AI

MDPI

Article

Deep Learning Based Wildfire Event Object Detection from 4K Aerial Images Acquired by UAS

Ziyang Tang [1], Xiang Liu [1], Hanlin Chen [1], Joseph Hupy [2,*] and Baijian Yang [1,*]

[1] Department of Computer and Information Technology, Purdue University, 401 N. Grant Street, West Lafayette, IN 47907, USA; tang385@purdue.edu (Z.T.); xiang35@purdue.edu (X.L.); chen1368@purdue.edu (H.C.)

[2] Aviation Technology, Purdue University, 1401 Aviation Drive, West Lafayette, IN 47907, USA

* Correspondence: jhupy@purdue.edu (J.H.); byang@purdue.edu (B.Y.); Tel.: +1-765-496-6201 (J.H.); +1-765-496-7143 (B.Y.)

Received: 26 February 2020; Accepted: 21 April 2020; Published: 27 April 2020

Abstract: Unmanned Aerial Systems, hereafter referred to as UAS, are of great use in hazard events such as wildfire due to their ability to provide high-resolution video imagery over areas deemed too dangerous for manned aircraft and ground crews. This aerial perspective allows for identification of ground-based hazards such as spot fires and fire lines, and to communicate this information with fire fighting crews. Current technology relies on visual interpretation of UAS imagery, with little to no computer-assisted automatic detection. With the help of big labeled data and the significant increase of computing power, deep learning has seen great successes on object detection with fixed patterns, such as people and vehicles. However, little has been done for objects, such as spot fires, with amorphous and irregular shapes. Additional challenges arise when data are collected via UAS as high-resolution aerial images or videos; an ample solution must provide reasonable accuracy with low delays. In this paper, we examined 4K (3840 × 2160) videos collected by UAS from a controlled burn and created a set of labeled video sets to be shared for public use. We introduce a coarse-to-fine framework to auto-detect wildfires that are sparse, small, and irregularly-shaped. The coarse detector adaptively selects the sub-regions that are likely to contain the objects of interest while the fine detector passes only the details of the sub-regions, rather than the entire 4K region, for further scrutiny. The proposed two-phase learning therefore greatly reduced time overhead and is capable of maintaining high accuracy. Compared against the real-time one-stage object backbone of YoloV3, the proposed methods improved the mean average precision(mAP) from 0.29 to 0.67, with an average inference speed of 7.44 frames per second. Limitations and future work are discussed with regard to the design and the experiment results.

Keywords: wildfire detection; deep learning; unmanned aerial systems; high resolution images; dataset

1. Introduction

Wildfire is one of the most common hazardous threats to human society. According to the 2017 report from National Fire Protection Association [1], 3400 people died in United States wildfires with associated property damage estimated at around 23 billion USD. To face the growing threat of wildfire, fire fighting crews and first-responders rely on different ways to combat, control, and eliminate the threat of wildfire. Among the methods and tools available, Unmanned Aerial Systems (UAS), commonly referred to as drones, are seeing increasing use in supporting ground crews in fire events [2,3]. UAS are a powerful tool in wildfire because of their ability to deploy rapidly and into areas where manned aircraft cannot.

Despite the utility of UAS in hazard events such as wildfire, the technology still requires an individual on the ground to interpret the images and remain vigilant using the human eye alone to identify fires and other associated hazards. However, in an actual wildfire event, there is a distinct level of organized chaos where the operator of a UAS platform is often monitoring hundreds of spot fires and having to track multiple fire crews and related equipment on the ground, often over extend periods as long as 10 h. Since humans are fallible and suffer from fatigue and shortened attention spans, some objects can be overlooked. In comparison, the machines can stand by for 24/7 and assist fire fighters in wildfire detection based upon imagery gathered via UAS. In this paper, we aim at filling this gap and introduce an algorithm that will assist ground-based crews with identification of fire and other ground-based objects related to a fire event.

In past few decades, algorithms using sensor and traditional hand-craft features like edge detection [4] have been applied in fire detection, but the sensor-based methods can only detect fire in the images, and cannot detect other related objects like firefighters and vehicles. Recently, object detection methods using deep neural networks that have witnessed great breakthroughs in detecting objects in images, e.g., YOLO, SSD [5,6]. However, no previous work has been done in wildfire detection using deep neural networks. To apply deep learning into wildfire detection, we have two main challenges:

(1) Aerial UAS imagery are usually acquired in high-resolution. However, the promising results from CNN-based object detectors are from low-resolution images (600 px $\times$ 400 px). The results from high-resolution aerial images are far from satisfactory because of the small and sparse objects.
(2) CNN-based detectors heavily rely on well-annotated datasets. The fire can be amorphous and cannot be annotated with a single rectangle bounding box. Therefore, it is time-consuming to label the fire from high resolution images. To our best knowledge, there is no available public aerial fire dataset.

To address the aforementioned issues, we introduce a new wildfire dataset. To the best of our knowledge, this dataset is the first public 4K UAS fire dataset with annotations containing 1400 aerial images and 18,449 instances. Each instance represents an object such as fire, trucks, or people. Based on this dataset, we propose an Adaptive sub-Region Select Block (ARSB) to extract a rough area that contains the objects from high resolution images. After the extraction, we then zoom into these areas to detect the small objects, and fuse the final results back to original images. Further experiments show that the method can achieve a promising accuracy while keeping the processing speed.

The contributions of this work are as follows:

- We compiled a large-scale aerial dataset, heavily annotated for fire, persons, and vehicles. The dataset contains 1400 images from 4K videos taken with a ZenMuse XT2 sensor from Purdue Wildlife Area (West Lafayette, IN, USA) on a DJI M600 drone over a controlled burn covering approximately two hectares.
- We provide a fast and accurate coarse-to-fine pipeline to detect the small objects in 4K images. By a carefully designed CNN model, we can locate the center point of the objects and reduce the image size into low resolution with no loss of the objects.
- In this work, we applied the deep learning models in 4K aerial fire detection. In addition, compared to other fire detectors using special sensors, we can not only predict the fire with state-of-art accuracy, but also provide predictions for other objects such as persons or vehicles.

The remainder of the paper is structured as follows. In Section 2, relevant background is introduced. In Sections 3 and 4, the dataset and the methodology are presented, respectively. Experiments, analyses, ablation studies, and limitations are discussed in Section 5. The paper is concluded in Section 6.

2. Related Work

In this section, we summarize related object detection methods using deep learning methods, and review machine learning and deep learning methods used in wildfire detection.

2.1. Object Detection Using Deep Learning

In early research centered on object detection tasks, researchers use hand-crafted features like SIFT [7] and classifiers like DPM [8] to find the objects. In past decades, with the GPU support on the large parallel computation for deep convolution networks, the deep CNN-based detectors gradually outperformed traditional hand-crafted based detectors. We can roughly divide the deep CNN-based detectors into two categories, the two-stage approach for higher accuracy and one-stage approach for fast speed.

2.1.1. Two-Stage Object Detectors

The two-stage detectors first generate several candidate sets of Regions of Interest (ROI) areas and determine accurate regions based on the ROIs. In previous work, researchers implemented several optimizations for extracting ROIs, including Selective Search [9], Deep Mask [10], and RPN [11]. Similarly, numerous effective techniques have been proposed to improve detection in the second stage, from the perspective of network structures [12,13], optimizers [14], multiple features [15], and prior and contextual information [16].

2.1.2. One-Stage Object Detectors

Unlike two-stage detectors, one-stage object detectors mainly focus on processing speeds. Instead of predicting candidate sets of ROI first, the one-stage detectors predict the regions in a single forward manner. Two representative structures in one-stage detectors are YOLO [5] and SSD [6]. Based on these two backbone structures, further modifications are implemented to improve either the accuracy or speed of the one-stage detectors. Among these modifications, Yolov2 [17] provided a detector with higher mAP and faster detection speed by applying batch normalization and anchor mechanism. DSSD [18] introduced transform convolution networks for additional context. Other works also proposed different perspectives including using RNN [19], enlarging the reception field [20].

2.2. Detection of Small Objects in High-Resolution Aerial Images

Most popular methods dealing with high-resolution aerial image detection use the coarse-to-fine pipeline approach. The idea is to crop high-resolution images into smaller pieces, and feed these pieces into object detection frameworks like YOLO [5] or SSD [6]. However, in most cases, the targets in the images are sparse and non-uniform. Therefore, no existing function can easily crop the most relevant part. Since this work is focused on deep learning, we only review some relevant cropping methods in the area of deep learning. Sommer et al. [21] extended a two-stage object detection method and stated a coupled region-based CNN for aerial vehicle detection. Lu et al. [22] introduced an adaptive cropping method to locate the sub-region with these small targets. Alexe [23] proposed a context-driven region search method and Rzicka [24] created an attention pipeline to actively crop the sub-regions within small objects. Recently, Yang [25] extracted sub-regions with a cluster proposed sub-network and used an iterative cluster merging algorithm to merge the sub-regions.

Although previous works have explored several directions to use regional search algorithms effectively, most of the algorithms aimed at finding fixed shapes, such as vehicles. Otherwise, they only focus on detecting the fires [26]. On the contrary, in our research, we are detecting both wildfire and other related objects such as vehicles and persons. In our work, one significant challenge is amorphous behavior of fire: it can be a thin, long line, as well as taking shape as a polymorphous, polygonal fire. Therefore, the annotation of the fire is much more challenging than the objects with fixed shapes.

To our best knowledge, there is no existing well-annotated public aerial fire dataset and we provide a well-annotated fire dataset and applying the deep learning method in fire detection.

2.3. Fire Detection via Machine Learning and Deep Learning Methods

Fire detection is often achieved by analyzing imagery captured by satellites and aerial platforms equipped with multi-band sensors [27]. Researchers initially analyzed the data by detecting the spectral signature, or color property, of fire [28]. Çelik et al. [29] employed two different color spaces for fire detection. They used RGB color space for fire detection and YCbCr color space to improve the performance of the results from RGB color space analysis. They also applied fuzzy logic to discriminate fire from the other fire-like objects. These color based methods presented high false positives due to the presence of other fire-like objects. To overcome this issue, Qiu et al. [4] proposed an auto-adaptive edge detection method to detect fire, but the results were still unsatisfactory with regard to false positives and desired accuracies. Mueller et al. [30] proposed an optical flow based method to distinguish fire from the fire-like objects. Other previous research has leveraged data mining techniques [31] and image processing methods [32] to detect fire. Such an approach is limited in that it can only detect a rough region of the fire from special images captured by thermal or spectral sensors.

With the increasing popularity of deep learning, various deep learning methods are seeing use in fire detection. Zhang et al. [33] employed AlexNet [34] and fine-tuning technique to detect fire in wildfire events. Sharma et al. [35] proposed ResNet-based methods [13] for fire detection. The disadvantage of these methods is that the neural network models require high-configuration hardware and are unsuitable for embedded systems. SqueezeNet and MobileNetV2 were proposed to meet the constraints of embedded systems [36,37]. At the cost of lower accuracy, these lightweight neural network models are computationally efficient and can be deployed to platforms like Raspberry Pi. Both SqueezeNet and MobileNetV2 were not designed for fire detection. Firenet was proposed in 2019 Jadon et al. [38]. The "designed-from-scratch" lightweight neural network structure was carefully structured for fire detection. However, these methods can only detect fire. In reality, detecting persons, vehicles, and other object of interest are crucial in fire fighting strategies. In contrast, our proposed methods can detect not only fire, but also other objects like vehicles and persons.

3. Dataset Specification

3.1. Overview of the Dataset

The data gathered for this research were collected during a controlled burn activity. The controlled burn was conducted for wildlife management purposes, and provided an excellent opportunity to gather a sample dataset to work with. Although it is normally illegal to fly UAS in larger fire events where a temporary flight restriction (TFR) restricts all flights in the vicinity, one of the authors in this paper, is authorized to fly these fires through the U.S. Department of the Interior. Such larger fires are very complex, and it was determined that this research should first focus on a smaller, more contained, controlled burn to develop and validate the method. A DJI M600 equipped with a ZenMuse XT2 Eo/Thermal Sensor was used to gather the videos above burn activity. A total of eight videos were gathered with the resolution of 4K (3840×2160). Among the videos, 1400 key frames are extracted and then divided into a training and validation set with the split rate of 80%/20%. In addition, 18,449 instances are annotated in the training set while 4093 instances in the validation set.

3.2. Category Selection and Annotation Methods

During a fire event, it is important to be able to easily recognize and identify key ground-based objects. Other than fire identification in the form of spot fires and fire lines, the most common objects are the vehicles and persons associated with combating the fire. Hence, we chose these three objects as the categories in our dataset. The vehicles and persons can be easily annotated using regular rectangle bounding boxes, but the shape of fire is irregular and we cannot label it with a single rectangular

bounding box. The fire takes shape along its leading edge as a line, or as smaller sporadically located spot fires. Applying a single and large bounding box on the fire line can cover a large area of the background and result in poor performance of distributing the fire in the background. Instead, we use a small bounding box to annotate fire in the images.

4. Methodology

As Figure 1 shows, we applied a coarse-to-fine strategy to detect the small objects in 4K resolution images. In the coarse stage, an adaptive sub-Region Select Block (ARSB) is applied to find rough areas in 4K resolution images. Then, we zoomed into these sub-regions from the original images, while maintaining the size of the area bounding box. In the fine stage, we applied state-of-art object detection backbones like YoloV3 [39] to detect the objects. Finally, we combined the bounding boxes in the sub-regions and zoom back out to the original images.

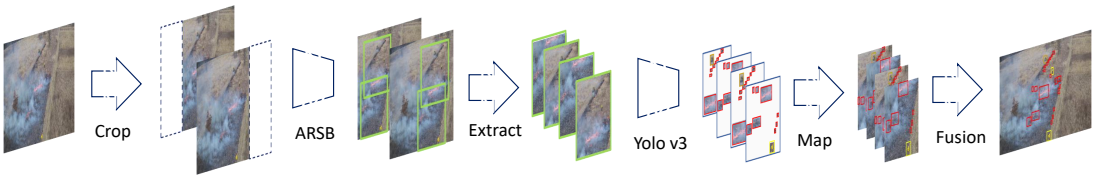


Figure 1. An overview structure of the proposed method. The pipeline crop 4K images into image chips, then the proposed ARSB extracts rough sub-regions, which zoom in for further detection. Final results come from the fusion of these sub-regions.

4.1. Adaptive Sub-Region Select Block (ARSB)

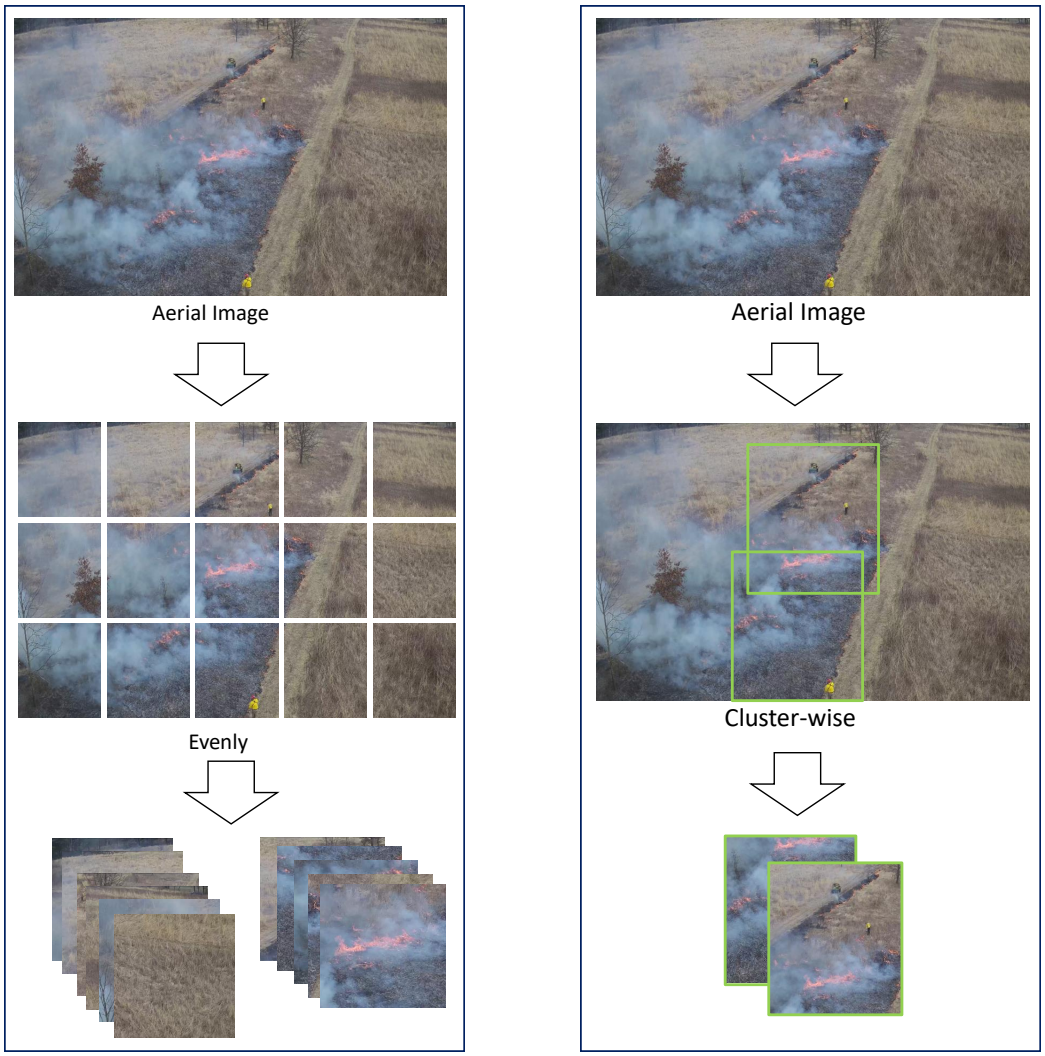
One challenging task in the field of object detection for high resolution images is to find small objects correctly. A straightforward method is to split the high resolution images evenly. However, this simple cropping method can be inefficient when the objects are sparse in the images. As Figure 2a shows, more than half of the evenly cropped image clips have no objects due to the sparseness. However, these additional image clips will increase the processing time dramatically. A characteristic of the objects in the images is that they are usually clustered in several sub-regions. In this case, we propose an adaptive sub-region select block (ARSB) to get the sub-regions. Our idea is inspired by region-based proposal networks (RPN) [11], which aims at the candidate proposals from the feature maps. Similarly, we develop a mechanism for the coarse detector to select a rough region containing objects. Because of the sparseness of the objects in aerial images, the number of the predicted sub-regions is fewer than the evenly cropped regions. Instead, it can adaptively select regions and feed the sub-regions clips to the next fine detector and predict precise bounding box positions. Although our idea is similar to RPN, the method is different because we are cropping the sub-regions from the original images, and then zooming into these cropped regions for fine detection. As Figure 2b illustrates, significantly fewer image clips are selected compared to the even cropping methods.

The process of extracting the sub-region can be considered as a supervised learning method. However, none of the current datasets will provide ground-truthed information for the rough sub-regions containing objects. In this work, we propose an algorithm called Iterative Bounding-Box Merge (IBBM) to merge the small objects as ground truths of the sub-regions. IBBM can also be applied to other public datasets to merge and generate large regions for coarse detection. The idea of IBBM comes from the algorithm of non-maximum suppression (NMS) [40]. NMS compares the predicted bounding boxes with the ground truth, keeping the bounding box with the largest confidence and suppressing the rest of the bounding boxes with IoU scores larger than the pre-defined threshold. We modified this algorithm and used a pre-defined large bounding box, denoted as an anchor, to merge the ground truth that has IoU scores larger than the threshold τ_{IBBM}. The pseudo-code of IBBM is shown in Algorithm 1, and the function RECENTER in the algorithm is applied to relocate the center of B_i to ensure that the rescaled B_i with new height h_b and width w_b is still in the image.

Algorithm 1 Iterative Bounding-Box Merge (IBBM)

Input: Bounding boxes of an image $\mathcal{B} = \{B_i\}_{i=1}^{N_\mathcal{B}}$,
classes of the bounding boxes $\mathcal{C} = \{C_i\}_{i=1}^{N_\mathcal{C}}$, desired bounding box height h_b and width w_b, non max merge threshold τ_{IBBM}
Output: Merged bounding boxes $\mathcal{B}'$

$\mathcal{B}' = \{\}$
for $i \leftarrow 1$ to $N_\mathcal{B}$ **do**
  if B_i is **visited then**
    continue
  Flag B_i as **visited**
  $B_i^\dagger \leftarrow \text{RECENTER}(B_i, h_b, w_b)$
  for $j = i + 1$ to $N_\mathcal{B}$ **do**
    if $C_i \neq C_j$ **then**
      continue
    if $\text{IoU}(B_i^\dagger, B_j) > \tau_{IBBM}$ **then**
      $\mathcal{B}' \leftarrow \mathcal{B}' \cup \{B_j\}$
      Flag B_j as **visited**
return $\mathcal{B}'$



(**a**) evenly cropping method (**b**) ASRB

Figure 2. Comparison of evenly cropping methods (**a**) vs. ASRB (**b**). Most evenly cropped image clips have no objects. The proposed ARSB methods crop only the area that contains the objects. It greatly reduce the number of image clips needed for further processing.

Given the generated ground truth, we follow the ideas in Yolov3 [39] and consider the sub-region extraction as a regression problem. In the extraction phase, we are interested in whether this region contains objects or not, hence we can only predict two classes (p_{obj} and p_{noobj}), along with the information of the prediction bounding boxes (x, y, w, h). x, y is the center point of the bounding box and w, h is the box size. The width and height of the generated ground truth are fixed to a certain size and we can accordingly fix the (w, h) output in this training phase and only regress the results for (x, y). The overall cost function can be derived as a weighted sum of the confidence loss L_{conf} and bounding box regression loss L_{reg}:

$$L_{conf} = L_{bce}(p_{obj}, g_{obj}) + \lambda_1 L_{bce}(p_{noobj}, g_{noobj}) \tag{1}$$

$$L_{reg} = L_{mse}(p_x, g_x) + L_{mse}(p_y, g_y) \tag{2}$$

$$L_c = L_{conf} + L_{reg} \tag{3}$$

where p_{obj} and p_{noobj} represent the possibility that the anchor in a cell has objects or not. g_{obj} and g_{noobj} means whether a cell has objects. In the implementation, we use an IoU threshold $\tau = 0.5$ to get the value of g_{obj} and g_{noobj}. p_x and p_y are the center of the bounding boxes, g_x and g_y are the center of the ground truth box.

L_{bce} is the loss of binary cross-entropy and L_{mse} is the loss of mean square, which could be represented as follows:

$$L_{bce}(x, y) = \frac{1}{n} \sum_{i}^{n} \sum_{j}^{m} \mathbb{I}_{ij}^{c} (y_j log x_i + (1 - y_j) log(1 - x_i)) \tag{4}$$

$$L_{mse}(x, y) = \frac{1}{n} \sum_{i}^{n} \sum_{j}^{m} \mathbb{I}_{ij}^{c} (x_i - y_j)^2 \tag{5}$$

where $\mathbb{I}_{ij}^{c}$ is an indicator function for matching the ith prediction box with the Jth ground truth box of category c. In this phase, c will the same because we only have two classes: objects and no objects. n is the size of the prediction boxes, and m is the size of the ground truth boxes.

In most cases, the cell will not contain any objects. Therefore, the negative samples will be much more than the positive samples. In this case, we add L_{noobj} as a penalty term to regularize the imbalance case, and set up a scalar λ_1 as a weight hyper-parameter.

4.2. Fine Object Detection

Because we fix the width and height of the output from a coarse detector, the fine object detection can directly use the output to crop from the original images and feed into the network for final detection. Notice that, in fine detection phase, we resize the image from the size of 1080×1080 to 416×416, instead of scaling the 3840×2160 images to 416×416. From Yolov2 [17], prior information about the anchor box sizes help to better train the model. We followed the modified k-means algorithm to generate the priors of the anchor size. We used the distance defined in Yolov2 [17] instead of the standard Euclidean distance:

$$d(box, centroid) = 1 - \text{IoU}(box, centroid) \tag{6}$$

where we set the number of centroid to 9. The IoU means the Intersection over Union of two boxes, which is a criterion to measure the similarity of the two boxes. For our proposed dataset, the ratio of the width and height can vary from extremely short wide boxes (w/h ratio = 5) to extremely tall, thin boxes (w/h ratio = 0.2). In addition, about 70% of the boxes are smaller than 50×50. When we directly apply the k-means to the ground truth, most of the top-9 boxes are small sized. To give some prior information for larger objects, we set up a threshold to label the boxes into three types: small, medium,

and large. Then, we run the k-means algorithms for different types of boxes with centroid = 3 and combine the nine total centroids as the prior for the anchors.

The prediction is similar to the process in the coarse detection phase. The difference is that the width and height of the prediction boxes are not fixed in the fine detection phase, and therefore we should learn the box size from the networks. We also need to predict multiple classes as opposed to just two classes. Thus, we can derive the loss function for fine detection. To differentiate from the notations in coarse phase, we denote the confidence of the bounding box as fine confidence loss L_{fconf}, and fine regression loss accordingly L_{freg}. In addition, we also predict the class of the bounding box, so we have one more loss to predict the class L_{cls}:

$$L_{fconf} = L_{bce}(p_{obj}, g_{obj}) + \lambda_2 L_{bce}(p_{noobj}, g_{noobj}) \tag{7}$$

$$L_{freg} = L_{mse}(p_x, g_x) + L_{mse}(p_y, g_y) + L_{mse}(p_w, g_w) + L_{mse}(p_h, g_h) \tag{8}$$

$$L_{cls} = \sum_{k}^{C} L_{bce}(p_c^k, g_c^k) \tag{9}$$

$$L_f = L_{fconf} + L_{freg} + L_{cls} \tag{10}$$

where C is the number of the classes. p_c^k means the cell has an object in a particular class k. g_c^k means the class in best match ground truth box. We are using one-hot encoding to predict the class and therefore the value of p_c^k and g_c^k will be either 1 or 0. Thus, we use the binary cross entropy loss for L_{cls}.

Finally, we can combine loss in the coarse phase and loss in the fine stage together, and the overall loss is:

$$Loss = \lambda_c L_c + \lambda_f L_f \tag{11}$$

where λ_c and λ_f are the weighted scalars. In real implementation, we first set $\lambda_c = 1, \lambda_f = 0$ to fix the parameter in fine detector and train the coarse detector. Then, we fix the coarse detector and train the fine detector by setting $\lambda_c = 0, \lambda_f = 1$.

4.3. Fusion of the Final Results

Final detection within an aerial image can be obtained by the fusion of all bounding boxes with the standard NMS [40] algorithm. The sub-regions extracted from ARSB can be overlapped with each other, and therefore we use the standard Non-Maximum Suppression (NMS) to get rid of the bounding boxes with high overlap on the same object.

5. Experiments

5.1. Implementation Details

We used our proposed 4K fire aerial dataset to train the models. The training set contained 1151 images, with 18,469 instances in total. Among the instances, 6.21% of the instances are vehicles (ATVS, cars, light duty trucks), 10.27% of the instances are people, and 83.52% of the instances are fire. We also calculated the 4K imagery bounding box sizes. As Figure 3 shows, 16,325 instances have a small size less than 250 px × 250 px, and 1871 medium boxes with the size from the range of 250 px × 250 px to 500 px × 500 px and 273 large boxes with a size larger than 500 px × 500 px. In the coarse detection phase, we applied ARSB to zoom into the large sub-regions. We observed that 99.9% of the box is smaller than the size of 1080 px × 1080 px, and therefore we set up the sub-region size as 1080 px × 1080 px and used the IBBM to generate the ground truth. In the implementation, we applied two different methods in dealing with the original approach. The first method adds padding to a 4K image and makes it into 3840 px × 3840 px, before we resize it to 608 px × 608 px and extract the sub-regions

from the resized images. We propose this method as ARSB_pad for the duration of this paper. We also cropped the 4K images and converted them into two square images 2160 px × 2160 px. We then resized the cropped images to 608 px × 608 px to extract the sub-regions. This method was named as ARSB_crop. The ARSB was built up with three layers of ResNet Blocks. Each layer downsamples the input size by 2, and the number of the ResNet Blocks are 1, 2, 8. We then used the large anchor to search the grid cells and predict the sub-regions.

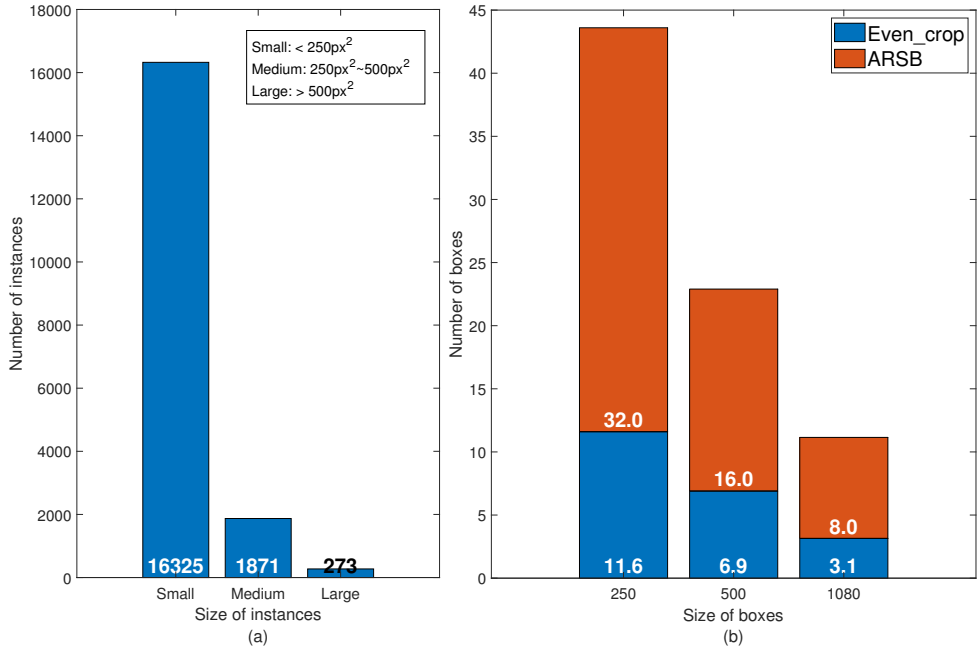


Figure 3. The observation of the ground truth details. (**a**) shows that the distribution of small, medium and large bounding box size; (**b**) shows the comparison of the cropped image clips for fine detectors.

In the fine detection phase, we observed the prior anchor size using a k-means algorithm [17]. In our experiments, we are using the nine cluster centers from Table 1 as prior anchors. A darknet-53 with Feature Pyramid Network (FPN) on the last three layers is applied as the backbone and we used three anchors for the grid search in each prediction layer.

Table 1. Nine cluster centers with objects of different sizes.

Scale	Cluster Centers
small	$(61, 9), (17, 22), (22, 50)$
medium	$(36, 30), (43, 65), (68, 41)$
large	$(67, 107), (108, 63), (156, 134)$

We trained the model in two steps. At first, we cropped the original images with the size of the bounding box 1080 px × 1080 px at the center of each ground truth. We also experimented other box sizes as 720 px × 720 px and 540 px × 540 px. As Table 2 illustrates, the performance is better with the box size of 1080 px × 1080 px. We then trained the model for 20 epochs with the learning rate of 1×10^{-3} for the first five epochs and 1×10^{-4} for the rest of the epochs. We then used the output from the coarse detectors and fine-tuned the fine detector for 20 epochs with the learning rate of 1×10^{-4}. We adopted the Adam optimizer [14] when training the model.

Table 2. Performance comparison on **UAS fire test data**.

Method	Size	AP_{30} Car	AP_{30} Person	AP_{30} Fire	mAP	AP_{50} Car	AP_{50} Person	AP_{50} Fire	mAP	AP_{70} Car	AP_{70} Person	AP_{70} Fire	mAP
yolo_ori	-	0.63	0.63	0.38	0.55	0.42	0.37	0.09	0.29	0.07	0.05	0.01	0.04
yolo_crop	540	0.80	0.86	0.56	0.74	0.72	0.75	0.32	0.60	0.45	0.35	0.08	0.29
	720	0.88	0.83	0.53	0.75	0.85	0.73	0.27	0.62	0.56	0.34	0.09	0.33
	1080	0.87	0.85	0.57	0.76	0.87	0.72	0.36	0.65	0.68	0.30	0.07	0.35
ARSB_pad	540	0.41	0.73	**0.62**	0.59	0.47	0.85	**0.39**	0.57	0.20	0.23	**0.14**	0.19
	720	0.64	0.79	0.60	0.68	0.71	0.80	0.23	0.58	0.39	0.30	0.03	0.24
	1080	**0.91**	0.88	0.50	0.76	0.75	0.79	0.35	0.63	0.67	0.31	0.11	0.37
ARSB_crop	540	0.59	**0.95**	0.58	0.71	0.60	**0.91**	0.18	0.56	0.45	0.35	0.07	0.29
	720	0.76	0.86	0.56	0.73	0.78	0.80	0.11	0.57	0.44	0.34	0.04	0.27
	1080	**0.91**	0.84	0.59	**0.78**	**0.88**	0.76	0.37	**0.67**	**0.82**	**0.41**	0.12	**0.45**

5.2. Evaluation Metric

We evaluated our model through the lens of accuracy and processing time. For accuracy measurement, we followed the standard Intersection over Union (IoU) metric defined in Pascal VOC [41]:

$$\text{IoU} = \frac{area(B_{pred} \cap B_{gt})}{area(B_{pred} \cup B_{gt})}, \tag{12}$$

where B_{pred} denotes the prediction bounding boxes and B_{gt} refers to the ground truth bounding boxes. To classify if a predicted bounding box is a true positive, the typical approach is to determine if an IoU is greater than a predefined threshold. For example, AP_{30} means that a prediction bounding box is considered true positive when its IoU is greater than 30%. Likewise, AP_{50} and AP_{70} means the lower bound of true positive are IoU of 50% and 70%, respectively.

A slightly different evaluation metric, mAP, is defined in the MS COCO competition [42]. Instead of relying on a single IoU threshold, mAP reflects the mean value of multiple IoU thresholds, ranging from 0.5 to 0.95, with a step size of 0.05. The formal definition of mAp is illustrated in Equation (13):

$$mAP = \frac{1}{11} \sum_{\text{IoU}=0.5:0.05:0.95} AP_{\text{IoU}} \tag{13}$$

We measured the processing time of the model by calculating the number of frames per second (FPS). In this paper, we aimed to propose an approach to deal with the high resolution images with fast detection speed while maintaining the accuracy as the detection performance in low resolution images. Hence, we show processing time measurements.

5.3. Baseline Methods

We apply the fast one-stage detector algorithm of Yolov3 [39] as the baseline. We denote the original Yolov3 methods as Yolo_ori. We also extend the original Yolo method by splitting the high resolution image evenly and detect the object in each image clips. For convenience, we named the method as Yolo_crop.

5.4. Ablation Study

In this section, we conducted extensive experiments on each component of our methods and to show the improvement from them. We first applied settings from yolov3 [39], then evaluated improvement from even cropping methods. To compare the effect fairly, we applied the same backbone to our proposed method.

5.4.1. The Baseline from Original Yolov3

Results are shown in Table 3. The speed from the original Yolo is fast, but the detection performance is poor. One of the main reasons is that a grid cell in the feature maps may represent tens of pixels from the original images, resulting in an offset of the prediction bounding boxes to actual objects. As Figure 4a illustrates, when the objects are small from the image, a slight offset will result in the inaccuracy in the performance.

Table 3. Detection performance on the **UAS fire test data** for different models. The mAP is based on AP_{50}.

Method	mAP	Frame per Second (FPS)
yolo_ori [39]	0.29	**50.6**
yolo_crop	0.65	1.92
ARSB_pad	0.66	**20.6**
ARSB_crop	**0.67**	7.44

5.4.2. The Improvement from Yolo_crop

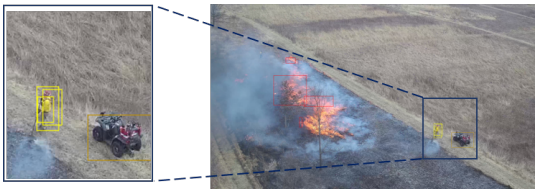
The results are listed in Table 3. By cropping the high resolution images to several smaller images, the mAP increases dramatically. Meanwhile, the processing time drops because more images need to be processed. In addition, as Figure 4b demonstrates, even cropping methods truncate some objects, which increases false positives.

5.4.3. Improvement by Our Methods

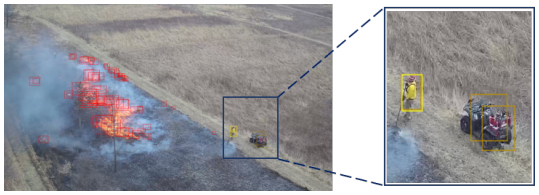
The result of our method has better mAP and speed than Yolo_crop. As Figure 4c,d shows, the coarse detection finds much fewer sub-regions than Yolo_crop. We also provide a speed-accuracy trade-off in our methods. As the results convey in Table 3, ARSB_pad is faster and ARSB_crop is more accurate. Hence, both results are faster in processing time because of the lesser number of processing images. In addition, the mAP is also increased. This is because the sub-regions approach lowered false positives. Consequently, the value of Precision increases, and hence there is a much better-improved mAP using our approach.

(**a**) the prediction results from Yolo_ori

(**b**) the prediction results from Yolo_crop



(**c**) the prediction results from ASBR_pad



(**d**) the prediction results from ARSB_crop

Figure 4. Result comparison of baseline and our proposed models. (**a**) the predictions from original Yolov3 have obvious mislabels from the actual objects; (**b**) the prediction results from Yolo_corp. The car in the figure is truncated. (**c**) the results of ARSB_pad can detect the objects correctly; (**d**) the results of ARSB_crop. The prediction is more accurate than other methods.

5.5. Limitations

Our proposed model relies heavily on the detection results from the coarse detector. If a sub-region is ignored by the coarse detectors, the object in this sub-region will not be detected by the fine detector. Due to time and resource constraints, only YoloV3 was chosen as the baseline in this study because it outperforms other deep learning approaches in similar applications.

The size of our wild-fire image collection used in this study also presents constraints and limitations. Because of the legal constraints of flying UAS in wild-fire events, UAS fire event image collections are rare and, to our knowledge, not publicly available. The dataset used here was gathered during a controlled burn and limited to scenes acquired during daylight hours. Hence, our proposed model may not be robust enough with wildfire in a different background at night, and with thermal imagery. Future research will address this limitation through inclusion of more imagery through ongoing UAS wildfire image acquisition efforts.

Finally, it may still be challenging to apply our method to onboard UAS processing. Applying the whole model to onboard UAS processing requires faster CPUs and larger memory GPUs.

6. Conclusions and Future Work

In this paper, we aimed at implementing an application in detecting fire and other critical ground-based objects in a wildfire event using high resolution aerial images. We propose a well annotated fire dataset with 1400 4K images. We also present a coarse-to-fine strategy to deal with the 4K images, which achieves high accuracy while maintaining fast speeds. Our methods can also be added to different backbones in object detection methods and extended to deal with high resolution images.

Ongoing and future research objectives involve expansion of the UAS wildfire imagery collection, and working with a UAS platforms equipped with more powerful CPUs and GPUs. Fusing data collected from multiple types of sensors can provide additional wisdom in wildfire fighting scenarios. Additional Machine Learning approaches, especially a hybrid approach that combines signal processing with deep learning, will be investigated to discover a faster and more accurate technique to identify small objects of interests and objects with irregular boundaries in high definition videos and images.

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

Funding: This research was partially funded by the Purdue Polytechnic Institute.

Acknowledgments: We should thank Peter Menet from MenetAero for performing the flights, and also Jarred Brooke from Purdue Forestry and Natural Resources extension for coordinating with us on the controlled burn.

Conflicts of Interest: The authors declare no conflict of interest. The funder had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Evarts, B. Fire Loss in the United States during 2017. 2018. Available online: https://www.darley.com/documents/inside_darley/NFPA_2017_Fire_Loss_Report.pdf (accessed on 22 April 2020).
2. Apvrille, L.; Tanzi, T.; Dugelay, J.L. Autonomous drones for assisting rescue services within the context of natural disasters. In Proceedings of the 2014 XXXIth URSI General Assembly and Scientific Symposium (URSI GASS), Beijing, China, 16–23 August 2014; pp. 1–4.
3. Eyerman, J.; Crispino, G.; Zamarro, A.; Durscher, R. *Drone Efficacy Study (DES): Evaluating the Impact of Drones for Locating Lost Persons in Search and Rescue Events*; European Emergency Number Association: Brussels, Belgium, 2018.

4. Qiu, T.; Yan, Y.; Lu, G. An autoadaptive edge-detection algorithm for flame and fire image processing. *IEEE Trans. Instrum. Meas.* **2012**, *61*, 1486–1493. [CrossRef]
5. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 779–788.
6. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 21–37.
7. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
8. Felzenszwalb, P.F.; Girshick, R.B.; McAllester, D.; Ramanan, D. Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell.* **2009**, *32*, 1627–1645. [CrossRef]
9. Uijlings, J.R.; Van De Sande, K.E.; Gevers, T.; Smeulders, A.W. Selective search for object recognition. *Int. J. Comput. Vis.* **2013**, *104*, 154–171. [CrossRef]
10. Pinheiro, P.O.; Collobert, R.; Dollár, P. Learning to segment object candidates. In *Advances in Neural Information Processing Systems*; NIPS: Montreal, QC, Canada, 2015; pp. 1990–1998.
11. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*; NIPS: Montreal, QC, Canada, 2015; pp. 91–99.
12. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A.A. Inception-v4, inception-resnet and the impact of residual connections on learning. In Proceedings of the Thirty-First, AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; pp. 4278–4284.
13. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.
14. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980
15. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
16. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2961–2969.
17. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
18. Fu, C.Y.; Liu, W.; Ranga, A.; Tyagi, A.; Berg, A.C. Dssd: Deconvolutional single shot detector. *arXiv* **2017**, arXiv:1701.06659.
19. Ren, J.; Chen, X.; Liu, J.; Sun, W.; Pang, J.; Yan, Q.; Tai, Y.W.; Xu, L. Accurate single stage detector using recurrent rolling convolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 5420–5428.
20. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 834–848. [CrossRef] [PubMed]
21. Sommer, L.W.; Schuchert, T.; Beyerer, J. Fast deep vehicle detection in aerial images. In Proceedings of the 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), Santa Rosa, CA, USA, 24–31 March 2017; pp. 311–319.
22. Lu, Y.; Javidi, T.; Lazebnik, S. Adaptive object detection using adjacency and zoom prediction. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 2351–2359.
23. Alexe, B.; Heess, N.; Teh, Y.W.; Ferrari, V. Searching for objects driven by context. In *Advances in Neural Information Processing Systems*; NIPS: Lake Tahoe, NV, USA, 2012; pp. 881–889.
24. Růžička, V.; Franchetti, F. Fast and accurate object detection in high resolution 4K and 8K video using GPUs. In Proceedings of the 2018 IEEE High Performance Extreme Computing Conference (HPEC), Waltham, MA, USA, 25–27 September 2018; pp. 1–7.
25. Yang, F.; Fan, H.; Chu, P.; Blasch, E.; Ling, H. Clustered Object Detection in Aerial Images. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Korea, 27 October–3 November 2019.

26. Zhao, Y.; Ma, J.; Li, X.; Zhang, J. Saliency detection and deep learning-based wildfire identification in UAV imagery. *Sensors* **2018**, *18*, 712. [CrossRef] [PubMed]
27. Sudhakar, S.; Vijayakumar, V.; Kumar, C.S.; Priya, V.; Ravi, L.; Subramaniyaswamy, V. Unmanned Aerial Vehicle (UAV) based Forest Fire Detection and monitoring for reducing false alarms in forest-fires. *Comput. Commun.* **2020**, *149*, 1–16. [CrossRef]
28. Chen, T.H.; Wu, P.H.; Chiou, Y.C. An early fire-detection method based on image processing. In Proceedings of the 2004 International Conference on Image Processing, 2004—ICIP'04, Singapore, 24–27 October 2004; Volume 3, pp. 1707–1710.
29. Çelik, T.; Özkaramanlı, H.; Demirel, H. Fire and smoke detection without sensors: Image processing based approach. In Proceedings of the 2007 15th European Signal Processing Conference, Poznan, Poland, 3–7 September 2007; pp. 1794–1798.
30. Mueller, M.; Karasev, P.; Kolesov, I.; Tannenbaum, A. Optical flow estimation for flame detection in videos. *IEEE Trans. Image Process.* **2013**, *22*, 2786–2797. [CrossRef] [PubMed]
31. Kamalakannan, J.; Chakrabortty, A.; Bothra, G.; Pare, P.; Kumar, C.P. Forest Fire Prediction to Prevent Environmental Hazards Using Data Mining Approach. In Proceedings of the 2nd International Conference on Data Engineering and Communication Technology, Pune, India, 15–16 December 2017; pp. 615–622.
32. Mahmoud, M.A.I.; Ren, H. Forest Fire Detection and Identification Using Image Processing and SVM. *J. Inf. Process. Syst.* **2019**, *15*, 159–168.
33. Zhang, Q.; Xu, J.; Xu, L.; Guo, H. Deep convolutional neural networks for forest fire detection. In Proceedings of the 2016 International Forum on Management, Education and Information Technology Application, Guangzhou, China, 30–31 January 2016; pp.568–575.
34. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*; NIPS: Lake Tahoe, NV, USA, 2012; pp. 1097–1105.
35. Sharma, J.; Granmo, O.C.; Goodwin, M.; Fidje, J.T. Deep convolutional neural networks for fire detection in images. In Proceedings of the International Conference on Engineering Applications of Neural Networks, Athens, Greece, 25–27 August 2017; pp. 183–193.
36. Muhammad, K.; Ahmad, J.; Lv, Z.; Bellavista, P.; Yang, P.; Baik, S.W. Efficient deep CNN-based fire detection and localization in video surveillance applications. *IEEE Trans. Syst. Man Cybern. Syst.* **2018**, *49*, 1419–1434. [CrossRef]
37. Muhammad, K.; Khan, S.; Elhoseny, M.; Ahmed, S.H.; Baik, S.W. Efficient fire detection for uncertain surveillance environment. *IEEE Trans. Ind. Inform.* **2019**, *15*, 3113–3122. [CrossRef]
38. Jadon, A.; Omama, M.; Varshney, A.; Ansari, M.S.; Sharma, R. FireNet: A Specialized Lightweight Fire & Smoke Detection Model for Real-Time IoT Applications. *arXiv* **2019**, arXiv:1905.11922.
39. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
40. Neubeck, A.; Van Gool, L. Efficient non-maximum suppression. In Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06), Hong Kong, China, 20–24 August 2006; Volume 3, pp. 850–855.
41. Everingham, M.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [CrossRef]
42. Lin, T.; Maire, M.; Belongie, S.J.; Bourdev, L.D.; Girshick, R.B.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. *arXiv* **2014**, arXiv:1405.031.