

Forest Fire Alert System : Fire detection using Deep Learning: the FLAME dataset

Vaibhav Kharatmal, Selvaganpati M , Shristhi Mondal

*PdamaBhusamVasantdada Patil Pratishthan College of Engineering,
Mumbai, India*

Abstract

Wildfires are one of the costliest and deadliest natural disasters in the US, causing damage to millions of hectares of forest resources and threatening the lives of people and animals. Of particular importance are risks to firefighters and operational forces, which highlights the need for leveraging technology to minimize danger to people and property. *FLAME (Fire Luminosity Airborne-based Machine learning Evaluation)* offers a dataset of aerial images of fires along with methods for fire detection and segmentation which can help firefighters and researchers to develop optimal fire management strategies.

This paper provides a fire image dataset collected by drones during a prescribed burning piled detritus in an Arizona pine forest. The dataset includes video recordings and thermal heatmaps captured by infrared cameras. The captured videos and images are annotated, and labeled frame-wise to help researchers easily apply their fire detection and modeling algorithms. The paper also highlights solutions to two machine learning problems: (1) Binary classification of video frames based on the presence [and absence] of fire flames. An Artificial Neural Network (ANN) method is developed that achieved a 76% classification accuracy. (2) Fire detection using segmentation methods to precisely determine fire borders. A deep learning method is designed based on the U-Net

*Corresponding author
Email address: vu1s1819062@pvppcoe.ac.in (Vaibhav Kharatmal)

up-sampling and down-sampling approach to extract a fire mask from the video frames. Our FLAME method approached a precision of 92%, and recall of 84%. Future research will expand the technique for free burning broadcast fire using thermal images.

Keywords: Aerial imaging, fire monitoring dataset, fire detection and segmentation, deep learning.

1. Introduction: Scope, significance, and problem definition

Wildfires have caused severe damage to forests, wildlife habitats, farms, residential areas, and ecosystems during the past few years. Based on the reports from National Interagency Fire Center (NIFC) in the USA, total number of 51,296 fires burned more than 6,359,641 acres of lands yearly on average from 2010 to 2019 accounting for more than \$6 billion in damages [1, 2]. These alarming facts motivate researchers to seek novel solutions for early fire detection and management. In particular, recent advances in aerial monitoring systems can provide first responders and operational forces with more accurate data on fire behaviour for enhanced fire management.

Traditional approaches to detecting and monitoring fires include stationing personnel in lookout towers or using helicopters or fixed-wing aircraft to surveil fires with visual and infrared imaging. Recent research has suggested Internet of Things (IoT) innovations based on wireless sensor networks [3, 4, 5, 6, 7], but such networks would require further investment and testing before providing practical information. At broader scales, satellite imagery is widely used for assessing fires globally [8, 9], but typically at relatively coarse resolution and with the availability of repeat images constrained by satellite orbital patterns.

Considering the challenges and issues of these methods, using Unmanned Aerial Vehicles (UAVs) for fire monitoring is gaining more traction in recent years [10, 11, 12]. UAVs offer new features and convenience including fast deployment, high maneuverability, wider and adjustable viewpoints, and less human intervention [13, 14, 15, 16, 17]. Recent studies investigated the use of

UAVs in disaster relief scenarios and operations such as wildfires and floods, particularly as a temporary solution when terrestrial networks fail due to damaged infrastructures, communication problems, spectrum scarcity, or coalition formation [18, 19, 20, 21, 22].

Recent advances in artificial intelligence (AI) and machine learning have made image-based modeling and analysis (e.g., classification, real time prediction, and image segmentation) even more successful in different applications [23, 24, 25, 26]. Also, with the advent of nanotechnology semiconductors, a new generation of Tensor Processing Units (TPUs) and Graphical Processing Units (GPUs) can provide an extraordinary computation capability for data-driven methods [27]. Moreover, modern drones and UAVs can be equipped with tiny edge TPU/GPU platforms to perform on-board processing on the fly to facilitate early fire detection before a catastrophic event happens [28, 29].

Most supervised learning methods rely on large training datasets to train a reasonably accurate model. Studies such as [30] used a fire dataset from public sources to perform fire detection based on pre-trained ANN architectures such as MobileNet and AlexNet. However, that dataset was based on terrestrial images of the fire. To the best of our knowledge, there exists no aerial imaging dataset for fire analysis, something in urgent need to develop fire modeling and analysis tools for aerial monitoring systems. Note that aerial imagery exhibits different properties such as low resolutions, and top-view perspective, substantially different than images taken by ground cameras.

In this paper, we introduce a new dataset as a collection of fire videos and images taken by drones during a prescribed burning slash piles in Northern Arizona. The images were taken by multiple drones with different points of view, different zoom, and camera types including regular and thermal cameras. Pile burns can be very helpful to study spot fires and early-stage fires. Pile burns are typically used by forest management for cleaning up forest residues (“slash”) such as branches and foliage from forest thinning and restoration projects. Forest treatments are a key management strategy for reducing fuels and the burning of slash piles is often the most economically efficient and safe means of removing

slash. Piles must be monitored by fire managers for a few days after ignition to avoid spread outside the intended burn area. Using automated aerial monitoring systems can substantially reduce the forest management workload.


We propose two sample problems to evaluate the use of dataset for real-world fire management problems. The contributions of this paper include i) proposing the first of its kind aerial imaging dataset for pile burn monitoring which includes both normal and thermal palettes as well as FLAME (Fire Luminosity Airborne-based Machine learning Evaluation), ii) a DL-based algorithm for frame-based fire classification which can be used for early fire detection, and iii) a DL-based image segmentation method for pixel-wise fire masking for fire expansion modeling. The rest of the paper is structured as follows. Section 2 presents the FLAME dataset along with the related information regarding the hardware and data. Section 3 discusses the methodology based on the two defined challenges, namely fire classification and fire segmentation. The experiments and results are illustrated in Section 4 over a variety of metrics. Conclusions and discussion points are provided in Section 6.

2. FLAME Dataset: Hardware and Applicable Data

This section details the hardware used to collect information, the data modalities, and types of the captured information.

Prescribed burning of slash piles is a common occurrence primarily during the winter months in high-elevation forests of the Southwest. Prescribed fires provide excellent opportunities for researchers to collect and update imagery data. The current study shows the results of the first test, and from which is available to continually update the dataset by adding more test results. The test was conducted with fire managers from the Flagstaff (Arizona) Fire Department who carried out a burn of piled slash on city-owned lands in a ponderosa pine forest on Observatory Mesa. The prescribed fire took place on January 16th, 2020 with the temperature of 43°F ($\sim 6^{\circ}\text{C}$) and partly cloudy conditions and no wind.

Table 1: Technical specification of hardware and tools

	<p>Phantom 3 Professional, DJI, 1280 gram, diagonal size=350mm,</p> <p>Max speed=16m/s(\sim57kph), max flight time is 23 minutes,</p> <p>Flight time is reduced to 18 mins due to additional weight [31].</p>
	<p>Matrice 200, DJI, 3.80kg, size:716mm \times 220mm \times 236mm,</p> <p>payload up to 2kg, 16m/s (\sim61kph), batteries: (TB50) and TB55</p> <p>Max flight time: 38 minutes, operation range of 7km [32].</p>
	<p>Zenmuse X4S, DJI, gimbal: Matrice 200, weight: 253 gram,</p> <p>Field Of View (FOV): 84°, resolution: Full HD to Cinematic 4K</p> <p>sensor: CMOS 20MPixels [33].</p>
	<p>Vue Pro R, FLIR, IR camera, control: Bluetooth and</p> <p>Pulse Width Modulation (PWM) signal, FOV: 45°, resolution: 640 \times 512</p> <p>Lens: 6.8mm thermal, no gimbal [34].</p>
	<p>Phantom 3 camera, DJI, sensor: 1/2.3" CMOS 12.4MPixels</p> <p>FOV: 94°, (FPSs): 24 to 60, resolution: HD, FHD, UHD [31].</p>

2.1. Hardware

This study utilizes different drones and cameras to create a dataset of fire aerial images. Table 1 describes the technical specification of the utilized drones and cameras.

2.2. Applicable data for the defined problem

This section presents the details of the captured images, videos. The captured videos are converted to frames based on the recorded Frames Per Second (FPS). Four types of video including the normal spectrum, fusion, white-hot, and green-hot palettes are available in the FLAME dataset [35].



Figure 1: Frame samples of the normal spectrum palette.

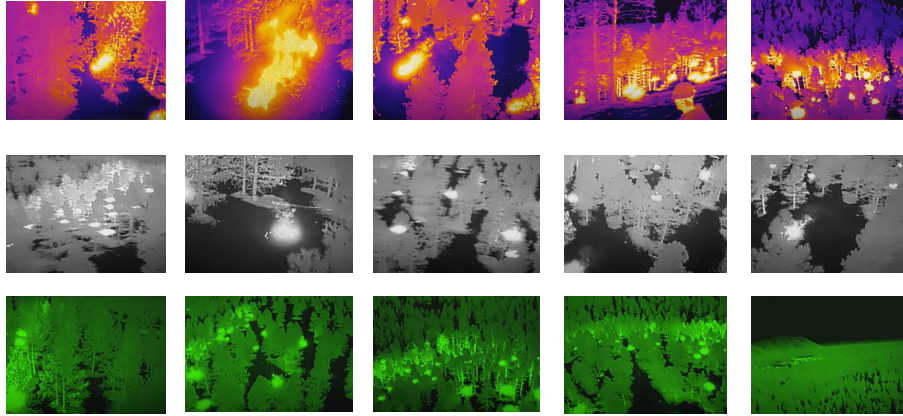


Figure 2: Frame samples of thermal images including Fusion, WhiteHot, and GreenHot palettes from top row to the bottom row.

The normal spectrum palette was recorded using both Zenmuse X4S and the phantom 3 camera. Other thermal and IR outputs were collected using the Forward Looking Infrared (FLIR) vue Pro R camera. Several video clips which include both fire and no fire footage are available. The FLIR camera has a 1280×720 resolution with 29 frame per seconds (FPS). Another 6 minutes of video is available for one pile burn from the start of the burning at 1280×720 resolution and 29 FPS. The H.264 codec was used for all the recordings. More details about these videos are available in Table 2 along with the dataset link. Figure 1 demonstrates some representative frames from both fire and no-fire

Table 2: Dataset information ([Link](#)) [35].

	Type	Camera	Palette	Duration	Resolution	FPS	Size	Application	Usage	Labeled
1	Video	Zenmuse	Normal(.MP4)	966 seconds	1280×720	29	1.2 GB	Classification	-	N
2	Video	Zenmuse	Normal(.MP4)	399 seconds	1280×720	29	503 MB	-	-	N
3	Video	FLIR	WhiteHot(.MOV)	89 seconds	640×512	30	45 MB	-	-	N
4	Video	FLIR	GreenHot(.MOV)	305 seconds	640×512	30	153 MB	-	-	N
5	Video	FLIR	Fusion(.MOV)	25 mins	640×512	30	2.83 GB	-	-	N
6	Video	Phantom	Normal(.MOV)	17 mins	3840×2160	30	32 GB	-	-	N
7	Frame	Zenmuse	Normal(.JPEG)	39,375 frames	254×254	-	1.3 GB	Classification	Train/Val	Y
8	Frame	Phantom	Normal(.JPEG)	8,617 frames	254×254	-	301 MB	Classification	Test	Y
9	Frame	Phantom	Normal(.JPEG)	2,003 frames	3480×2160	-	5.3 GB	Segmentation	Train/Val/Test	Y(Fire)
10	Mask	-	Binary(.PNG)	2,003 frames	3480×2160	-	23.4 MB	Segmentation	Train/Val/Test	Y(Fire)

videos. The full videos are available in the FLAME dataset repository.

The FLAME dataset also includes thermal videos such as Fusion, WhiteHot, and GreenHot palettes. All videos were captured with the resolution of 640×512 and with 30 FPS. Multiple videos of fire and no-fire types with different lengths are available. Figure 2 shows some randomly selected frames for these thermal videos. More details about the FLAME dataset are available in Table 2. Also, a sample video of this dataset is available on YouTube [36]. Sections 3.1 and 3.2 demonstrate some of the videos conversions into frames to address research challenges such as fire classification and fire segmentation. Researchers can use applications of their choice to extract the frames from the videos based on the required FPS. The FLAME dataset including all images, videos, and data are available on IEEE-Dataport [35].

3. Goals: Suggested Experiments and Methodology

This section presents two example applications that can be defined based on the collected FLAME dataset along with Deep Learning solutions for these problems. The first problem is the fire versus no-fire classification using a deep neural network (DNN) approach. The second problem deals with fire segmentation, which can be used for fire detection by masking the identified fire regions on video frames classified as fire-containing in the first problem.

3.1. Fire vs No-Fire Classification

The image classification problem is one of the challenging tasks in the image processing domain. In the past, traditional image processing techniques utilized RGB channel comparison to detect different objects such as fire in frames or videos [37, 38, 39]. These traditional methods are not free of errors and are not fully reliable [40]. For instance, RGB value comparison methods that usually consider a threshold value to detect fire may detect sunset and sunrise as a false positive outcome. However, training a DNN to perform this image classification task helps to learn elements not germane to the fire. Also, some studies such as [41, 42] perform pixel-based classification and segmentation based on the HSV (Hue, Saturation, Value) format. In the present study, a supervised machine learning method is used to classify the captured frames from camera. For mixed images when fire and non-fire parts coexist, the frame will be considered as the fire-labeled frame and when there is no fire in the frame, it will be considered as non-fire-labeled. Instead of the green or fusion heat map, the normal range spectrum of images for the classification was selected using the Zenmuse X4S and the camera from DJI Phantom 3. The binary classification model which was used in this study is the Xception network [43] proposed by Google-Keras¹. The Xception model is a deep Convolutional Neural Network (DCNN). The structure of the DCNN is shown in Fig. 3. Replacing the standard *Inception*

¹https://keras.io/examples/vision/image_classification_from_scratch/

modules of the Inception architecture with depth-wise separable convolutions resulted in the Xception network [43, 44, 45].

Figure 3 is the concise version of the Xception model. The Xception model has three main blocks: 1) the input layer, 2) the hidden layers, and 3) the output layer. The size of the input layer depends on the image size and the number of channels which in our case is $(254 \times 254 \times 3)$. Then the value of RGBs in different channels are scaled to a float number between 0 and 1.0. The hidden layers rely on depth-wise separable convolutions and shortcut between the convolution blocks (ResNet [46]). The entry flow of the hidden layers is a pair of 2-Dimensional (2D) convolutional blocks with a size of 8 and a stride of 2×2 . Each block follows a batch normalization and a Rectified Linear Unit (ReLU) activation function [47]. The batch normalization is used to speed up the training process and bring more randomness by decreasing the importance of initial weights and regularize the model. Next, the model follows two separable 2D convolutional blocks. The last block of the hidden layer is a separable 2D convolutional layer with a size of 8 followed by another batch normalization and the ReLU function. Since the fire-detection is a binary classification task (Fire/No Fire), the activation function for the output layer is a Sigmoid function. The equation for the Sigmoid function is shown in (1),

$$P(\text{label=Fire}) = \sigma(\text{label=Fire}|\zeta(\theta)) = \frac{1}{1 + e^{-\zeta(\theta)}}, \quad (1)$$

where $\zeta(\theta)$ is the value of the output layer which is extracted based on the input frames and the RGB values of each pixel and all the weights across the hidden network. θ is the weight for the last layer of the network. The output value of the Sigmoid function is the probability of fire detection based on the imported frames into the network. To train the Xception network and find the weights of all neurons, a value loss function is targeted to increase the accuracy of the networks and find the optimal values for the weights. As the problem in this section is a binary classification, the considered loss function is a binary

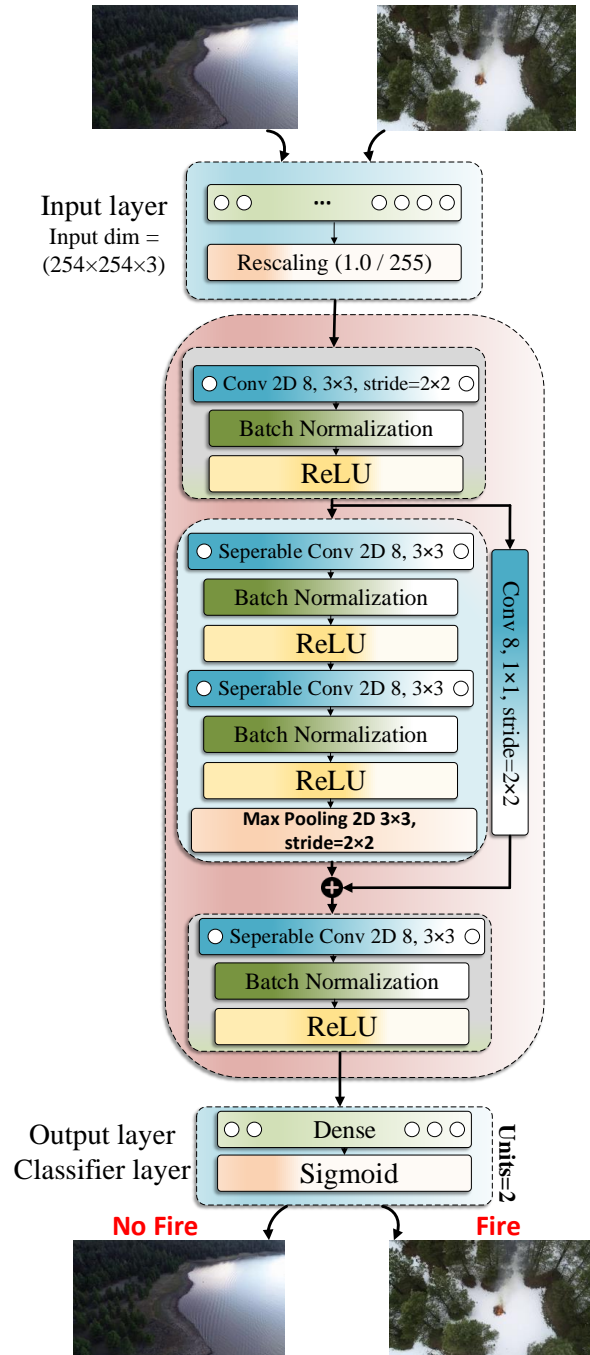


Figure 3: Small version of the Xception network for the fire classification.

cross-entropy defined as

$$\begin{aligned} \mathcal{L}(y, \hat{y}) = & \\ & - \frac{1}{N} \sum_{i=1}^N (y_i * \log(p(\hat{y}_i)) + (1 - y_i) * \log(1 - p(\hat{y}_i)), \end{aligned} \quad (2)$$

where N is the number of total samples in each batch used to update the loss function for each epoch. y is the ground truth label for the frames of types fire ($y = 1$) and no/fire ($y = 0$) based on the training data. $p(\hat{y})$ is the predicted probability of a frame belonging to the fire class. Next, the Adam optimizer is used to minimize the loss function and find the optimal weights during the learning process. After training the network with the training dataset, the evaluation is performed using a test dataset in Section 4. The implemented code for this learning model is available on GitHub [48].

3.2. Fire Segmentation

This section considers the problem of image segmentation for frames labeled as "fire" by the fire classification algorithm presented in section 3.1. Studying the fire segmentation problem is useful for scenarios like detecting small fires [49]. Also, fire segmentation helps fire managers localize different discrete places of active burning for the purpose of fire monitoring. The goal is to propose an algorithm to find the pile burn segments in each frame and generate relevant masks. These segmentation problems were handled differently in the past using image processing and RGB threshold values to segment different data batches which exhibits relatively high error rates [50, 40, 51]. The goal is to develop an image semantic segmentation to perform a pixel-wise classification for each frame at the pixel level to define a fire mask for the generated output. To accomplish this task, a DCNN model is implemented to predict the label of each pixel based on the imported data. This segmentation problem can be recast as a binary pixel-wise classification problem, where each pixel can take two labels: "fire" and "non-fire" (background). To accomplish the image segmentation task, the fire test dataset from Section 3.1 is considered as a training dataset. To train

a DCNN model, a Ground Truth Mask dataset is required. Different tools and applications such as Labelbox [52], Django Labeller [53], LabelImg [54], MATLAB Image Labeler [55], GNU Image Manipulation Program (GIMP) [56], etc are available to perform different types of the manual image segmentation such as pixel labeling, annotation (rectangles, lines, and cuboid) on the Regions Of Interest (ROI) to provide training data for the utilized deep learning model. The MATLAB (TM) Image Labeler is used on 2003 frames to generate the Ground Truth Masks. This subcategory of the FLAME dataset of masks and images is presented in Table 2. The implemented image segmentation model is adopted from the U-Net convolutional network developed for biomedical image segmentation [57]. U-Net is an end-to-end technique between the raw images and the segmented masks. A few changes are made to this network to accommodate the FLAME dataset and adapt it to the nature of this problem. The ReLU activation function is changed to Exponential Linear Unit (ELU) of each two-dimensional convolutional layer to obtain more accurate results [58]. The ELU function has a negative outcome smaller than a constant α for the negative input values and it exhibits a smoother behavior than the ReLU function. The structure of the customized U-Net is shown in Figure 4. The backbone of the U-Net consists of a sequence of up-convolutions and concatenation with high-resolution features from the contracting path.

The size of the input layer is $512 \times 512 \times 3$ designed to match the size of the inputs images and three RGB channels. For computational convenience, the RGB values (between 0 and 255) are scaled down by 255 to yield float values between 0 and 1. Next, it follows the first contracting block including a two-dimensional fully convolutional layers with the ELU activation function, a dropout layer, another same fully convolutional layer, and a two-dimensional max pooling layer. This structure is repeated another three times to shape the left side of the U shape. Next, there are two two-dimensional fully connected layers with a dropout layer in between, the same structure of the left side is repeated for the right side of the U shape to have a symmetric structure for the up-convolution path in each block. Also, there exists a concatenation between

the current block and the peer block from the contracting path. Since the pixel-wise segmentation is a binary classification problem, the last layer has the Sigmoid activation function.

The DCNN utilizes a dropout method to avoid the overfitting issue in the FLAME dataset analysis and realize a more efficient regularization noting the small number of ground truth data samples. The utilized loss function is the binary cross entropy similar to (2). The Adam optimizer is used to find the optimal value of weights for the neurons. The evaluation of the FLAME-trained model with the ground truth data is described in Section 4.2. The implemented code for this section is available on GitHub [48].

4. Results: Metrics and guidance on reporting results

In this section, we present the results of the two different problems of fire classification and fire segmentation. First, we provide the details of the parameters used in our experiments. Next, we discuss the results of each algorithm. All simulations for the training, validation, and testing phases, are performed using a AMD Ryzen 9 3900X with NVidia RTX 2080 Ti on an Ubuntu system.

4.1. Fire vs No-Fire Classification

In the training section, the total number of the frames is 39,375 which includes 25,018 frames of type "fire" and 14,357 frames of type "non-fire". The training dataset is further split to 80% training and 20% validation sets. All frames are shuffled before feeding into the network. Also, augmentation methods such as horizontal flipping and random rotation are used to create new frames and address the issue of bias for unbalanced number of samples in the two "fire" and "non-fire" classes. The training phase ran over 40 epochs and the learning rate for the Adam optimizer is set to 0.001 which remains fixed during the training phase. Also, the batch size of 32 is used to fit the model in the training phase. To evaluate the accuracy and loss on the test dataset, 8,617 frames including 5,137 fire-labeled frames and 3,480 No-fire-labeled frames are

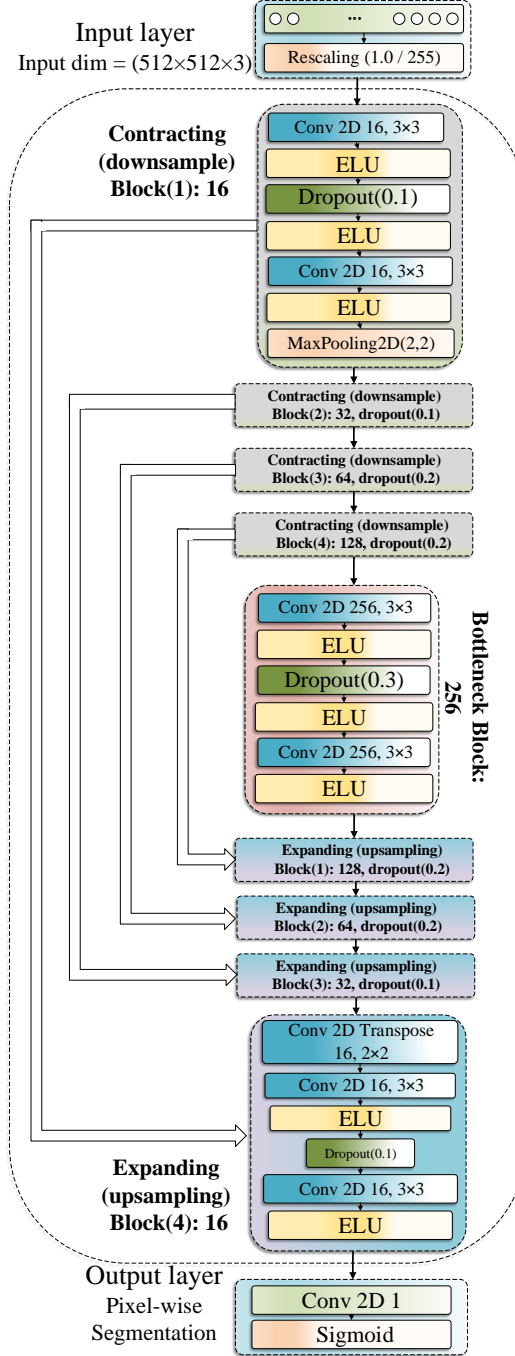


Figure 4: Customized version of the U-Net for the fire segmentation.

Table 3: Accuracy and loss for evaluation of the fire classification.

	Performance	
Dataset	Loss	Accuracy(%)
Test set	0.7414	76.23
Validation set	0.1506	94.31
Training set	0.0857	96.79

fed into the pre-trained networks. Table 3 reports loss and accuracy on training, validation, and test sets. It is noteworthy to mention that all frames for the training phase are collected using the Matrice 200 drone using Zenmuse X4S camera and all frames for the test set are collected using the Phantom drone and its default mounted camera. Therefore, no overlap exists between the training and test samples. This fact confirms that our method is not biased to the imaging equipment properties, and the actual accuracy would be even higher when using the same imaging conditions for the training and test phase. The achieved accuracy of the “Fire vs No-Fire” classification is 76.23%.

Figure 5 demonstrates the loss and accuracy for the training phase for both the training and validation sets. Also, Figure 6 presents the confusion matrix for this binary fire classification task for all predictions. The vertical axis shows the true label of frames and the horizontal axis expresses the predicted label. The confusion matrix considers two classes which is plotted for the test dataset. Since, the ratio of fire and No-fire frames was imbalanced at the training phase, the rate of the false positive (classifying a true no-fire as fire) is higher than the false negative rate (classifying a true fire as no-fire).

4.2. Fire Segmentation

The purpose of fire segmentation is to accurately localize and extract the fire regions from the background. Therefore, the video frames within the test set

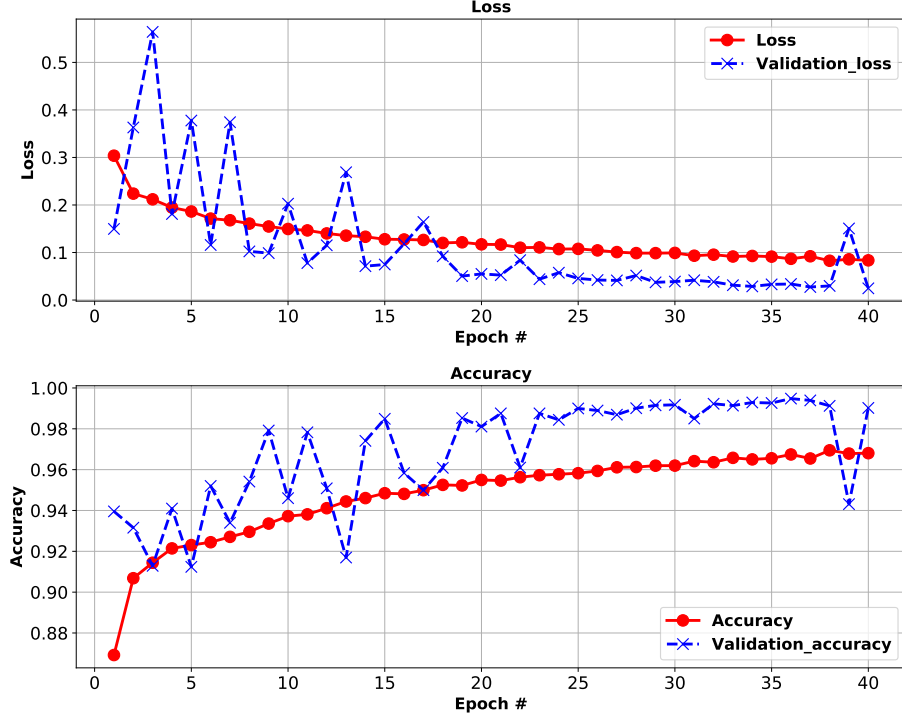


Figure 5: Accuracy and loss values for the training and validation sets.

which are labeled as "fire" by the fire classification stage in section (Section 3.1) are used here for training, validation, and test. The total number of frames is 5,137 and 2003 masks generated using the MATLAB (TM) Image Labeler tool for the ground truth data. The ground truth masks and data are generated based on the human subject matter expert (SME) eye efficiency to mark the fire pixels using manual polygon shape in MATLAB (TM) Image Labeler. The split ratio between the training and validation data is 85% and 15%. The frames and ground truth data were shuffled accordingly before importing into the training model. The maximum number of epochs is 30; however, an early stop callback was considered when the performance does not substantially change. The batch size for the training is 16. Figure 7 demonstrates six samples of the test set along with the expected ground truth masks and the generated masks from the trained network. The first row is the input frame to the model, the second

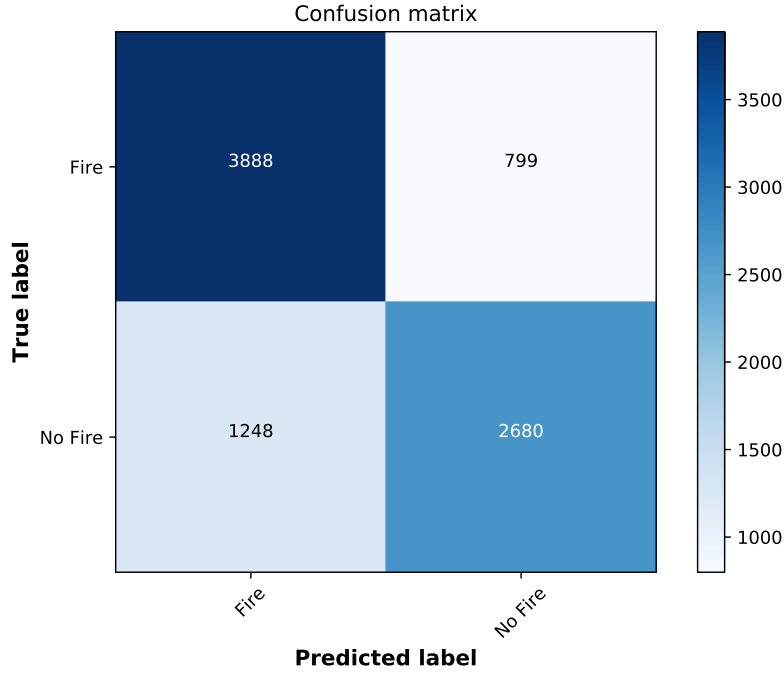


Figure 6: Confusion matrix for the true and predicted labels.

row is the ground truth (gTruth) which is the expected mask, and the last row is the generated mask by the trained model. Also, Table 4 shows the performance evaluation of this model. In this table, precision, recall, and Area Under Curve (AUC), F1-score, sensitivity, specificity, and Mean Intersection-Over-Union (Mean IOU) are reported.

Table 4: Performance evaluation of the customized U-Net on the fire dataset for the fire segmentation.

	Performance evaluation						
Dataset	Precision(%)	Recall(%)	AUC(%)	F1-Score(%)	Sensitivity(%)	Specificity(%)	IOU(%)
Image Segmentation	91.99	83.88	99.85	87.75	83.12	99.96	78.17

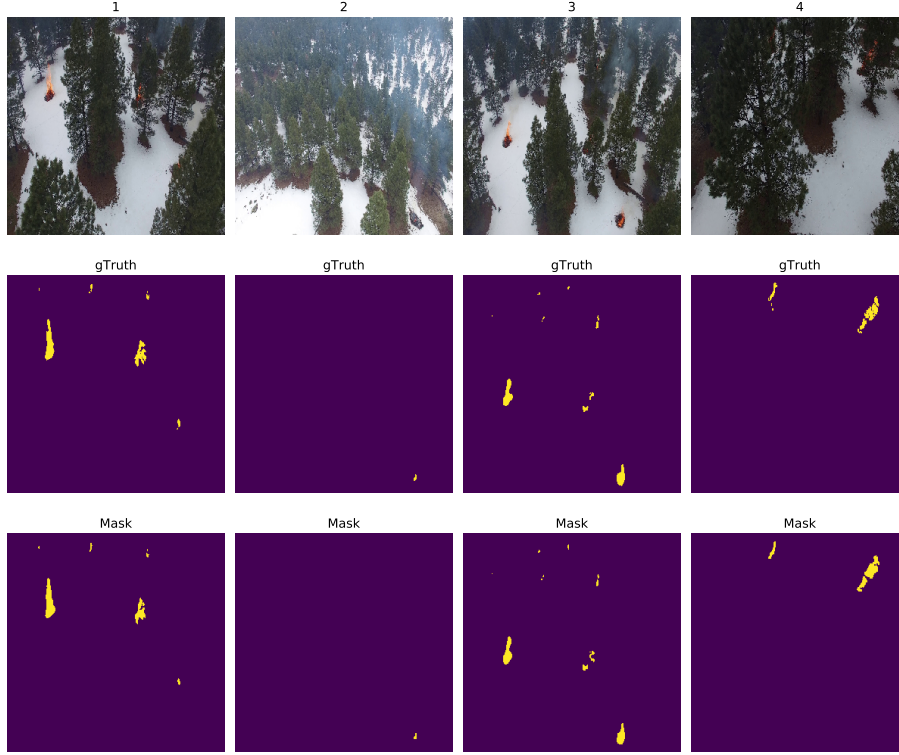


Figure 7: Performance of the fire segmentation on six frames of the test set.

5. Open Challenges regarding the dataset

This study proposed two different challenges regarding the dataset. We encourage other researchers to consider this available FLAME dataset and improve the accuracy of the fire classification problem which might include providing more ground truth data as a labeled mask for the fire segmentation. Also, three other thermal images such as GreenHot, WhiteHot, and the fusion are also available for further investigation regarding the fire segmentation and classification. Other considerations include the type of fire elements including the different structures of the fire (white hot core, exterior, etc). These elements can be segmented as different parts of the fire to have better understanding of the fire behavior. Another challenge or problem could be investigating different fire detection models on these thermal images to see which type of the data has better

accuracy for the model. Perhaps, another important research direction would be developing integrative imagery-based fire spread models by incorporating other environmental factors such as the terrain model, and the vegetation fuel profile of the region. Extracting such factors from the images and videos and comparing with alternative sources can advance the image-based fire modeling algorithms. Other open challenges and future directions regarding the FLAME dataset include but not limited to 1) transfer learning, 2) context-based fire detection using a model and then zero-shot learning, 3) fire content analysis, 4) temporal analysis, 5) surrogate airborne perspective analysis, 6) metric design, 7) performance standards, 8) user displays, 9) edge node efficiency, and 10) occlusion robustness.

6. Conclusion

This paper provided the FLAME (Fire Luminosity Airborne-based Machine learning Evaluation) dataset for pile burns in Northern Arizona forest. Two drones were used to collect aerial frames and videos in four different palettes of normal, Fusion, WhiteHot, and GreenHot using normal and thermal cameras. The frames were used in two different applications, in the first challenge, a convolutional neural network was used as a deep learning binary fire classification to label data. In the second approach, a machine learning approach was proposed to extract fire masks from fire labeled data as an image segmentation technique. These exemplary applications show the utility of the FLAME dataset in developing computer tools for fire management and control. Also, FLAME dataset can be used as a benchmark dataset for testing generic image processing algorithms. We provide numerical result for the performance of the proposed two algorithms developed for image classification and detection. We believe that developing more advanced models by the research community can further improve the reported results. Another potential use for this dataset is developing fire classification and detection algorithms by a collective analysis of different imaging modalities including regular and thermal images. Also, re-

searchers can utilize fire segmentation methods to define related networking and monitoring problems, such as optimal task scheduling for a fleet of drones to optimally cover the pile burns in a certain region at shortest time possible.

Acknowledgment

This material is based upon work supported by the Air Force Office of Scientific Research under award number FA9550-20-1-0090 and the National Science Foundation under Grant Number 2034218. Thanks to Neil Chapman and Paul Summerfelt of the Flagstaff Fire Department for providing access to the prescribed fire.

References

- [1] National Interagency Fire Center, <https://www.nifc.gov/fireInfo/nfn.htm>, (Accessed on 10/28/2020).
- [2] National Interagency Fire Center, https://www.nifc.gov/fireInfo/fireInfo_statistics.html, (Accessed on 10/28/2020).
- [3] J. Toledo-Castro, P. Caballero-Gil, N. Rodríguez-Pérez, I. Santos-González, C. Hernández-Goya, R. Aguasca-Colomo, Forest fire prevention, detection, and fighting based on fuzzy logic and wireless sensor networks, *Complexity* 2018 (2018) 1–17. doi:10.1155/2018/1639715.
- [4] H. Kaur, S. K. Sood, Fog-assisted IoT-enabled scalable network infrastructure for wildfire surveillance, *Journal of Network and Computer Applications* 144 (2019) 171–183.
- [5] J. L. Coen, W. Schroeder, S. D. Rudlosky, Transforming wildfire detection and prediction using new and underused sensor and data sources integrated with modeling, in: *Handbook of Dynamic Data Driven Applications Systems*, Springer, 2018, pp. 215–231.

- [6] F. Afghah, A. Shamsoshoara, L. L. Njilla, C. A. Kamhoua, Cooperative spectrum sharing and trust management in iot networks, *Modeling and Design of Secure Internet of Things* (2020) 79–109.
- [7] C. A. Kamhoua, L. Njilla, A. Kott, S. Shetty, Modeling and design of secure internet of things (2020).
- [8] Q. Huang, A. Razi, F. Afghah, P. Fule, Wildfire Spread Modeling with Aerial Image Processing, in: 2020 IEEE 21st International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM), IEEE, 2020, pp. 335–340.
- [9] P. Friedlingstein, M. Jones, M. O’sullivan, R. Andrew, J. Hauck, G. Peters, W. Peters, J. Pongratz, S. Sitch, C. Le Quéré, et al., Global carbon budget 2019, *Earth System Science Data* 11 (4) (2019) 1783–1838.
- [10] M. Keshavarz, A. Shamsoshoara, F. Afghah, J. Ashdown, A real-time framework for trust monitoring in a network of unmanned aerial vehicles, in: *IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, IEEE, 2020, pp. 677–682.
- [11] M. Keshavarz, M. Anwar, Towards improving privacy control for smart homes: A privacy decision framework, in: 2018 16th Annual Conference on Privacy, Security and Trust (PST), IEEE, 2018, pp. 1–3.
- [12] M. Keshavarz, M. Anwar, The automatic detection of sensitive data in smart homes, in: *International Conference on Human-Computer Interaction*, Springer, 2019, pp. 404–416.
- [13] M. Erdelj, E. Natalizio, K. R. Chowdhury, I. F. Akyildiz, Help from the sky: Leveraging UAVs for disaster management, *IEEE Pervasive Computing* 16 (1) (2017) 24–32.
- [14] F. Afghah, M. Zaeri-Amirani, A. Razi, J. Chakareski, E. Bentley, A coalition formation approach to coordinated task allocation in heterogeneous

- uav networks, in: 2018 Annual American Control Conference (ACC), 2018, pp. 5968–5975. doi:10.23919/ACC.2018.8431278.
- [15] R. Aggarwal, A. Soderlund, M. Kumar, D. Grymin, Risk aware suas path planning in an unstructured wildfire environment, in: 2020 American Control Conference (ACC), IEEE, 2020, pp. 1767–1772.
- [16] F. Afghah, A. Razi, J. Chakareski, J. Ashdown, Wildfire monitoring in remote areas using autonomous unmanned aerial vehicles, in: IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2019, pp. 835–840. doi:10.1109/INFCOMW.2019.8845309.
- [17] F. Afghah, A. Shamsoshoara, L. Njilla, C. Kamhoua, A reputation-based stackelberg game model to enhance secrecy rate in spectrum leasing to selfish iot devices, in: IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), IEEE, 2018, pp. 312–317.
- [18] A. Shamsoshoara, M. Khaledi, F. Afghah, A. Razi, J. Ashdown, Distributed cooperative spectrum sharing in uav networks using multi-agent reinforcement learning, in: 2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC), IEEE, 2019, pp. 1–6.
- [19] A. Shamsoshoara, M. Khaledi, F. Afghah, A. Razi, J. Ashdown, K. Turck, A solution for dynamic spectrum management in mission-critical UAV networks, in: 2019 16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), IEEE, 2019, pp. 1–6.
- [20] A. Shamsoshoara, F. Afghah, A. Razi, S. Mousavi, J. Ashdown, K. Turk, An Autonomous Spectrum Management Scheme for Unmanned Aerial Vehicle Networks in Disaster Relief Operations, IEEE Access 8 (2020) 58064–58079.

- [21] S. Mousavi, F. Afghah, J. D. Ashdown, K. Turck, Use of a quantum genetic algorithm for coalition formation in large-scale uav networks, *Ad Hoc Networks* 87 (2019) 26–36.
- [22] S. Mousavi, F. Afghah, J. D. Ashdown, K. Turck, Leader-follower based coalition formation in large-scale uav networks, a quantum evolutionary approach, in: *IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, IEEE, 2018, pp. 882–887.
- [23] S. S. Mousavi, M. Schukat, E. Howley, Traffic light control using deep policy-gradient and value-function-based reinforcement learning, *IET Intelligent Transport Systems* 11 (7) (2017) 417–423.
- [24] S. S. Mousavi, M. Schukat, E. Howley, Deep reinforcement learning: an overview, in: *Proceedings of SAI Intelligent Systems Conference*, Springer, 2016, pp. 426–440.
- [25] R. Sarcinelli, R. Guidolini, V. B. Cardoso, T. M. Paixão, R. F. Berriel, P. Azevedo, A. F. De Souza, C. Badue, T. Oliveira-Santos, Handling pedestrians in self-driving cars using image tracking and alternative path generation with frenét frames, *Computers & Graphics* 84 (2019) 173–184.
- [26] S. Mousavi, M. Schukat, E. Howley, A. Borji, N. Mozayani, Learning to predict where to look in interactive environments using deep recurrent q-learning, *arXiv preprint arXiv:1612.05753*.
- [27] Y. E. Wang, G.-Y. Wei, D. Brooks, Benchmarking TPU, GPU, and CPU platforms for deep learning, *arXiv preprint arXiv:1907.10701*.
- [28] Edge TPU - Google, <https://cloud.google.com/edge-tpu>, (Accessed on 10/29/2020).
- [29] NVIDIA Jetson Nano, <https://developer.nvidia.com/embedded/jetson-nano-developer-kit>, (Accessed on 10/29/2020).

- [30] H. Wu, H. Li, A. Shamsoshoara, A. Razi, F. Afghah, Transfer Learning for Wildfire Identification in UAV Imagery, in: 2020 54th Annual Conference on Information Sciences and Systems (CISS), IEEE, 2020, pp. 1–6.
- [31] DJI - Phantom 3 Professional, <https://www.dji.com/phantom-3-pro>, (Accessed on 08/30/2020).
- [32] D. Company, DJI - Matrice 200 V1, <https://www.dji.com/matrice-200-series/info#specs>, (Accessed on 08/30/2020).
- [33] DJI Zenmuse X4S - Specifications, FAQs, Videos, Tutorials, Manuals, DJI GO - DJI, <https://www.dji.com/zenmuse-x4s/info#specs>, (Accessed on 08/30/2020).
- [34] FLIR Vue Pro R Radiometric Drone Thermal Camera — FLIR Systems, <https://www.flir.com/products/vue-pro-r/>, (Accessed on 12/08/2020).
- [35] A. Shamsoshoara, F. Afghah, A. Razi, L. Zheng, P. Fulé, E. Blasch, Aerial images for Pile burn detection using drones (UAVs) (2020). doi:10.21227/qad6-r683.
URL <https://dx.doi.org/10.21227/qad6-r683>
- [36] A. Shamsoshoara, Aerial Images for Pile burn Detection Using Drones (UAVs), https://youtu.be/bHK6g37_KyA, wireless Networking & Information Processing (WINIP) LAB, accessed on 11/19/2020 (2020).
- [37] T. Celik, H. Demirel, Fire detection in video sequences using a generic color model, Fire safety journal 44 (2) (2009) 147–158.
- [38] M. M. Umar, L. C. D. Silva, M. S. A. Bakar, M. I. Petra, State of the art of smoke and fire detection using image processing, International Journal of Signal and Imaging Systems Engineering 10 (1-2) (2017) 22–30.
- [39] N. I. binti Zaidi, N. A. A. binti Lokman, M. R. bin Daud, H. Achmad, K. A. Chia, Fire recognition using RGB and YCbCr color space, ARPN Journal of Engineering and Applied Sciences 10 (21) (2015) 9786–9790.

- [40] C. Yuan, Z. Liu, Y. Zhang, UAV-based forest fire detection and tracking using image processing techniques, in: 2015 International Conference on Unmanned Aircraft Systems (ICUAS), IEEE, 2015, pp. 639–643.
- [41] X. Qi, J. Ebert, A computer vision based method for fire detection in color videos, *International journal of imaging 2 (S09)* (2009) 22–34.
- [42] S. Kundu, V. Mahor, R. Gupta, A highly accurate fire detection method using discriminate method, in: 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), IEEE, 2018, pp. 1184–1189.
- [43] F. Chollet, Xception: Deep learning with depthwise separable convolutions, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.
- [44] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.
- [45] C. Szegedy, S. Ioffe, V. Vanhoucke, A. Alemi, Inception-v4, inception-resnet and the impact of residual connections on learning, *arXiv preprint arXiv:1602.07261*.
- [46] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [47] Y. Li, Y. Yuan, Convergence analysis of two-layer neural networks with relu activation, in: *Advances in neural information processing systems*, 2017, pp. 597–607.
- [48] A. Shamsoshoara, Fire-Detection-UAV-Aerial-Image-Classification-Segmentation-UnmannedAerialVehicle,

[https://github.com/AlirezaShamsoshoara/
Fire-Detection-UAV-Aerial-Image-Classification-Segmentation-UnmannedAerialVehicle](https://github.com/AlirezaShamsoshoara/Fire-Detection-UAV-Aerial-Image-Classification-Segmentation-UnmannedAerialVehicle)
(2020).

- [49] C. Yuan, Z. Liu, Y. Zhang, Aerial images-based forest fire detection for firefighting using optical remote sensing techniques and unmanned aerial vehicles, *Journal of Intelligent & Robotic Systems* 88 (2-4) (2017) 635–654.
- [50] T. Çelik, H. Özkaramanlı, H. Demirel, Fire and smoke detection without sensors: Image processing based approach, in: 2007 15th European Signal Processing Conference, IEEE, 2007, pp. 1794–1798.
- [51] A. Khalil, S. U. Rahman, F. Alam, I. Ahmad, I. Khalil, Fire Detection Using Multi Color Space and Background Modeling, *Fire Technology* (2020) 1–19.
- [52] Image segmentation software from Labelbox, <https://labelbox.com/product/image-segmentation>, (Accessed on 10/25/2020).
- [53] image labelling tool: Bitbucket, <https://bitbucket.org/ueacomputervision/image-labelling-tool/src/master/>, (Accessed on 10/25/2020).
- [54] labelImg: LabelImg is a graphical image annotation tool and label object bounding boxes in images, <https://github.com/tzutalin/labelImg>, (Accessed on 10/25/2020).
- [55] Image Labeler, MATLAB & Simulink, <https://www.mathworks.com/help/vision/ug/get-started-with-the-image-labeler.html>, (Accessed on 10/25/2020).
- [56] GIMP - GNU Image Manipulation Program, <https://www.gimp.org/>, (Accessed on 10/25/2020).
- [57] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: International Conference on Medical

image computing and computer-assisted intervention, Springer, 2015, pp. 234–241.

- [58] ELU Activation Function, ML Glossary documentation, https://ml-cheatsheet.readthedocs.io/en/latest/activation_functions.html#elu, (Accessed on 10/26/2020).