consequently, some features were simplified, with the understanding that the focus of SWIFT is on the visual fidelity of fires rather than the accurate simulation of this phenomenon.

The simulation performance requires more improvements to be interactive for large-scale fires, but it is better than the first prototype and is enough to be rendered with no performance loss. Fire in SWIFT is simulated by separating the visual and mechanical aspects. The visual fire is built with a combination of UE Niagara particles, a system that allows complex simulations by unifying features such as indirect light, heat distortion, or mirage, and different types of particles for fire, embers, and smoke.

Fire contact is determined by two geometries, a sphere around the center of the fire and a fan-shaped polygon. These geometries are determined by the fire size, the strength of the wind, and its direction. Any flora inside or in contact with these two objects is considered within the fire's reach and will be directly affected by it. When affected by the fire, a plant's temperature increases with each frame following Equation (1).

$$AddedTemperature = SP \times \frac{(PHA \times (1 - PH \times 0.5)) \times FH}{PF \times 0.33} \quad (1)$$

where SP is the current simulation speed, PHA is the plant's heat absorption, PF is the plant's total fuel, PH is the plant's humidity, and FH is the fire's heat. Many aspects of this mathematical statement are transformed into multiplications from fractions to reduce process costs. Firstly, the humidity value is calculated, generating a range between 0.5 and 1. This value affects the rate of heat accumulation, meaning that a plant with higher humidity will heat more slowly. The humidity absorption is then multiplied by the heat absorption of the plant and the heat of the fire. This calculation gives the value of the heat absorbed by the plant, which is then divided by one-third of its total mass, represented by the remaining fuel. This means that a larger flame will be required to heat a bulkier plant. This process is then multiplied by the simulation speed, which simplifies the calculation for multiple iterations without needing to recalculate the value each time. This simplification implies minimal errors, which are not significant in large-scale simulations. When the plant's temperature rises sufficiently to reach its flash point, i.e., the point at which the material evaporates into flammable gas and ignites, the simulation marks the plant as burned and creates a new fire to consume it.

Fire cannot burn on a large scale if the flora it burns is small or has little available fuel, which means that the heat of the fire is limited by the biomass of the flora. However, when the fire adequately consumes its fuel source, the increase in heat follows Equation (2).

$$IncreaseHeat = SP \times (((1 - PH \times 0.5) \times 0.45) - (GH \times 0.4)) \quad (2)$$

where SP is the simulation speed, PH is the plant's humidity, and GH is the global humidity. Similar to Equation (1) of flora temperature increase, the fire can only burn the biomaterials available to it. This is simulated by reducing material consumption according to humidity; all functions share this same decrease of up to 50% due to flora humidity. Then, this value is multiplied by 0.45, an arbitrary value that determines how fast the heat of a fire will increase. Ideally, this value should be related to fuel type and energy release, but for

simplification purposes, an arbitrary number was chosen. In addition, there is a second part to this function in the form of a heat reduction according to the global humidity to emulate saturated surroundings that are extremely humid, such as those found in snow or rain. This means that our simulated fire will be progressively dampened and eventually put out by wet weather conditions.

This fire has a visual effect that increases as the fire heats up. This increase in size is also influenced by the amount of fuel remaining, which indicates that a single example of flora that is very close to dying or is initially small will not produce a large flame in the same way as a tree. When the example of flora achieves a temperature above 60 degrees, its water content begins to evaporate, slowly decreasing over time, as shown in Equation (3).

$$RemovedHumidity = SP \times (0.001 \times \frac{PT - 60}{300 - 60}) \quad (3)$$

where SP is the simulation speed and PT is the plant's current temperature. This equation normalizes a value to a range between the minimum of 60 degrees and the maximum of 300 degrees and then multiplies the result by 0.1%. Therefore, when the temperature reaches 300 degrees, the plant will lose 0.1% of its humidity each time the function runs. Since the plant temperature is affected by its mass, the plant's mass will also affect this calculation.

3.3. SWIFT Data

To generate the SWIFT dataset, a UE feature, namely, Sequencer, was widely used to produce movie renders and mass images. A complete forest environment was developed, as shown in Figure 2. It was directly generated using heightmaps extracted from satellite images of a region in Moncton, New Brunswick, Canada. PCG was then applied to this generated landscape, producing three distinct biomes over the expanse of the Moncton region.

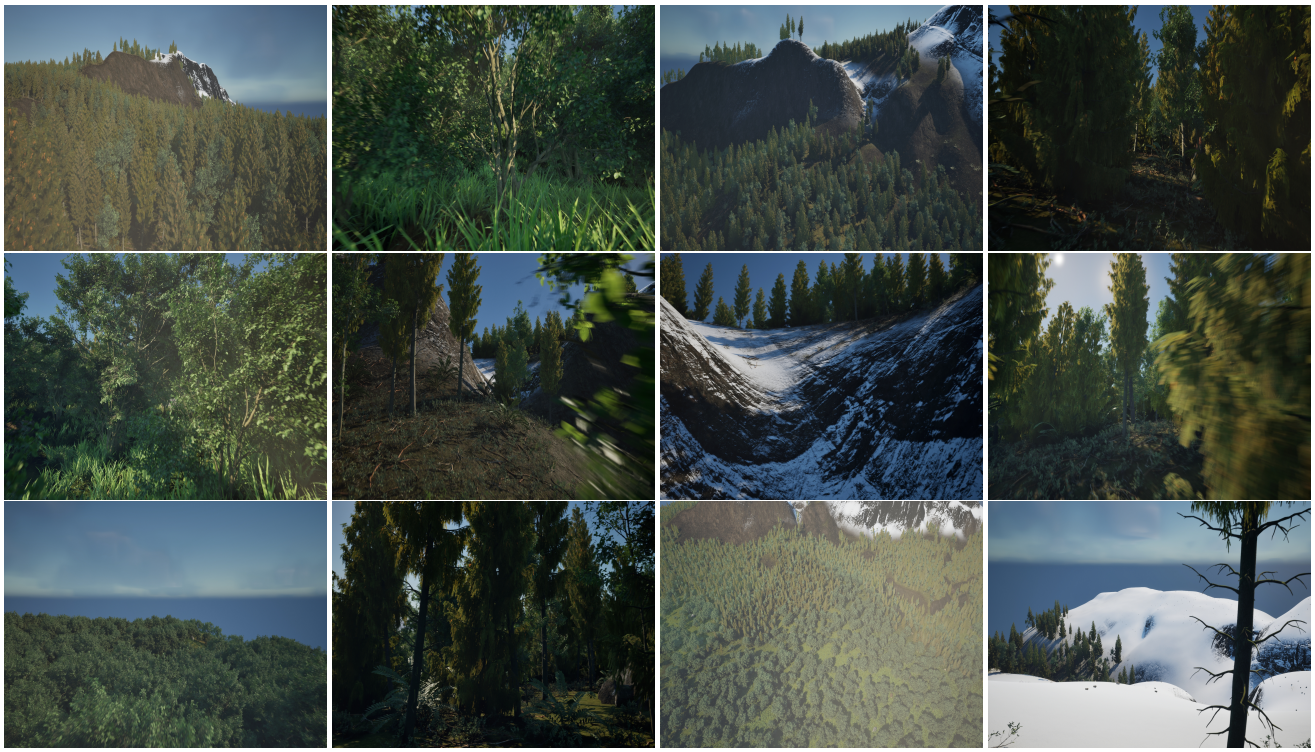

Figure 2. Background image examples.

This environment simulates approximately 4.7 million plants, encompassing various types from trees to grass chunks. These plants lack code functionality, which greatly

improves performance for such a large environment but also makes it difficult to simulate the actual spread of a fire.

All simulations and data were processed on a machine with an NVIDIA GeForce RTX 3090, an AMD Ryzen 9 5900X 12-Core Processor 3.70 GHz, and 32 GB of RAM (NVIDIA, Santa Clara, CA, USA).

Four wildfire scenarios were developed in the vast Moncton landscape, each with different angles and viewpoints. They include ground and aerial views, providing a comprehensive visual exploration of each scenario as well as a rich and diverse dataset for training DL models. The videos were also recorded in a daytime simulated environment without weather change (a clear sky, a temperature of 21 Celsius, and at 1 p.m.).

With the simulation ready, four render sequences were developed to showcase multiple wildfire scenarios. Each sequence ranged from 1 to 2 min and produced both images and videos for SWIFT. These sequences were also designed to provide a wide variety of data by changing the angle and position of the camera. Quick movements simulated fast-flying drones resulting in blurry images, while slower movements captured clearer images. Aerial views provided an overview of the fire, and the final sequence included both distant and close-up fire shots.

These render sequences were then processed; the time to process each frame was approximately 0.127 s, with an average of 5000 images per render sequence before filtering out bad frames. It took roughly 11 min to render an entire scenario of 2 min and 47 frames per second. These were then converted into an MP4 (MPEG-4) video file using the FFmpeg conversion algorithm.

Both normal images and their corresponding ground truths (masks) were generated. They required different passes to be generated. For each class sequence and its ground truth, the processing time was effectively doubled. The switch between images was performed internally via UE by changing material properties and rendering configurations. When generating a ground-truth mask, all lighting features were turned off, turning the world completely black. However, the fire or smoke material was set to be emissive, allowing it to be visible even without external lighting. This technique ensured that the fire or smoke was the only visible element in the scene.

RGB images and their corresponding ground-truth images were generated separately using the Niagara feature. The latter enables consistent simulations that remain unchanged even after being regenerated numerous times. Adding to that, this feature allowed us to develop a single scenario and record it from multiple angles and image types without any differences from the actual fire simulation.

Ground-truth images are the basis for DL training. They are a copy of a normal image but in binary or grayscale format. The learning method can recognize the fire in the normal image by matching it with the corresponding ground truth. For fire images, we generated grayscale images as the corresponding ground-truth images, as shown in Figure 3. When these images include more than one class, such as fire and smoke in the same image, the ground truth is set to adjust dynamically, displaying multicolored pixels whenever the two masks overlap, as depicted in Figure 4.

Smoke ground-truth images were generated using two methods, as depicted in Figure 5. One is a normal grayscale image, which can be used to acquire specific information such as the strength of the effect, i.e., when a flame has moved far from its source and is quickly dying or when smoke particles are less dense and therefore less visible. The second method is the use of a graphical binary filter. This code analyzes the pixels in the frame and paints them either completely white for pixels that have values above 0.015 for their respective colors or completely dark in the case of failure to achieve this threshold.

The SWIFT dataset comprises around 70,000 images of 1920 $\times$ 1080 pixels, as illustrated in Table 3. These images were filtered to improve quality, notably by removing overly blurred images. Three background videos were generated, containing around 18,000 images. They contain no fire or smoke and therefore required no ground-truth variants.

For actual wildfire data, four scenarios were developed, each of which was used to generate ground-truth data. These scenarios were then processed in different configurations: only fire effects, only smoke effects, and visible fire and smoke effects. This distinction was useful for generating a broad database for different types of DL methods.

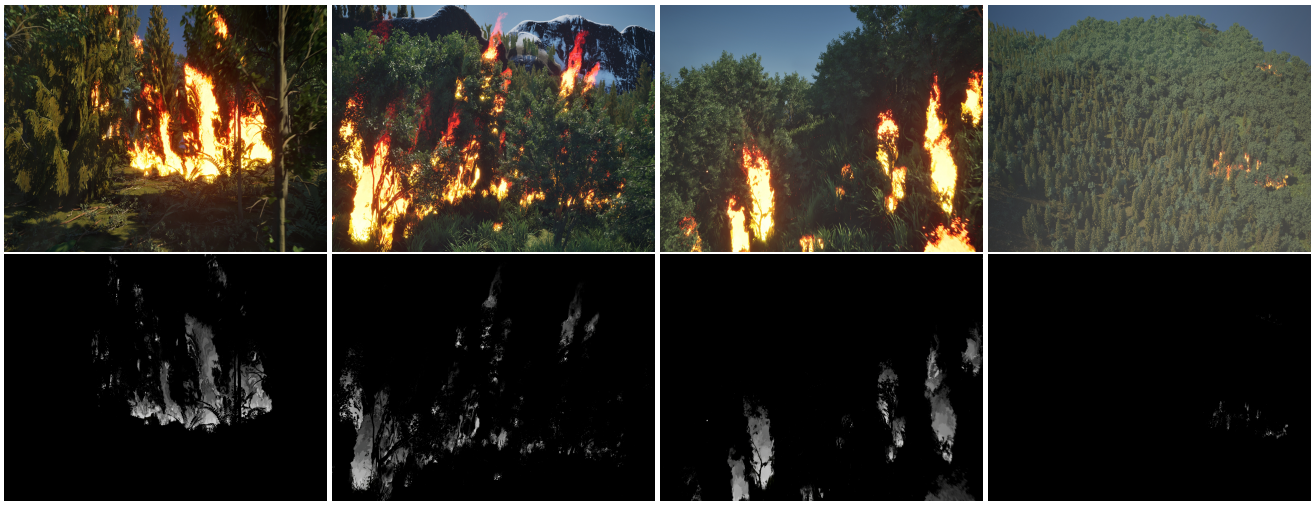

Figure 3. Fire image examples. (**Top**): RGB fire images. (**Bottom**): Their corresponding ground-truth images.

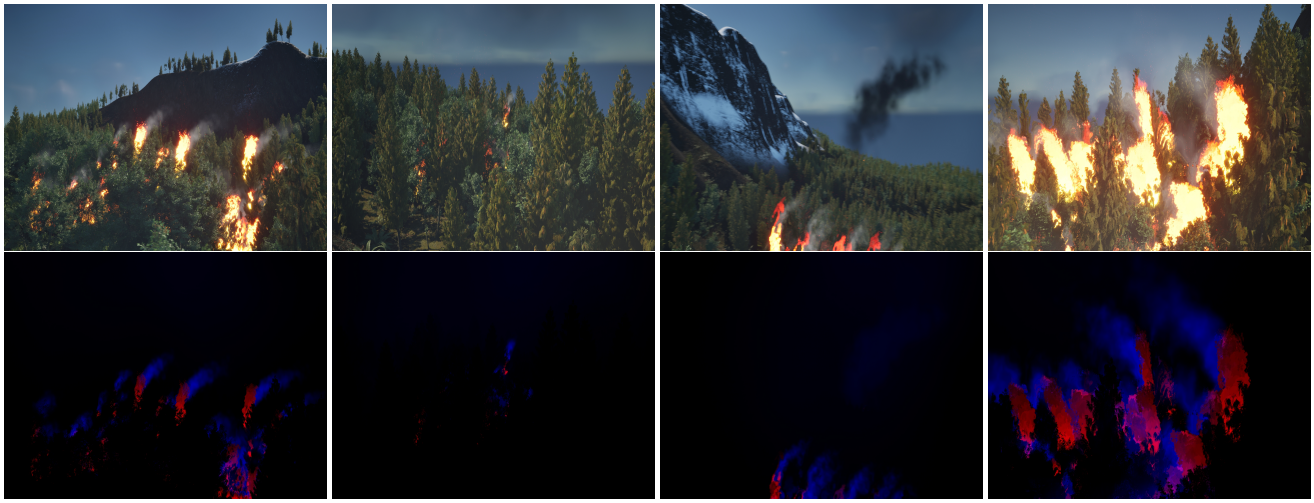

Figure 4. Fire and smoke image examples. (**Top**): RGB images. (**Bottom**): Their corresponding ground-truth images.

Table 3. SWIFT dataset overview.

Data Type	Images	Videos
Only fire	16,404	4
Only smoke	17,638	4
Fire and smoke	17,116	4
Total of fire, smoke, and both together	51,158	12
Background, no wildfire	18,048	3
Total of fire, smoke, and background	69,206	15

Besides the fire data, five simulation videos were produced. These videos showcase the simulation in effect, its behaviors, and its functionalities, allowing for a demonstration of the actual developed simulation. These simulation videos contain their simulation

parameters, such as weather conditions, wind, simulation speed, temperature, and time of day, as shown in Table 4.

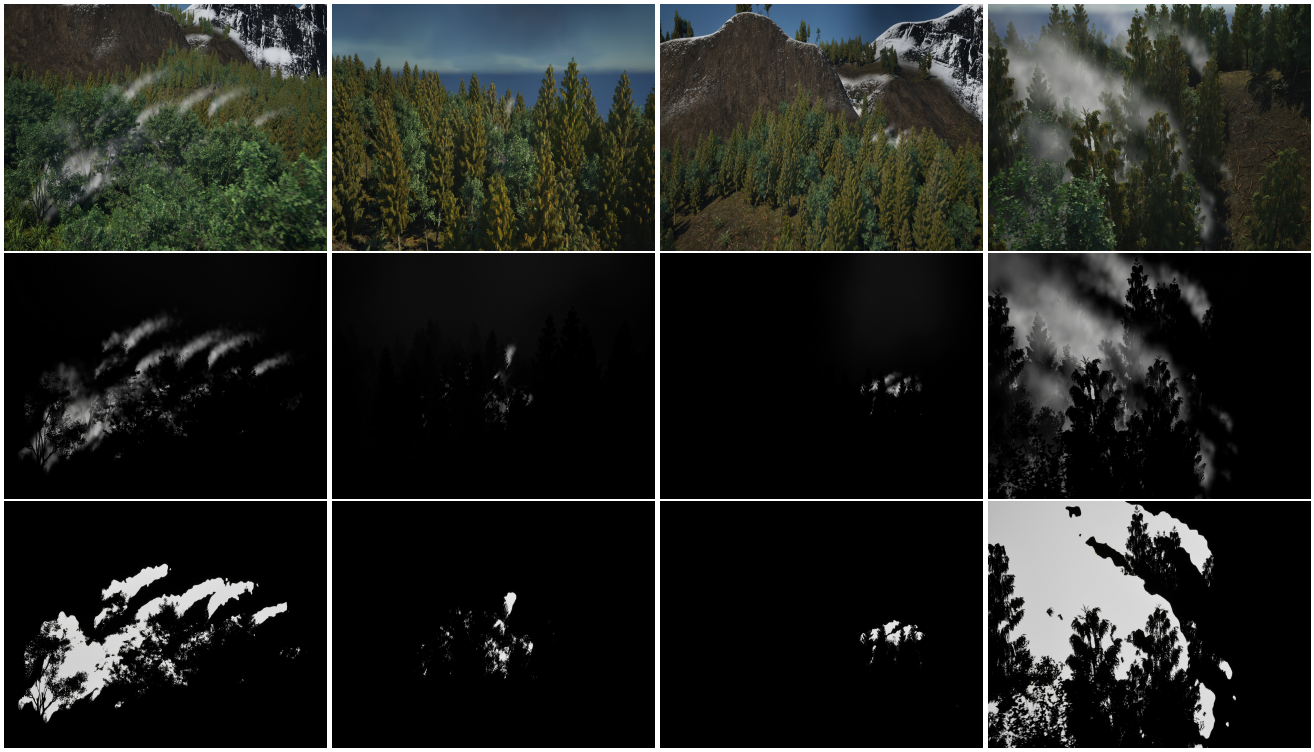

Figure 5. Smoke example images. (**Top**) to (**Bottom**): RGB smoke images, their corresponding grayscale ground truth, and their corresponding binary ground truth.

Table 4. SWIFT dataset overview.

Name	View Type	Day Time	Temperature	Humidity	Wind Speed	Wind Direction
GroundView_Fire_1	Terrestrial	1 p.m.	31 to −4 Celsius	0 to 1	32 km/H	(0,−1,0) south
GroundView_Fire_2	Terrestrial	Complete day cycle	21 Celsius	0.25	24 km/H	(0,−1,0) south
AerialView_Fire	Aerial	1 p.m.	21 Celsius	0.25	24 km/H	(0,1,0) north
InsideFire	Terrestrial	1 p.m.	21 Celsius	0.25	24 km/H	(0,−1,0) south
IndividualFire	Terrestrial	1 p.m.	21 Celsius	0.25	24 km/H	(0,1,0) north

In summary, the SWIFT dataset was generated using UE. It comprises 69,206 synthetic images and 15 videos of wildland fires with a high resolution of 1920 × 1980 pixels, depicting aerial, terrestrial, and forest interior views. This dataset is annotated and divided into four categories: fire (16,404 images and 4 videos), smoke (17,638 images and 4 videos), both fire and smoke (17,116 images and 4 videos), and no-fire/no-smoke (18,048 images and 3 videos). Each folder also includes the corresponding ground-truth images and videos. Additionally, the SWIFT dataset contains five simulation videos with their simulation parameters, including weather conditions, wind speed, humidity, wind direction, temperature, and time of day.

3.4. Proposed Methods

3.4.1. BoucaNet

BoucaNet [26] was developed to improve the smoke recognition task using satellite data. It is an ensemble learning method, incorporating two deep learning methods, EfficientFormerV2 [51] and EfficientNetV2 [52], as depicted in Figure 6. First, the input images are resized to 224 × 224 pixels. Then, these images are analyzed simultaneously by the EfficientFormerV2 and EfficientNetV2 methods to extract deep complex features and

generate two diversified feature maps. Next, the Gaussian dropout method with a rate of 0.3 is used to improve BoucaNet generalization and prevent overfitting after concatenating the two generated feature maps. Finally, a sigmoid function determines a probability score between 0 and 1, assigning the relevant class such as fire and no-fire to the input images.

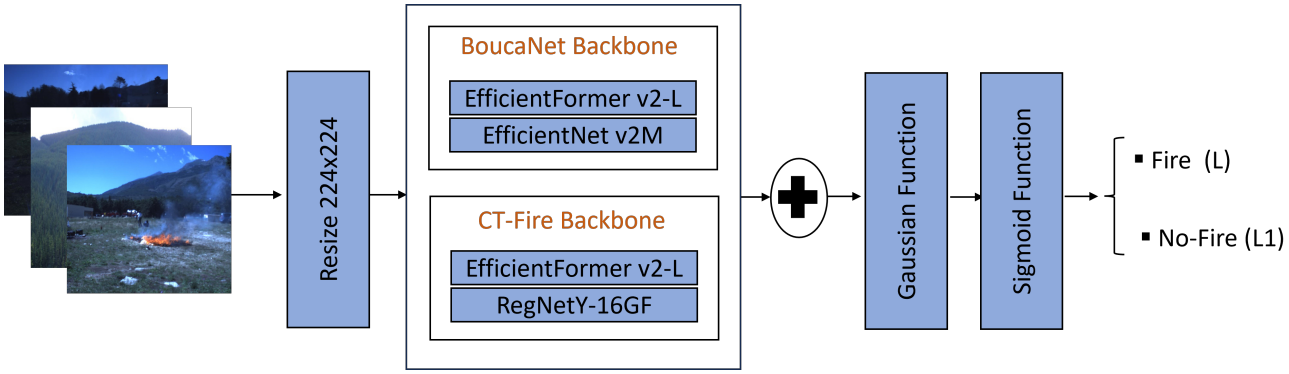

Figure 6. The proposed architecture of BoucaNet and CT-Fire methods. L and L1 refer to the likelihood of the input image being classified as fire or no-fire.

3.4.2. CT-Fire

CT-Fire [25] was introduced to improve wildfire recognition and detection. It is an ensemble learning method, combining the vision transformer EfficientFormerV2 [51] and the deep CNN RegNetY [53], as shown in Figure 6. First, the input images are resized to 224 × 224 pixels. Then, the RegNetY and EfficientFormerV2 models are adopted simultaneously for extracting wildfire features. Then, the two generated feature maps are concatenated before applying Gaussian dropout with a rate of 0.3. Finally, a sigmoid function generates CT-Fire's output, providing a probability score between 0 and 1 and indicating wildland fire presence in the input data.

3.4.3. DC-Fire

DC-Fire [27] is a deep learning method designed for detecting and identifying wildfires in infrared images. DC-Fire combines two deep CNNs, DenseNet-201 [54] and EfficientNet-B5 [55]. Firstly, as shown in Figure 7, the image, resized to 224 × 224 pixels, is fed simultaneously into these CNNs to extract relevant characteristics of wildland fires. Following the concatenation of the feature maps generated by these models, an average pooling layer is used to reduce the dimensions of the concatenated feature map. Subsequently, Gaussian dropout with a rate of 0.3 is utilized. Finally, a sigmoid function is employed to determine a probability value, ranging from 0 to 1, indicating the presence of wildfires in the input images.

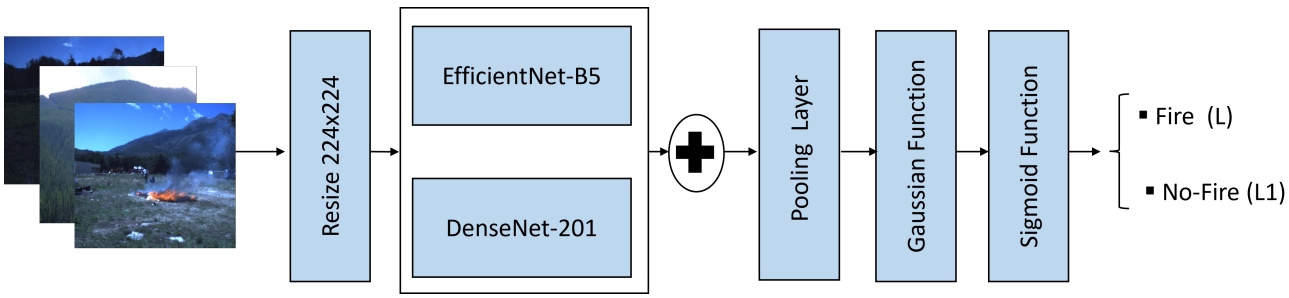

Figure 7. The proposed architecture of DC-Fire. L and L1 refer to the likelihood of the input image being classified as fire or no-fire.

3.5. Evaluation Metrics

To analyze and evaluate the performance of the proposed DL methods, we employed popular evaluation metrics, notably accuracy, precision, recall, and F1-score, used in object

classification tasks as well as in wildland fire recognition. These metrics are defined in terms of false-positive rate (FP), true-positive rate (TP), false-negative rate (FN), and true-negative rate (TN).

- Accuracy is the proportion of images correctly predicted over the total number of images, as shown in Equation (4).
- F1-score is the harmonic mean of recall and precision metrics, as given by Equation (5). Precision determines the percentage of correct predictions, meaning the number of images predicted as fire which are fire, as presented by Equation (6). Recall measures the percentage of actual fire images correctly recognized by the proposed DL models, as illustrated in Equation (7).

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{4}$$

$$\text{F1-Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{6}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{7}$$

Additionally, the inference time, which represents the mean average time required by the proposed DL model to predict the presence of wildland fire using testing data, is employed to determine the ability of these models for early detection of wildfires.

4. Results Analysis

The BoucaNet, DC-Fire, and CT-Fire models were developed using Python and TensorFlow on a machine with an Intel Xeon Gold 6148 and an NVIDIA Tesla V100SXM2 GPU. These models were trained using our proposed dataset, SWIFT. They were also tested using real wildfire images collected from the CorsicanFire [18], DeepFire [19], and Fire [56] datasets. The training data include 28,130 images, of which 13,692 are fire images and 14,438 are no-fire images. The testing data consist of 780 images, comprising 541 fire images and 239 no-fire images.

We employed a batch size of 8, 150 epochs, a learning rate of 0.001, and we input images with 224 × 224 pixels to train these models. We also adopted the loss function binary cross-entropy [57], which determines the likelihood of the presence of wildland fires using input images, as shown in Equation (8).

$$Cross - entropy = -\frac{1}{n}\sum_{i=1}^{n}(x_i \log(\hat{x}_i) + (1 - x_i)\log(1 - \hat{x}_i)) \tag{8}$$

where $\hat{x}_i$ refers to the generated output and x_i represents the label (no-fire and fire).

On the other hand, no data augmentation techniques were applied during the training stage. An early stop was implemented after 15 epochs without improvement in validation loss. The model yielding the lowest loss of validation was selected as the best-performing model.

Table 5 illustrates the parameter numbers for the proposed models, BoucaNet, DC-Fire, CT-Fire, RegNetY-16GF, and ResNeXt-101, in recognizing wildland fires. Among these, CT-Fire and BoucaNet are complex models, as each is an ensemble learning with two sophisticated DL models. CT-Fire has the highest number of parameters (109,850,670 parameters). BoucaNet and RegNetY-16GF have over 80 million parameters, significantly more than ResNeXt-101 (44,317,562 parameters) and DC-Fire (46,843,449 parameters), which integrates two simple CNNs, compared with those in the CT-Fire and BoucaNet models. Model complexity can improve modeling data ability as well as learning speed. However, it also requires more computing resources during the training step.

Table 5. Number of parameters of BoucaNet, DC-Fire, CT-Fire, and other models.

Models	Number of Parameters
BoucaNet	80,569,102
DC-Fire	46,843,449
CT-Fire	109,850,670
RegNetY-16GF	83,714,958
ResNeXt-101	44,317,562

To evaluate the performance of CT-Fire, DC-Fire, and BoucaNet, we first analyzed their testing results based on the obtained accuracy, precision, recall, F1-score, and inference time, which represents the average time taken by each model for predicting the presence of wildland fires. Additionally, these results were compared with the performance of RegNetY-16GF [53] and ResNeXt-101 [58] as baseline methods. Next, we present the F1-score values of these models for each no-fire and fire class. Then, the confusion matrix of DC-Fire, BoucaNet, and CT-Fire is introduced. Finally, examples of predicted images are presented.

Figure 8 presents the loss curves of the BoucaNet, DC-Fire, CT-Fire, RegNetY-16GF, and ResNeXt-101 models during the training and validation stages. We can see that all models showed rapid improvement during the early learning epochs, with training and validation losses rapidly decreasing and converging. This indicates that these models are well learned and generalized with no overfitting.

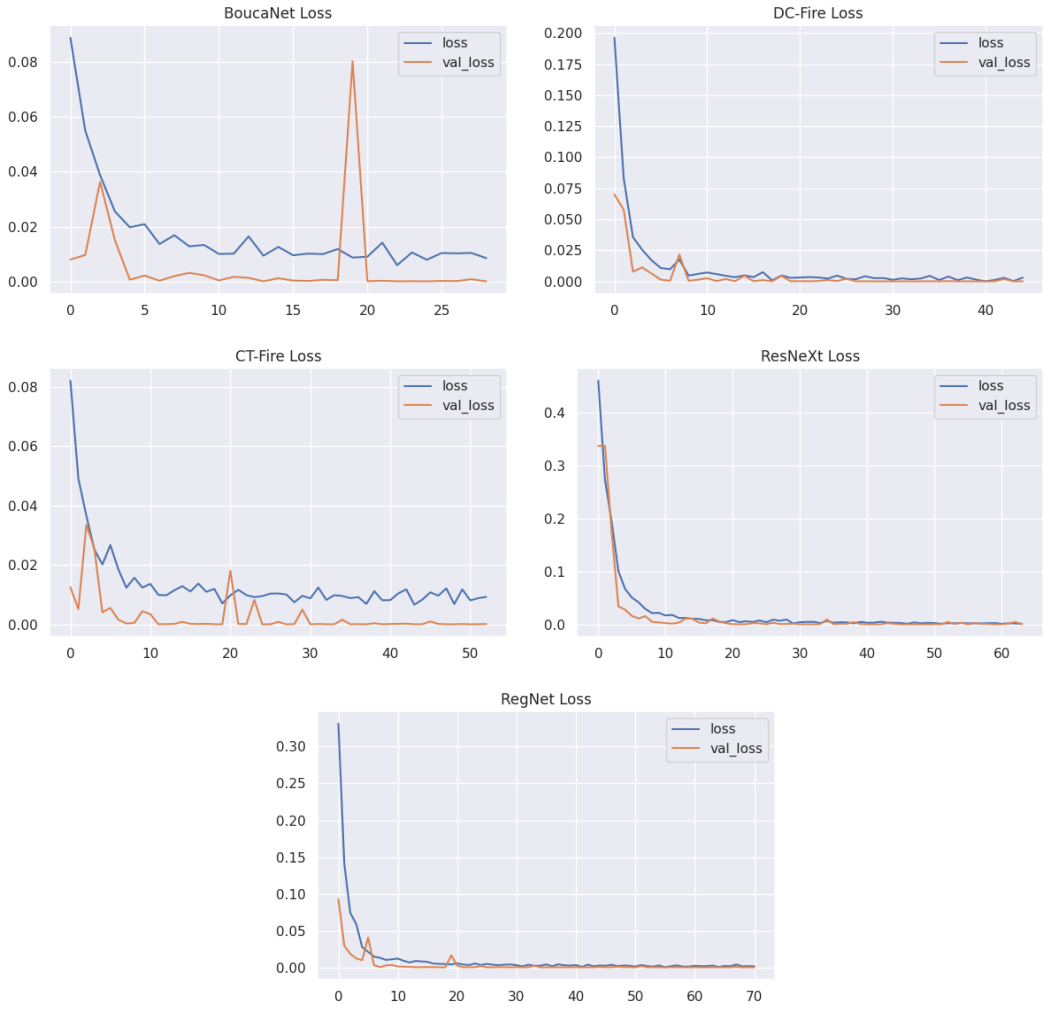

Figure 8. Loss curves for the proposed DL methods (BoucaNet, DC-Fire, CT-Fire, RegNetY-16GF, and ResNeXt-101) during training and validation steps.

Table 6 presents the performance of DC-Fire, BoucaNet, CT-Fire, RegNetY-16GF, and ResNeXt-101 using real images. Based on the accuracy and F1-score, BoucaNet performed well, with an accuracy of 93.21% and an F1-score of 93.13%, demonstrating its reliability in recognizing wildfires. CT-Fire achieved the lowest results, with an accuracy of 88.33% and an F1-score of 88.50%. In addition, BoucaNet achieved a precision of 93.17%, superior to DC-Fire, CT-Fire, RegNetY-16GF, and ResNeXt-101. This result shows the accurate ability of this model in predicting positive scenarios using real data. Moreover, BoucaNet obtained the best recall value of 93.21% among all proposed DL models, which also indicates its reliability in recognizing wildfires and solving challenges related to wildfire detection tasks. Based on the F1-score, BoucaNet slightly outperformed DC-Fire, RegNetY-16GF, and ResNeXt-101 by 0.49%, 1.89%, and 1.14%, respectively. The high performance of BoucaNet is achieved thanks to the rich and diverse feature maps generated by its backbone models, EfficientFormerV2 and EfficientNetV2. These feature maps include comprehensive global and local features, including colors, edges, shape, contrast, and textures for each fire and no-fire class. As a result, BoucaNet can differentiate wildland fires from complex background features and predict the presence of small areas of wildland fire. In addition, BoucaNet obtained a fast processing time of 0.09 s, better than the inference time of DC-Fire, CT-Fire, and RegNetY-16GF by 0.03, 0.02, and 0.01 s, respectively, enabling real-time detection. However, this inference time is higher compared with the time taken by ResNeXt-101, which is 0.03 s.

Table 6. Comparative analysis of BoucaNet, DC-Fire, CT-Fire, and other models using real images.

Models	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Inference Time (s)
BoucaNet	93.21	93.17	93.21	93.13	0.09
DC-Fire	92.82	92.96	92.82	92.64	0.12
CT-Fire	88.33	88.89	88.83	88.50	0.11
RegNetY-16GF	91.41	91.39	91.41	91.24	0.10
ResNeXt-101	92.05	91.98	92.05	91.99	0.03

Table 7 illustrates the F1-score results of BoucaNet, DC-Fire, CT-Fire, RegNetY-16GF, and ResNeXt-101 for recognizing no-fire and fire classes. BoucaNet achieved superior performance with F1-scores of 95% and 89% for fire and no-fire classes, respectively, showing its ability to distinguish between wildfires and complex backgrounds. DC-Fire reached the high F1-score of 95% for the fire class, like BoucaNet. However, it obtained an F1-score slightly lower than that of BoucaNet by 2% in recognizing the no-fire class. CT-Fire showed the lowest performance, with F1-scores of 91% and 82% for fire and no-fire classes, respectively. RegNetY-16GF and ResNeXt-101 also achieved lower results than BoucaNet for both fire and no-fire classes.

Table 7. Comparative analysis of BoucaNet, DC-Fire, CT-Fire, and other models for both fire and no-fire classes using real images.

Models	F1-Score (%)	
	Fire	No-Fire
BoucaNet	95	89
DC-Fire	95	87
CT-Fire	91	82
RegNetY-16GF	94	85
ResNeXt-101	94	87

Figure 9 depicts three confusion matrices of BoucaNet, CT-Fire, and DC-Fire for no-fire and fire classes using real data, reporting the number of true positives (fire images correctly recognized as fire), false negatives (no-fire images incorrectly classified as fire), false positives (fire images incorrectly predicted as no-fire), and true negatives (no-fire

images correctly predicted as no-fire). DC-Fire and BoucaNet achieved a high number of true positives (531 images for DC-Fire and 523 images for BoucaNet) compared to CT-Fire (482 images), indicating their ability to identify wildfires. However, they had a low number of missed wildfire instances, with a false-positive rate of 18 images for BoucaNet and 10 images for DC-Fire. On the other hand, BoucaNet and CT-Fire showed a high rate of correctly recognizing no-fire images (204 images for BoucaNet and 207 images for CT-Fire) compared to DC-Fire (193 images), demonstrating their proficiency in identifying scenarios without wildfires and overcoming false alarms.

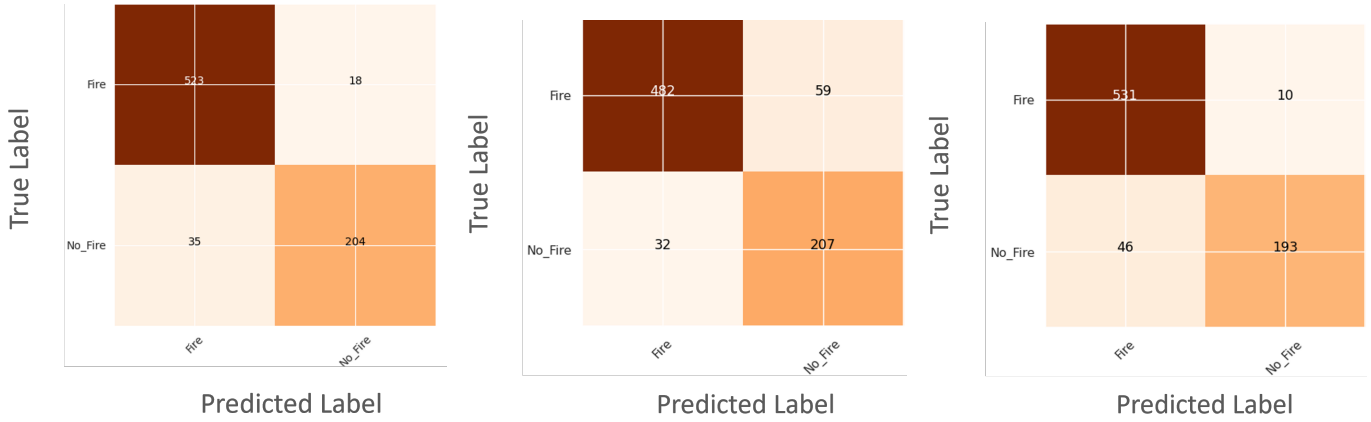

Figure 9. Confusion matrices of BoucaNet, CT-Fire, and DC-Fire using real images. From (**left**) to (**right**): BoucaNet results, CT-Fire results, and DC-Fire results.

As depicted in Figures 10 and 11, BoucaNet, DC-Fire, and ResNeXt-101 showed their accurate potential in differentiating between wildland fire and no-wildland-fire images. For example, they well predicted wildland fire images as fire images with high confidence scores of 1.00 for BoucaNet, 0.99 for DC-Fire, and 0.97 for ReseNext-101, as presented in Figure 10c. They also correctly identified small wildland fire areas (see Figure 10b) and differentiated between wildfire and no-wildfire images, which have similar features as fire, such as colors (see Figure 11b). However, CT-Fire badly predicted wildfire images, as shown in Figure 10b,c. RegNetY-16 GF also misclassified no-wildfire images as fire images (see Figure 11b).

(**a**) **Ground truth:** fire.
Predictions:
BoucaNet: fire (score: 0.99).
DC-Fire: fire (score: 0.99).
CT-Fire: fire (score: 0.84).
ResNeXt-101: fire (score: 0.97).
RegNetY-16GF: fire (score: 0.98).

(**b**) **Ground truth:** fire.
Predictions:
BoucaNet: fire (score: 0.64).
DC-Fire: fire (score: 0.82).
CT-Fire: no-fire (score: 0.88).
ResNeXt-101: fire (score: 0.97).
RegNetY-16GF: fire (score: 0.98).

(**c**) **Ground truth:** fire.
Predictions:
BoucaNet: fire (score: 1.00).
DC-Fire: fire (score: 0.99).
CT-Fire: no-fire (score: 0.99).
ResNeXt-101: fire (score: 0.97).
RegNetY-16GF: fire (score: 0.98).

Figure 10. Fire classification results of the proposed models.

(**a**) **Ground truth:** No-Fire
Predictions:
BoucaNet: No-Fire (score: 0.99)
DC-Fire: No-Fire (score: 0.99)
CT-Fire: No-Fire (score: 0.98)
ResNeXt-101: No-Fire (score: 0.74)
RegNetY-16GF: No-Fire (score: 0.62)

(**b**) **Ground truth:** No-Fire
Predictions:
BoucaNet: No-Fire (score: 0.99)
DC-Fire: No-Fire (score: 0.69)
CT-Fire: No-Fire (score: 0.99)
ResNeXt-101: No-Fire (score: 0.75)
RegNetY-16GF: Fire (score: 0.95)

(**c**) **Ground truth:** No-Fire
Predictions:
BoucaNet: No-Fire (score: 1.00)
DC-Fire: No-Fire (score: 1.00)
CT-Fire: No-Fire (score: 1.00)
ResNeXt-101: No-Fire (score: 0.97)
RegNetY-16GF: No-Fire (score: 0.98)

Figure 11. No-Fire classification results of the proposed models.

To summarize, BoucaNet effectively predicted the presence of wildland fires using real images. It outperformed the baseline models DC-Fire, CT-Fire, ResNeXt-101, and RegNetY-16GF. It also performed well in challenging scenarios, including small wildland fire zones and complex backgrounds.

5. Discussion

In this paper, recent deep learning models (BoucaNet, CT-Fire, DC-Fire, RegNetY-16GF, and ResNeXt-101) were adopted for predicting wildland fires and addressing their challenging limitations. These models were trained using a synthetic dataset, namely, SWIFT, and were tested using real fire images (sim-to-real) to address the scarcity of available annotated wildfire data. BoucaNet performed well compared to other DL models based on F1-score, precision, recall, and accuracy metrics. It achieved an accuracy of 93.21% and an F1-score of 93.13%, better than CT-Fire, DC-Fire, RegNetY-16GF, and ResNeXt-101 by 4.88%, 0.39%, 1.80%, and 1.16%, respectively, in terms of accuracy and by 4.63%, 0.49%, 1.89%, and 1.14%, respectively, in terms of F1-score. It also obtained the highest precision of 93.17% and recall of 93.21% among these models. It showed its reliability in identifying wildfires and overcoming challenging scenarios such as the complexity of the background; the varying shape, intensity, and size of wildfires; and the detection of small wildland fire zones. Additionally, these proposed models obtained an interesting inference time when detecting the presence of wildfire zones in real images. Among them, BoucaNet, DC-Fire, and ResNeXt-101 obtained inference times of 0.09 s, 0.012 s, and 0.03 s, respectively. This allows early intervention to reduce the damage of wildfires and shows the reliability of deep learning models when integrating with drone or surveillance systems to detect wildland fires and improve wildfire detection strategies.

However, these models misclassified some wildland fire instances. There are still numerous challenging scenarios due to the varying size, shape, and intensity of wildland fires, varying meteorological factors, and the visual resemblance between fire and other objects regarding their colors, such as sunrise and sunset. Hence, generating more synthetic wildfire data, including these challenging scenarios, can improve the performance of wildfire recognition tasks and enhance the potential of deep learning models in detecting wildfires, as they need big data for accurate learning.

On the other hand, while simulation platforms play a crucial role in wildland fire prevention and management, SWIFT still lacks many features expected of a true simulator, such as more realistic smoke and dynamic burn of flora and ambient environment. At present, the simulation is only able to emulate fire growth and spread associated with the environment via simplified calculations. To address these limitations, we plan to improve the generated simulation data, integrating dynamic combustion of flora and interactive simulations with real-time fire spread. In addition, we will develop wildfire scenarios in urban environments. Then, we will use SWIFT for testing and evaluating wildfire segmentation and propagation methods. This offers a comprehensive representation of wildland fire behaviors and facilitates decision making in fire management.

6. Conclusions

In this paper, a synthetic wildland fire dataset, namely, SWIFT, is presented. SWIFT was developed in UE, utilizing its newest graphical features such as TSR, Nanite, and Lumen. It includes around 70,000 synthetic images, with their corresponding mask images, generated from wildland fire scenarios. This dataset contains images that vary between camera positions, aerial and terrestrial, and differ in the classes presented for ground truth, with smoke, wildfire, both smoke and fire, and no fire. It presents numerous wildland fire scenarios collected from multiple viewpoints, including interior forest views, near active fires, ground views, and aerial views. SWIFT also includes wildfire videos with their related environmental conditions, such as temperature, humidity, wind direction, and wind speed used for wildfire simulation. Additionally, this dataset was used to train the recent DL methods BoucaNet, CT-Fire, and DC-Fire. After training, these DL models were tested against real images (sim-to-real), extracted from the CorsicanFire, DeepFire, and Fire datasets. BoucaNet showed great results with an accuracy of 93.21% and an F1-score of 93.13%, better than DC-Fire, CT-Fire, and the baseline methods RegNetY-16GF and ResNetXt-101. It showed its ability to overcome challenging scenarios, such as complex backgrounds, detecting small wildfire zones, and diverse wildland fire intensities, shapes, and sizes. The proposed DL models also demonstrated remarkable speed in detecting wildfires, as BoucaNet and DC-Fire achieved inference times of 0.09 and 0.012 s, respectively. This rapid detection demonstrates the potential of integrating deep learning into drones or surveillance systems to enhance the detection of wildland fires.

Author Contributions: Conceptualization, L.F., R.G., and M.A.A.; methodology, L.F., R.G., and M.A.A.; software, L.F. and G.R; validation, L.F., R.G., and M.A.A.; formal analysis, L.F., R.G., and M.A.A.; writing—original draft preparation, L.F. and G.R; writing—review and editing, R.G. and M.A.A.; funding acquisition, M.A.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was enabled in part by support provided by the Natural Sciences and Engineering Research Council of Canada (NSERC), funding reference number RGPIN-2018-06233.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data that support the findings of this study are available from the corresponding author upon reasonable request.

Acknowledgments: This research was enabled in part by support provided by Calcul Québec (calculquebec.ca) and the Digital Research Alliance of Canada (www.alliancecan.ca) (accessed on 1 May 2024).

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

Abbreviations

The following abbreviations are used in this manuscript:

SWIFT	Simulated Wildfire Images for Fast Training
DL	Deep learning
UE	Unreal Engine
IDE	Integrated development environment
PCG	Procedural Content Generation
CNN	Convolutional neural network
DCNN	Deep convolutional neural network
TSR	Temporal Super Resolution
ROS	Robot Operating System
SITL	Software In The Loop
CMM	Cognitive Mission Manager
LiDAR	Light detection furthermore, ranging
SHAP	SHapley Additive exPlanations
SegNet	Segmented Neural Network

References

1. Jones, M.W.; Smith, A.; Betts, R.; Canadell, J.G.; Prentice, I.C.; Quéré, C.L. Climate Change Increases the Risk of Wildfires. Available online: https://sciencebrief.org/briefs/wildfires (accessed on 12 March 2024).
2. Pechony, O.; Shindell, D.T. Driving Forces of Global Wildfires over the Past Millennium and the Forthcoming Century. *Proc. Natl. Acad. Sci. USA* **2010**, *107*, 19167–19170. [CrossRef] [PubMed]
3. Di Virgilio, G.; Evans, J.P.; Blake, S.A.P.; Armstrong, M.; Dowdy, A.J.; Sharples, J.; McRae, R. Climate Change Increases the Potential for Extreme Wildfires. *Geophys. Res. Lett.* **2019**, *46*, 8517–8526. [CrossRef]
4. Natural Resources Canada. National Wildland Fire Situation Report. Available online: https://cwfis.cfs.nrcan.gc.ca/report (accessed on 11 March 2024).
5. European Commission. 2022 Was the Second-Worst Year for Wildfires. Available online: https://ec.europa.eu/commission/presscorner/detail/en/ip_23_5951 (accessed on 11 March 2024).
6. Thomas, D.S.; Butry, D.T.; Gilbert, S.W.; Webb, D.H.; Fung, J.F. *The Costs and Losses of Wildfires*; NIST Special Publication: Washington, DC, USA, 2017; pp. 1–64. [CrossRef]
7. Reid, C.E.; Brauer, M.; Johnston, F.H.; Jerrett, M.; Balmes, J.R.; Elliott, C.T. Critical Review of Health Impacts of Wildfire Smoke Exposure. *Environ. Health Perspect.* **2016**, *124*, 1334–1343. [CrossRef] [PubMed]
8. Ghali, P.; Akhloufi, M.A. Deep Learning Approaches for Wildland Fires Using Satellite Remote Sensing Data: Detection, Mapping, and Prediction. *Fire* **2023**, *6*, 192. [CrossRef]
9. Zhao, L.; Liu, J.; Peters, S.; Li, J.; Mueller, N.; Oliver, S. Learning Class-specific Spectral Patterns to Improve Deep Learning-based Scene-level Fire Smoke Detection from Multi-spectral Satellite Imagery. *Remote Sens. Appl. Soc. Environ.* **2024**, *34*, 101152. [CrossRef]
10. Ghali, P.; Akhloufi, M.A. Deep Learning Approaches for Wildland Fires Remote Sensing: Classification, Detection, and Segmentation. *Remote Sens.* **2023**, *15*, 1821. [CrossRef]
11. Bouguettaya, A.; Zarzour, H.; Taberkit, A.M.; Kechida, A. A Review on Early Wildfire Detection from Unmanned Aerial Vehicles using Deep Learning-based Computer Vision Algorithms. *Signal Process.* **2022**, *190*, 108309. [CrossRef]
12. Chen, X.; Hopkins, B.; Wang, H.; O'Neill, L.; Afghah, F.; Razi, A.; Fulé, P.; Coen, J.; Rowell, E.; Watts, A. Wildland Fire Detection and Monitoring using a Drone-Collected RGB/IR Image Dataset. *IEEE Access* **2022**, *10*, 121301–121317. [CrossRef]
13. Jin, L.; Yu, Y.; Zhou, J.; Bai, D.; Lin, H.; Zhou, H. SWVR: A Lightweight Deep Learning Algorithm for Forest Fire Detection and Recognition. *Forests* **2024**, *15*, 204. [CrossRef]
14. Zhao, H.; Jin, J.; Liu, Y.; Guo, Y.; Shen, Y. FSDF: A High-performance Fire Detection Framework. *Expert Syst. Appl.* **2024**, *238*, 121665. [CrossRef]
15. Ghali, R.; Akhloufi, M.A. Wildfires Detection and Segmentation Using Deep CNNs and Vision Transformers. In Proceedings of the Pattern Recognition, Computer Vision, and Image Processing. ICPR 2022 International Workshops and Challenges, Montreal, QC, Canada, 21–25 August 2023; pp. 222–232.
16. Alzubaidi, L.; Bai, J.; Al-Sabaawi, A.; Santamaría, J.; Albahri, A.S.; Al-dabbagh, B.S.N.; Fadhel, M.A.; Manoufali, M.; Zhang, J.; Al-Timemy, A.H.; et al. A survey on Deep Learning Ttools Dealing with Data Scarcity: Definitions, Challenges, Solutions, Tips, and Applications. *J. Big Data* **2023**, *10*, 46. [CrossRef]
17. Chino, D.Y.T.; Avalhais, L.P.S.; Rodrigues, J.F.; Traina, A.J.M. BoWFire: Detection of Fire in Still Images by Integrating Pixel Color and Texture Analysis. In Proceedings of the 28th SIBGRAPI Conference on Graphics, Patterns and Images, Salvador, Brazil, 26–29 August 2015; pp. 95–102.

18. Toulouse, T.; Rossi, L.; Campana, A.; Celik, T.; Akhloufi, M.A. Computer Vision for Wildfire Research: An Evolving Image Dataset for Processing and Analysis. *Fire Saf. J.* **2017**, *92*, 188–194. [CrossRef]
19. Khan, A.; Hassan, B.; Khan, S.; Ahmed, R.; Abuassba, A. DeepFire: A Novel Dataset and Deep Transfer Learning Benchmark for Forest Fire Detection. *Mob. Inf. Syst.* **2022**, *2022*, 5358359. [CrossRef]
20. Shamsoshoara, A.; Afghah, F.; Razi, A.; Zheng, L.; Fulé, P.Z.; Blasch, E. Aerial Imagery Pile Burn Detection using Deep Learning: The FLAME dataset. *Comput. Netw.* **2021**, *193*, 108001. [CrossRef]
21. Taylor, S.W.; Woolford, D.G.; Dean, C.B.; Martell, D.L. Wildfire Prediction to Inform Fire Management: Statistical Science Challenges. *Stat. Sci.* **2013**, *28*, 586–615. [CrossRef]
22. Artés, T.; Oom, D.; De Rigo, D.; Durrant, T.H.; Maianti, P.; Libertà, G.; San-Miguel-Ayanz, J. A global wildfire dataset for the analysis of fire regimes and fire behaviour. *Sci. Data* **2019**, *6*, 296. [CrossRef]
23. Singh, M.; Fuenmayor, E.; Hinchy, E.P.; Qiao, Y.; Murray, N.; Devine, D. Digital Twin: Origin to Future. *Appl. Syst. Innov.* **2021**, *4*, 36. [CrossRef]
24. Koenig, L.; Nowicki, A.; Montgomery, L.N.; Lordan, D. Machine Learning within the Cognitive Mission Manger for Wildland Fire Management. In Proceedings of the AGU Fall Meeting Abstracts, Chicago, IL, USA, 12–16 December 2022; p. U35C-0532.
25. Ghali, R.; Akhlouf, M.A. CT-Fire: A CNN-Transformer for wildfire classification on ground and aerial images. *Int. J. Remote Sens.* **2023**, *44*, 7390–7415. [CrossRef]
26. Ghali, R.; Akhlouf, M.A. BoucaNet: A CNN-Transformer for Smoke Recognition on Remote Sensing Satellite Images. *Fire* **2023**, *6*, 455. [CrossRef]
27. Ghali, R.; Akhlouf, M.A. DC-Fire: A Deep Convolutional Neural Network for Wildland Fire Recognition on Aerial Infrared Images. In Proceedings of the fourth Quantitative Infrared Thermography Asian Conference (QIRT-Asia 2023), Abu Dhabi, United Arab Emirates, 30 October–3 November 2023; pp. 1–6.
28. Miller, R.K.; Field, C.B.; Mach, K.J. Barriers and Enablers for Prescribed burns for Wildfire Management in California. *Nat. Sustain.* **2020**, *3*, 101–109. [CrossRef]
29. Shah, S.; Dey, D.; Lovett, C.; Kapoor, A. AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles. In Proceedings of the Field and Service Robotics, Zurich, Switzerland, 12–15 September 2017; pp. 621–635.
30. Bhattarai, M.; Martinez-Ramon, M. A Deep Q-learning based Path Planning and Navigation System for Firefighting Environments. In Proceedings of the 13th International Conference on Agents and Artificial Intelligence, Virtual Event, 4–6 February 2021; pp. 267–277.
31. Ma, C.; Zhou, Y.; Li, Z. A New Simulation Environment Based on Airsim, ROS, and PX4 for Quadcopter Aircrafts. In Proceedings of the 6th International Conference on Control, Automation and Robotics (ICCAR), Singapore, 20–23 April 2020; pp. 486–490.
32. Stava, S.; Pirk, S.; Kratt, J.; Chen, B.; Mech, R.; Deussen, O.; Benes, B. Inverse Procedural Modelling of Trees. *Comput. Graph. Forum* **2014**, *33*, 118–131. [CrossRef]
33. Makowski, M.; Hädrich, T.; Scheffczyk, J.; Michels, D.L.; Pirk, S.; Pałubicki, W. Synthetic Silviculture: Multi-Scale Modeling of Plant Ecosystems. *ACM Trans. Graph.* **2019**, *38*, 1–14. [CrossRef]
34. Pirk, S.; Jarząbek, M.; Hädrich, T.; Michels, D.L.; Palubicki, W. Interactive Wood Combustion for Botanical Tree Models. *ACM Trans. Graph.* **2017**, *36*, 1–12. [CrossRef]
35. Hädrich, T.; Banuti, D.T.; Pałubicki, W.; Pirk, S.; Michels, D.L. Fire in Paradise: Mesoscale Simulation of Wildfires. *ACM Trans. Graph.* **2021**, *40*, 1–15. [CrossRef]
36. Niță, M.D. Testing Forestry Digital Twinning Workflow Based on Mobile LiDAR Scanner and AI Platform. *Forests* **2021**, *12*, 1576. [CrossRef]
37. Buonocore, L.; Yates, J.; Valentini, R. A Proposal for a Forest Digital Twin Framework and Its Perspectives. *Forests* **2022**, *13*, 498. [CrossRef]
38. Li, S.; Yan, Q.; Liu, P. An Efficient Fire Detection Method Based on Multiscale Feature Extraction, Implicit Deep Supervision and Channel Attention Mechanism. *IEEE Trans. Image Process.* **2020**, *29*, 8467–8475. [CrossRef] [PubMed]
39. 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. [CrossRef]
40. University of Georgia. ForestryImages Dataset. Available online: https://www.forestryimages.org/ (accessed on 12 March 2024).
41. Cazzolato, M.T.; Avalhais, L.P.; Chino, D.Y.; Ramos, J.S.; de Souza, J.A.; Rodrigues, J.F., Jr.; Traina, A. Fismo: A Compilation of Datasets From Emergency Situations for Fire and Smoke Analysis. In Proceedings of the Brazilian symposium on databases-SBBD, Uberlandia, Brazil, 2–5 October 2017; pp. 213–223.
42. Flickr Team. Flickr-FireSmoke and Flickr-Fire Datasets. Available online: https://www.flickr.com/ (accessed on 12 March 2024).
43. Cazzolato, M.T.; Bedo, M.V.N.; Costa, A.F.; de Souza, J.A.; Traina, C.; Rodrigues, J.F.; Traina, A.J.M. Unveiling Smoke in Social Images with the SmokeBlock Approach. In Proceedings of the 31st Annual ACM Symposium on Applied Computing, Pisa, Italy, 4–8 April 2016; pp. 49–54.
44. Treneska, S.; Stojkoska, B.R. Wildfire Detection from UAV Collected Images Using Transfer Learning. In Proceedings of the 18th International Conference on Informatics and Information Technologies, Xi'an, China, 12–14 March 2021; pp. 6–7.
45. Ghali, R.; Akhloufi, M.A.; Mseddi, W.S. Deep Learning and Transformer Approaches for UAV-Based Wildfire Detection and Segmentation. *Sensors* **2022**, *22*, 1977. [CrossRef]

46. Zhang, L.; Wang, M.; Fu, Y.; Ding, Y. A Forest Fire Recognition Method Using UAV Images Based on Transfer Learning. *Forests* **2022**, *13*, 975. [CrossRef]
47. Ahmad, K.; Khan, M.S.; Ahmed, F.; Driss, M.; Boulila, W.; Alazeb, A.; Alsulami, M.; Alshehri, M.S.; Ghadi, Y.Y.; Ahmad, J. FireXnet: An Explainable AI-based Tailored Deep Learning Model for Wildfire Detection on Resource-constrained Devices. *Fire Ecol.* **2023**, *19*, 54. [CrossRef]
48. Wang, L.; Zhang, H.; Zhang, Y.; Hu, K.; An, K. A Deep Learning-Based Experiment on Forest Wildfire Detection in Machine Vision Course. *IEEE Access* **2023**, *11*, 32671–32681. [CrossRef]
49. Jonnalagadda, A.V.; Hashim, H.A. SegNet: A segmented Deep Learning Based Convolutional Neural Network Approach for Drones Wildfire Detection. *Remote Sens. Appl. Soc. Environ.* **2024**, *34*, 101181. [CrossRef]
50. Papadopoulos, G.D.; Pavlidou, F.N. A Comparative Review on Wildfire Simulators. *IEEE Syst. J.* **2011**, *5*, 233–243. [CrossRef]
51. Li, Y.; Hu, J.; Wen, Y.; Evangelidis, G.; Salahi, K.; Wang, Y.; Tulyakov, S.; Ren, J. Rethinking Vision Transformers for MobileNet Size and Speed. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 2–6 October 2023; pp. 16889–16900.
52. Tan, M.; Le, Q.V. EfficientNetV2: Smaller Models and Faster Training. In Proceedings of the 38th International Conference on Machine Learning, Virtual Event, 18–24 July 2021; pp. 10096–10106.
53. Radosavovic, I.; Kosaraju, R.P.; Girshick, R.; He, K.; Dollár, P. Designing Network Design Spaces. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, DC, USA, 14–19 June 2020; pp. 10428–10436.
54. Gao, Z.; Laurens, V.D.M.; Weinberger, K.Q. Densely Connected Convolutional Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
55. Tan, M.; Le, Q.V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In Proceedings of the 36th International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 6105–6114.
56. Saied, A. Fire Dataset. Available online: https://www.kaggle.com/datasets/phylake1337/fire-dataset?select=fire_dataset%2C+06.11.2021 (accessed on 12 March 2024).
57. Al-Dabbagh, A.M.; Ilyas, M. Uni-temporal Sentinel-2 Imagery for Wildfire Detection Using Deep Learning Semantic Segmentation Models. *Geomat. Nat. Hazards Risk* **2023**, *14*, 2196370. [CrossRef]
58. Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; He, K. Aggregated Residual Transformations for Deep Neural Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1492–1500.