atmosphere

MDPI

Article

Automatic Fire and Smoke Detection Method for Surveillance Systems Based on Dilated CNNs

Yakhyokhuja Valikhujaev [1], Akmalbek Abdusalomov [1] and Young Im Cho [2,*]

[1] Department of IT Convergence Engineering, Gachon University, Sujeong-Gu, Seongnam-Si, Gyeonggi-Do 461-701, Korea; yakhyo9696@gmail.com (Y.V.); akmaljon@gachon.ac.kr (A.A.)
[2] Department of Computer Engineering, Gachon University, Sujeong-Gu, Seongnam-Si, Gyeonggi-Do 461-701, Korea
* Correspondence: yicho@gachon.ac.kr; Tel.: +82-010-3267-4727

Received: 9 October 2020; Accepted: 17 November 2020; Published: 18 November 2020

Abstract: The technologies underlying fire and smoke detection systems play a crucial role in ensuring and delivering optimal performance in modern surveillance environments. In fact, fire can cause significant damage to lives and properties. Considering that the majority of cities have already installed camera-monitoring systems, this encouraged us to take advantage of the availability of these systems to develop cost-effective vision detection methods. However, this is a complex vision detection task from the perspective of deformations, unusual camera angles and viewpoints, and seasonal changes. To overcome these limitations, we propose a new method based on a deep learning approach, which uses a convolutional neural network that employs dilated convolutions. We evaluated our method by training and testing it on our custom-built dataset, which consists of images of fire and smoke that we collected from the internet and labeled manually. The performance of our method was compared with that of methods based on well-known state-of-the-art architectures. Our experimental results indicate that the classification performance and complexity of our method are superior. In addition, our method is designed to be well generalized for unseen data, which offers effective generalization and reduces the number of false alarms.

Keywords: fire detection; smoke detection; deep learning; dilated convolution; classification

1. Introduction

Despite the rapid growth of technologies and smart systems, certain problems remain unsolved or are solved with methods that deliver poor performance. One of these problems is the unexpected outbreak of a fire, an abnormal situation that can rapidly cause significant damage to lives and properties. According to the Korea Statistical Information Service, the National Fire Agency recordedthat, during the three years from 2016 to 2018, 129,929 fires occurred in South Korea, resulting in 1020 deaths, 5795 injuries, and damage to properties estimated at USD 2.4 billion [1].

The latest technological advancements in sensors and sensing technologies have inspired businesses to determine whether these improvements can help to reduce the damage and harm caused by fire. This is the most frequent and widespread threat to public and social development as well as to individuals' lives. Although fire prevention is the top priority to ensure fires do not occur in the first place, it is nonetheless essential to spot fires and to extinguish them before they have serious consequences. In this regard, a large number of methods were introduced and tested for early fire detection to reduce the number of fire accidents and the extent of the damage. Accordingly, different types of detection technologies in automated fire alarm systems have been formulated and are widely implemented in practice.

Two types of fire alarm systems are known: traditional fire alarm systems and computer vision-based fire detection systems. Traditional fire alarm systems employ physical sensors such as thermal detectors, flame detectors, and smoke detectors. These kinds of sensing devices require human intervention to confirm the occurrence of a fire in the case of an alarm. In addition, these systems require different kinds of tools to detect fire or fumes and alert humans by providing the location of the indicated place and extent of the flames. Furthermore, smoke detectors are often triggered accidentally, as they are unable to differentiate between smoke and fire. Fire detection sensors require a sufficient intensity of fire for clear detection, which can extend the time taken for detection, resulting in extensive damage and loss. An alternative solution, which could improve the robustness and safety of fire detection systems, is the implementation of visual fire detection techniques. In this regard, many researchers have endeavored to overcome the abovementioned limitations by investigating the combination of computer vision-based methods and sensors [2,3]. A vision-based detector is advantageous in that it can overcome the shortcomings of sensor-based methods. In addition, this type of system has several advantages, such as scalability, manageability of installation, and it does not demand any closedowns. Moreover, the use of computer vision for surveillance applications has become an attractive research area in which notable advances have been made in the last few years. Vision-based approaches also overcome various limitations of traditional fire alarm systems, such as the need for surveillance coverage, human intervention, response time, and detailed reports of the fire with particulars such as its intensity, rate of spread, and extent. However, the complexity and false triggering for diverse reasons continue to remain problematic. Accordingly, studies have been conducted to investigate and address these issues related to computer vision-based technology. Initially, computer vision-based fire detection applications focused on edgedetection [3] or the color of the fire or smoke within the framework of rule-based systems. Rule-based systems are vulnerable to environmental conditions such as illumination, variation in lighting, perspective distortion, and inter-object occlusion. Solving the abovementioned problems using deep neural networks, such as convolutional neural networks (CNNs) and region-based CNNs, despite their robustness to lighting variations and different conditions, continue to present problems. In this case, without appropriate adjustment, a standard CNN is not effective under any possible circumstances. Creating a robust fire detection system requires sophisticated effort because of the dynamic and static behaviors of fire, smoke, and the large amount of domain knowledge that is required to solve the problem. Problems of this nature and extent could be solved by using machine/deep learning approaches [4,5]. However, solving these problems requires appropriately designed network architecture to be trained with a huge volume of data to eliminate the overfitting problem. The abovementioned smoke detection system [4] relies on machine-learning-based image recognition software and a cloud-based workflow capable of scanning hundreds of cameras every minute.

In this study, we addressed the aforementioned issues by structuring a convolutional layer that uses a dilated convolution operator to detect a fire or smoke in a scene. The advantage of this model is that it can reduce false fire detections and misdetections. For this work, we collected a number of images containing diverse scenes of fire and smoke to enhance the capability of the fire and smoke detection model to generalize unseen data. In other words, the utilization of various fire and smoke images helps to make our approach more generalizable for unseen data. We used one subset of data for the learning process and evaluated it on a different subset. Several similar methods already exist, for example that proposed by Abdulaziz and Cho [6], who implemented adaptive piecewise linear units (APL units). However, when we used their method to process the specific dataset we constructed, the experiments showed that our proposed method is more effective than theirs in terms of complexity and accuracy. The majority of the relevant researchers used common CNN architectures to compare their work, such as AlexNet [7], VGG16, and VGG19 [8]. Therefore, we evaluated our method in comparison to the abovementioned deep neural network structures.

Accordingly, the following points outline our key contributions:

(1) We propose a CNN-based approach that uses a dilated CNN to eliminate the time-consuming efforts dedicated to introducing handcrafted features because our method automatically extracts a group of practical features to train it. Asit is essential to use a sufficient amount of data for the training process, we assembled a large collection of images of different scenes depicting fire and smoke obtained from many sources. Images were selected from a well-known dataset [9]. Our dataset is also available for further research.

(2) We used dilated convolutional layers to build our network architecture and briefly explain the principles thereof. Dilated convolution makes it possible to avoid learning much deeper, because it helps to learn larger features by ignoring smaller features.

(3) Small window sizes are used to aggregate valuable values from fire and smoke scenes. The use of smaller window sizes in deep learning is known to enable smaller but complex features in an image to be captured, and it offers improved weight sharing. Therefore, we decided to use a smaller kernel size for the training process.

(4) We determined the number of layers that are well suited to solve this task. Four convolutional layers were employed because an excessive number of layers allow the model to learn much deeper. This approach considers that, rather than having to classify a very large number of classes, the task is a simple binary classification. Therefore, employing many layers will exacerbate the overfitting problem. In Section 5, overfitting is demonstrated to occur. However, the latter studies used a larger number of layers, mostly six layers [6].

The remainder of this paper is organized as follows. In Section 2, information about fire and smoke detection approaches is introduced. The features of our custom dataset are presented in Section 3. A comprehensive explanation of our proposed method is provided in Section 4. In Section 5, we discuss all the experimental results. Section 6 highlights a few limitations of the proposed method. Finally, Section 7 concludes the manuscript with final remarks.

2. Related Work

2.1. Computer Vision Approaches for Fire and Smoke Detection

Many researchers who studied traditional fire and smoke detection systems focused on extracting crucial features from images. Many of these investigations focused on detecting geometrical characteristics of flames [10,11] and fires in the images [12,13]. For example, Bheemul et al. [11] suggested an efficient approach for extracting edges by detecting changes in the brightness of an image of a fire. Jian et al. [13] presented an enhanced edge detection operator, a Canny edge detector, which uses a multi-stage algorithm. However, the abovementioned computer vision-based methods were only applicable to images of simple and steady fires and flames. Other researchers applied new methods based on FFT (Fast Fourier Transform) and wavelet transform to analyze the contours of forest fires in video scenes [14]. Previous research has indicated that these approaches are suitable only under certain conditions.

Changes in fires were analyzed using red-green-blue (RGB) and (hue, saturation, intensity) HSI color models. For example, Chen [15] used a color-based approach to detect the discrepancy among sequential images. Celik et al. [16] proposed a generic rule-based approach that uses the YCbCr color space to discriminate luminance from chrominance to identify a variety of smoke and fires in images. Yu et al. [17] also used simultaneous motion and color features for detection purposes. The use of YCbCr can increase the detection rate of fire in images compared to RGB, because it can separate luminance more effectively than RGB color space [18]. However, color-based fire and smoke detection methods are not feasible, because these approaches are not independent from environmental factors such as lighting, shadows, and other distortions. In addition, color-based approaches are vulnerable to the dynamic behavior of fire and smoke, even though fire and smoke have a longer-term dynamic behavior.

The disadvantage of these methods is that they require specific knowledge to extract and explore the features of fire and smoke in images. In addition, almost all conventional fire detection methods

use color-based, edgedetection, or motion-based techniques, and these approaches are infeasible for analyzing tiny and noisy images. Therefore, these methods are limited, because they rely on limited characteristics of fire and smoke in images such as the motion, color, and edge of the fire or smoke. Furthermore, extracting these characteristics is also challenging because of the quality of the video or image.

2.2. Deep Learning Approaches for Fire and Smoke Detection

In recent years, deep learning has emerged significantly because of advances in hardware, the ability to process large-scale data, and substantial advances in the design of network structures and training strategies. Additionally, deep learning has been effectively implemented in various fields such as natural language processing (NLP), network filtering, games, medicine, and vision. Several deep learning applications have been shown to outperform human experts in certain cases [7,19,20]. In vision-related tasks, computers have already achieved human-level performance. Several studies have been carried out to detect fire and smoke in images using deep learning approaches to enhance the reliability and results of these methods.

These approaches for fire and smoke detection differ from those based on computer vision in various ways. First, deep learning performs automatic feature extraction using a massive amount of data for training and discriminative features learned by the neural network to detect a fire or smoke. Another advantage is that deep neural networks can be flexibly and successfully implemented in various fields, and instead of spending time on feature extraction, they can be changed to construct a robust dataset and appropriate network structure.

Recently, Abdulaziz [6] introduced a fire and smoke detection network with limited data based on CNNs and used it with a generative adversarial network (GAN) [21] for augmentation purposes. Instead of using the traditional activation function, Abdulazizet al. employed adaptive piecewise linear units as an activation function. Abdulaziz [6] conducted a number of experiments to show an increase in detection. Sebastien et al. [22] also suggested a model that uses a multilayer perceptron-type neural network to learn features by an iterative process of learning. In addition, Muhammad et al. [23] experimented with different fine-tuned versions of various CNN models, such as AlexNet [7], SqueezeNet [24], GoogleNet [25], and MobileNetV2 [26]. Our proposed model, which allows fire scenes to be semantically understood, is based on the SqueezeNet architecture. However, the abovementioned deep-learning-based models improved the fire detection accuracy, with minimum false alarms, but the complexity and size of the model are comparatively large, that is, 238MB [23]. All of these studies utilized Foggia's dataset [9] as the main source of their training data.Ba et al. [27] proposed a new convolutional neural network (CNN) model, SmokeNet, which incorporates spatial and channel-wise attention in CNN to enhance feature representation for scene classification. In this study, we proved that using a small kernel size and a small number of layers can improve the performance and generalizability of the current task. In fact, by conducting a number of experiments, we proved that this approach could overcome the overfitting problem for a small number of data samples.

In the image/video classification fields, CNN has outpaced and showed superior performance compared with other approaches because of its powerful feature extraction techniques and robust model structure. Consequently, in terms of performance, traditional computer vision methods are being replaced by deep learning methods. Our proposed method adopts a model to classify fire or smoke in images/videos. Misclassification of images or videos leads to an increase in false fire alarms because of variations in perspective distortions, shadows, and brightness. We detected images showing fire and smoke using a model based on dilated CNNs to learn and extract the robust features of a frame.

3. Dataset

One of the main limitations of vision-related tasks is the insufficiency of robust data for evaluating and analyzing the suggested method. To find a suitable dataset, we examined datasets that were used in prior studies. One of the datasets provided by Foggia et al. [9] contains fourteen fire and

seventeen non-fire videos. However, the diversity of this video data is insufficient to be suitable for training, and we cannot expect it to deliver good fire detection performance in realistic scenarios. Thus, we attempted to create a diverse dataset by extracting frames from fire and smoke videos and collecting images from internet sources. Our training set consists of fire images sampled from Foggia's dataset and from images on the internet. Images of smoke taken from different internet sources diversified our dataset. We extracted frames from videos and randomly sampled a few images from each video to build our final fire-smoke dataset for use in this study. Table 1 contains information on the number of fire and smoke images in our dataset.

Table 1. Distribution of images in the dataset.

Dataset	Fire Images	Smoke Images	Total
Our dataset	8430	8430	16,860

Examples of images that were collected are shown in Figure 1. The red-green-blue (RGB) images are stored in JPG format and the images are sized 100 × 100 pixels.

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

Figure 1. Examples of images showing (**a**) fire; (**b**) smoke.

4. Proposed Architecture

4.1. Brief Summary of Well-Known Network Architectures

We propose a novel model for fire and smoke detection. The model is constructed on the basis of dilated convolutions, and its performance was evaluated with respect to the following well-known architectures: AlexNet [7], VGGNet [6], ResNet50 [28], and Inception V3 [29]. In 2012, Alex Krizhevsky published work [7] that was a turning point for vision-related tasks in deep learning. This work was an advanced variant of LeNet [30] and became a winner of the ImageNet LSVRC-2012 [31] competition. The AlexNet model is constructed with five convolutional layers, and a max-pooling operation is applied after the convolutional layers. The output of the last two fully connected layers feeds data into thousand-way units to produce a probability distribution among thousands of classes of labels. The success of AlexNet started a revolution in deep learning, and then, VGGNet and the inception architecture of GoogLeNet achieved similarly high performance in the ImageNet LSVRC-2014 [31] classification challenge, where VGGNet scored second place after GoogLeNet. In 2015, ResNetintroduced a "bottleneck" architecture that employs skip connections to fit the input from the

previous layer to the next layer without changing it. Therefore, it enabled a deeper network and became the winner of ImageNet LSVRC-2015 [31] as well as the winner of MS COCO 2015 [32]. Although the aforementioned network models perform more efficiently on the current issue, these network models are too deep for our two-class classification task. This motivated us to build a model with high robustness by emphasizing the extraction of useful and specific characteristics of images. In our case, it is necessary to detect whether fire or smoke appears in a given image.

4.2. Dilated Convolution

The purpose of utilizing convolutions is to aggregate learnable features from the input images. In computer vision, there are several different filters to extract features for convolutions. Each type of filter is responsible for extracting different aspects or features from the input data, for example horizontal, vertical, and diagonal edges. Correspondingly, CNN uses convolutional layers to extract different features using various filters whose weights are spontaneously updated at the time of the learning process. All the extracted or learned features are then merged to make decisions concerning the input data. In addition, convolution takes the spatial relationship of pixels into consideration, and this is helpful especially in computer vision tasks. In a recent development [31], an additional hyper parameter referred to as dilation was introduced to the convolutional layer, as illustrated in Figure 2. The convolution operator is adapted to apply the filters in a different manner in convolutional layers. The modified version of the convolution operator is referred to as the dilated convolution operator. The standard (vanilla) convolution is shown in Figure 3. Equation (1) expresses the standard convolution and Equation (2) the dilated convolution.

$$(F * k)(p) = \sum_{s+t=p} F(s)k(t) \tag{1}$$

$$(F * k_l)(p) = \sum_{s+lt=p} F(s)k(t) \tag{2}$$

It is clear that, in summation, $s + lt = p$ indicates that certain points are skipped during convolution. Furthermore, dilated convolution allows the network to obtain more information from the context and requires less computational time with fewer parameters, and it allows the model to execute faster than a model that uses normal convolution. A common use of dilated convolutions is image segmentation, where each pixel is labeled by its corresponding category. Therefore, the network output needs to have the same size as the input image.

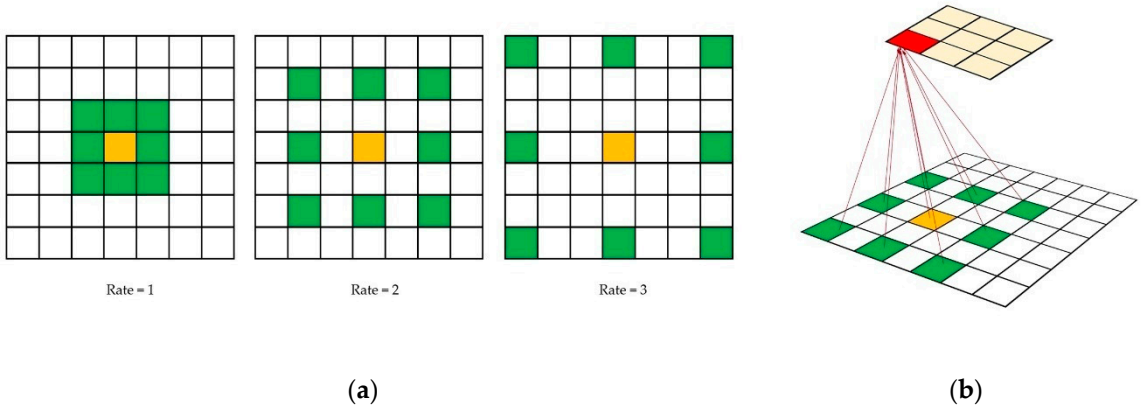
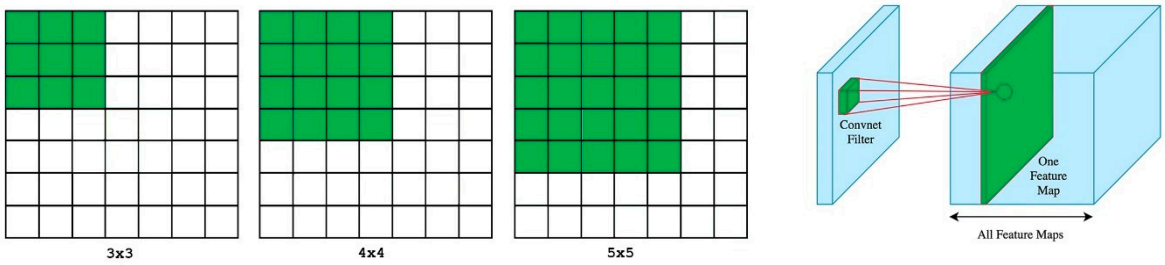


Figure 2. (**a**) Dilated convolutions. (**b**) A dilation rate of one is normal (vanilla) convolution.



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

Figure 3. (**a**) Standard (vanilla) convolutions. (**b**) 3D view of normal convolution operation.

This is the origin of the concept of employing convolutions with a dilation rate to solve the described problem. An excellent idea proposed by René et al. [33] is that of multi-scale context aggregation. Convolutional layers with the dilation rate have been implemented in various fields, such as text-to-speech [34] and text interpretation [35]. These methods used dilated convolutions to aggregate multi-scale context features from the input with fewer parameters. The former of these two methods employs dilated convolutions to generate speech and music from a raw audio waveform. Moreover, this method is implemented to recognize speech from a raw audio waveform.

4.3. Proposed Network Architecture

As mentioned earlier, our task is not a classification of 1000 groups; hence, we built a model with fewer layers, as shown in Figure 4.

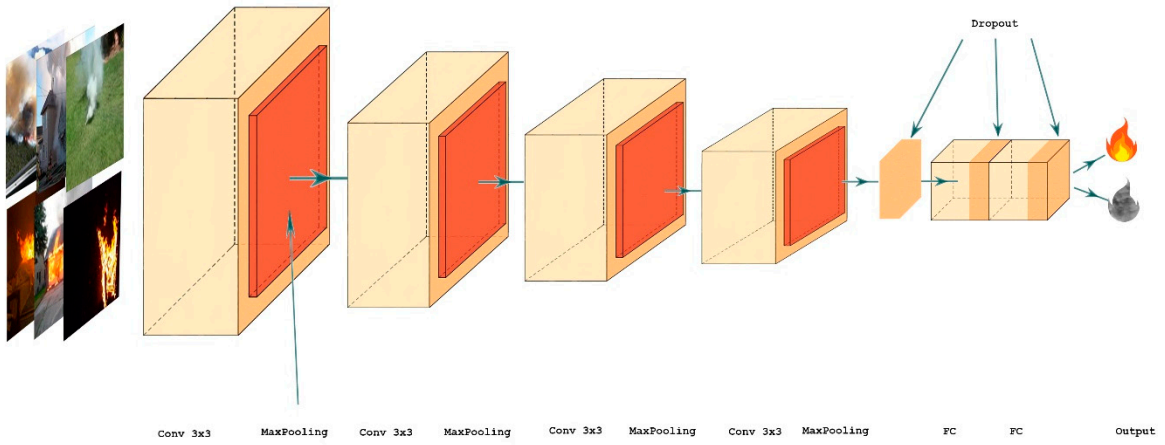


Figure 4. Structure of the network based on dilated convolutional neural networks.

All convolutional layers in this architecture use small receptive field sizes (3×3) and dilated convolutions with a rate of 2 are employed. The fourth convolutional layer is followed by two fully connected layers with 2024 nodes and a final output layer with two nodes. The architecture of the proposed method is provided in Table 2. The input layer takes input data with a fixed shape of $100 \times 100 \times 3$ ($width \times height \times color\ channel$), and all data points are resized to fit the given shape.

The output shape of the first convolutional layer is $96 \times 96 \times 128$. The calculation of the feature ma is given by Equation (3) [36]. In this equation, ($width \times height$) is the input shape, ($F_{width} \times F_{height}$) is the filter size, S_{width} and S_{height} are the stride, and P is the padding (it is chosen as "valid" padding in our case):

$$Output_{width} = \frac{width - F_{width} + 2P}{S_{width}} + 1 Output_{height} = \frac{height - F_{height} + 2P}{S_{height}} + 1 \quad (3)$$

We employed a rectified linear unit (ReLU) [37] as the activation function after all four convolutional layers. The mathematical form of the rectified linear unit is expressed as in Equation (4). The advantage of a ReLU is that its processing speed is higher than those of other nonlinear activation functions; in addition, a ReLU does not experience the gradient vanishing problem, because the gradient of the ReLU function is either 0 or 1, which means it never saturates, and so the gradient vanishing problem does not occur.

$$y = \max(0, x) \tag{4}$$

Table 2. Layered network structure.

Layer Type	Filters	Feature Map	Kernel Size	Stride
Input layer		$100 \times 100 \times 3$		
1st convolutional layer	128	$96 \times 96 \times 128$	$3 \times 3 \times 3$	1×1
Max-pooling layer	-	$32 \times 32 \times 128$	2×2	3×3
2nd convolutional layer	256	$32 \times 32 \times 256$	$3 \times 3 \times 3$	1×1
Max-pooling layer	-	$16 \times 16 \times 256$	2×2	-
3nd convolutional layer	512	$16 \times 16 \times 512$	$3 \times 3 \times 3$	1×1
Max-pooling layer	-	$8 \times 8 \times 512$	2×2	-
4th convolutional layer	512	$8 \times 8 \times 512$	$3 \times 3 \times 3$	1×1
Max-pooling	-	$4 \times 4 \times 512$	2×2	-
Dropout		-		
1st fc layer		2048		
Dropout		-		
2nd fc layer		2048		
Dropout		-		
Classification(output)layer		2		

After each convolutional layer, we employed a max-pooling layer for down sampling purposes. Max pooling has been proved to be more effective than average pooling for computer vision tasks such as classification, segmentation, and object detection. Our proposed method functions by increasing the number of filters by a factor of 2 until the 4th convolutional layer. At the initial layer, 128 kernels are employed with a dilation rate of 2. The following layer is formed of 256 kernels that is double the first convolutional layer. The third and fourth layers have the same depth, that is, 512 filters. A common problem in computer vision is that of over fitting. To prevent the overfitting problem, we use dropout regularization [38] after the final convolutional and each fully connected layer. AlexNet [7] additionally employs local response normalization that normalizes over local input regions. Our network architecture is shallower than that of AlexNet, and the amount of data used to train the model is considerably smaller. Therefore, the application of any normalizing technique might lead to the loss of the essential relationship between data points. Eventually, we employed sigmoid activation as the activation function, as presented in Equation (5), to indicate the probability of the evaluation result.

$$y = \frac{1}{1 + e^{-x}} \tag{5}$$

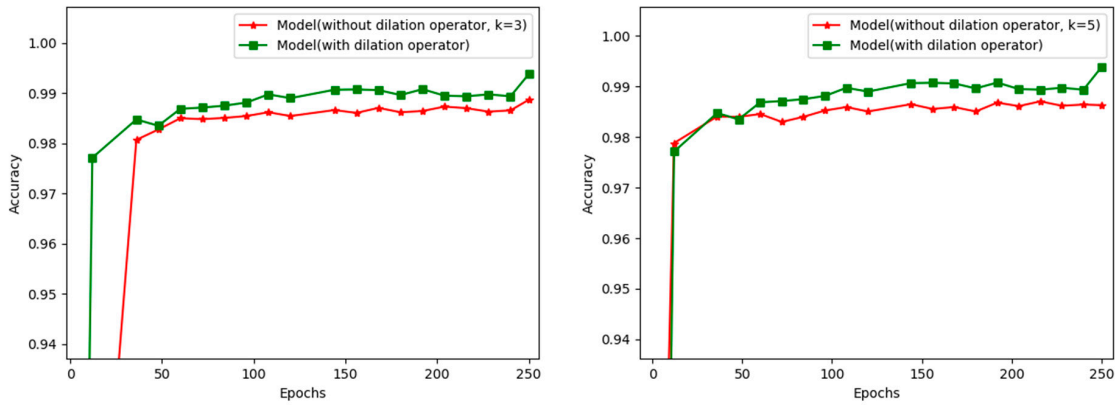
We trained our model using Keras, a high-level API of the TensorFlow framework, in our experiments. Keras is an open-source neural network library written in Python. The model was trained on a workstation with a 3.4 GHz AMD Ryzen Thread ripper 1950X 16-Core Processor and an NVDIA GeForce GTX 1080Ti GPU with 11 GB of memory. During training, data augmentation techniques were also used, and we set the number of epochs and batch size to 250 and 64, respectively.

We employed a stochastic gradient descent algorithm (SGD) [39] to optimize the training process and set the parameters as follows: initially, we set the momentum to 0.99, the learning rate was 10^{-5}, $l2$, and regularizationwas 5×10^{-4}. We used 80% of the data to train the model, and the remainder of the data to evaluate the model performance.

5. Experiments and Discussion

5.1. Investigating the Optimum Method for Fire and Smoke Detection

To analyze the efficiency of the model, we carried out extensive attempts to select the appropriate kernel size, dilation rate, and number of convolutional layers. We used the well-known machine learning library, Keras, built on top of TensorFlow. Initially, we compared two neural network models without dilation and with dilated convolutional layers. Figure 5 provides an indication of the training accuracy.



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

Figure 5. Comparison of training accuracies when kernel size equals (**a**) 3 and (**b**) 5, respectively. The x- and y-axes represent the number of epochs (250) and the accuracy, respectively.

Figure 5a,b show that, after the final epochs, the training accuracy for the model (without dilated convolutional layers) with kernel sizes of three and five was 98.86% and 98.63%, respectively. Compared with the other two models, the model with dilated convolutional layers delivered higher performance on training with 99.60% accuracy. The training and testing accuracies of these networks are provided in Table 3. These results indicate that the training and testing accuracies of the network that implements dilated convolutional layers are higher than those of the other models. One of the contributions of our study is the use of dilated convolutions, as we previously mentioned. We carried out a number of experiments to prove the advantages of using convolutional layers to which dilation is applied instead of using no dilation, as shown in Figure 5. As mentioned previously, a dilation operator is adapted to predict each label for each pixel in the images, because it has the capability of expanding the receptive field without losing coverage.

Table 3. Comparison of training and testing performance of the models.

Method	Training Scores	Testing Scores
Model (without dilation operator, k =3)	98.86%	97.53%
Model (without dilation operator, k = 5)	98.63%	95.81%
Model (with dilation operator)	99.3%	99.06%

We experimented on models by changing the number of layers to identify the model that performs the best on this task. Figure 6 compares the performance of the model by varying the number of

convolutional layers. At first sight, it is obvious which model has the highest accuracy, especially when comparing the models with three and five convolutional layers, the performance of which is similar. According to Figure 6, the plotted line for the model using four layers indicates the highest accuracy relative to the other three models.

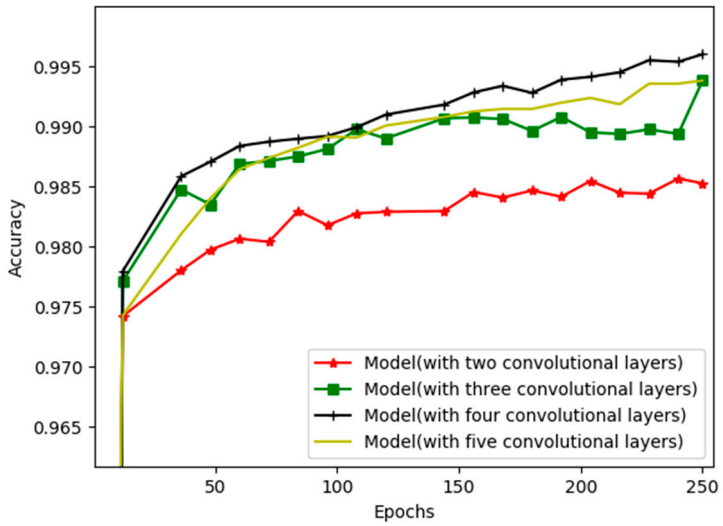


Figure 6. Comparison of the models with two, three, and four layers, respectively.

With minor variance, the model with four convolutional layers delivers the best performance. Accordingly, the lowest scores belong to the model two convolutional layers. The training and testing scores for the model with two convolutional layers are 98.52% and 98.03%, respectively, whereas the training scores for the models with three and five convolutional layers are 99.38% and 99.36%, respectively. These results are provided in Table 4. However, the generalization ability of neural networks with three and five convolutional layers is slightly lower than those of the model with four layers. Thus, we demonstrated the training and testing accuracy of our proposed method by making use of four convolutional layers.

Table 4. Comparison of training and testing scores of the models.

Method	Training Scores	Testing Scores
Model (with two convolutional layers)	98.52%	98.03%
Model (with three convolutional layers)	99.38%	99.06%
Model (with four convolutional layers)	99.60%	99.53%
Model (with five convolutional layers)	99.36%	98.07%

We mentioned above that employing small kernel sizes in dilated convolutional layers might assist the performance of models. Although the exact size of kernels that perform optimally on this task was not known for us, we conducted several experiments to find the optimal kernel size. The selected kernel size proved to be the best option for solving this problem. The performance of the models is compared in Figure 7.

We started by experimenting with training the model by using different kernel sizes from 3×3 to 13×13. The plotted lines illustrating the training scores of kernel sizes 11 and 13 indicate the lowest training scores along with lower testing scores. At the same time, the results for the performance of our model when the kernel size equals seven demonstrate average performance, as shown in Figure 7. The training scores for models that employ smaller kernel sizes are higher, and thus, the models are more efficient in terms of both training and evaluation.

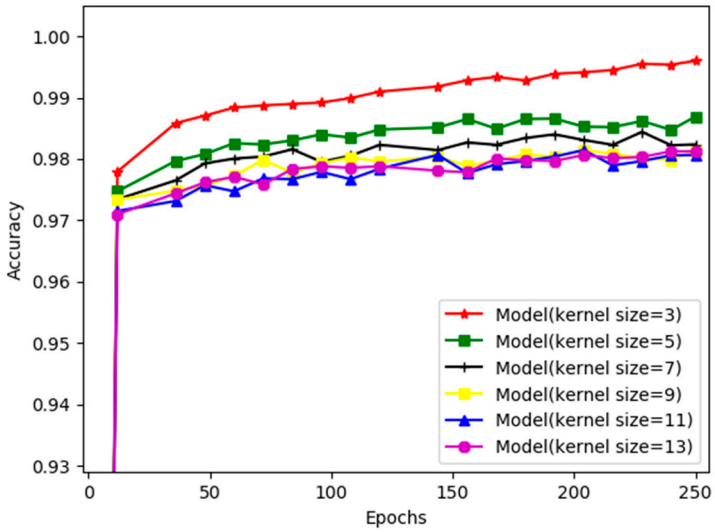


Figure 7. Experimenting with various kernel sizes to construct the optimal neural network. The x- and y -axes represent the number of epochs (250) and training accuracy, respectively.

The testing and training accuracies of the models with kernel sizes of 3, 5, 7, 9, 11, and 13 are summarized in Table 5. However, it is difficult to differentiate between the performances of the respective models. Furthermore, the models with smaller kernel sizes performed more accurately for the purpose of our task. We found a kernel size of 3×3 to be the most appropriate option to solve the fire and smoke detection problem.

Table 5. Comparing the training and testing scores of models using various kernel sizes.

Method	Training Scores	Testing Scores
Model (kernel size = 3)	99.60%	99.53%
Model (kernel size = 5)	98.69%	98.07%
Model (kernel size = 7)	98.23%	98.83%
Model (kernel size = 9)	98.13%	98.31%
Model (kernel size = 11)	98.06%	98.19%
Model (kernel size = 13)	98.12%	97.95%

5.2. Comparison of Our Network Model with Well-Known Architectures by Conducting Experiments on Our Dataset

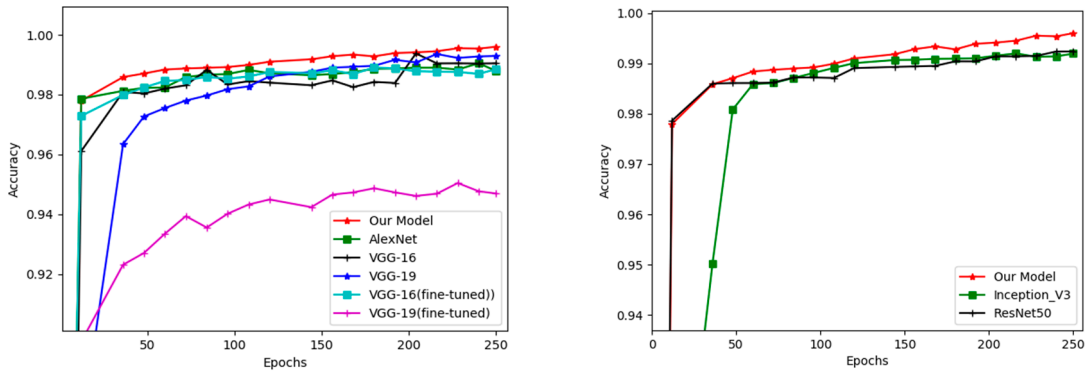
Our experiments mainly aimed to evaluate the performance of our proposed model against renowned deep learning models, such as VGGNet, AlexNet, ResNet, and Inception V3, all of which performed exceptionally well with respect to classification in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC). We compared the performance of our method in terms of accuracy. The training accuracies of these networks are shown in Figure 8.

As indicated in Figure 8, higher training accuracies were obtained by deeper models. Table 6 provides the detailed training and testing scores of all the models. We trained VGG16, VGG19, and their fine-tuned versions to evaluate their performance against our custom dataset. The results in Table 6 lead to a few conclusions. The highest training and testing scores of 99.6% and 99.53%, respectively, were achieved by our proposed network model, whereas the scores for VGG19 (fine-tuned) were the lowest, i.e., 94.6% and 94.88%, respectively. However, the performance of VGG16 was also higher, even though it is a less deep network in our experiments. The highest performance accuracies were obtained by the Inception V3 and ResNet50 network architecture. We additionally calculated other metrics, such as theF1-score, precision, and recall. The F1 score is the weighted average of precision and recall. Hence, this score considers both false positives and false negatives. Intuitively, it is not as

easy to understand as the accuracy, but F1 is more commonly used than accuracy. The accuracy is best used when the false positives and false negatives have similar costs. If the cost of these two metrics differs, it is more useful to consider both precision and recall. Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. Recall is the ratio of correctly predicted positive observations to all observations in the actual class, as shown in Equation (6). As indicated in Table 6, the F1 score of the proposed method is the highest overall, with a lower score on recall and precision. In particular, the F1 score of the proposed method is 0.9892 compared with the lowest result, which was recorded by AlexNet, of 0.7513.We calculated the precision and recall rates as follows:

$$Precision = \frac{TP}{TP + FP} Recall = \frac{TP}{TP + FN} F1 = \frac{2 \times precision \times recall}{precision + recall} \quad (6)$$

where TP denotes the number of true positives, FP the number of false positives, and FN the number of false negatives. Precision is defined as the number of true positives over the number of true positives plus the number of false positives. Recall is defined as the number of true positives over the number of true positives plus the number of false negatives.



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

Figure 8. Comparing the performance of the models (**a**, **b**). The x- and y -axes show the number of epochs (250) and the training accuracies of the models, respectively.

Table 6. Training and testing scores of our network against those of well-known networks.

Method	Training Scores	Testing Scores	F1-Score	Recall	Precision
Our Model	99.60%	99.53%	0.9892	0.9746	0.9827
Inception V3 [29]	99.19%	98.31%	0.9744	0.9980	0.9532
AlexNet [7]	98.78%	86.74%	0.7513	0.6131	0.7332
ResNet [28]	99.23%	98.79%	0.9425	0.9364	0.9486
VGG16 [8]	99.04%	98.67%	0.9278	0.8799	0.875
VGG19 [8]	99.29%	98.37%	0.9206	0.8566	0.9949
VGG16 (fine-tuned)	98.85%	98.76%	0.8754	0.8215	0.9368
VGG19 (fine-tuned)	94.6%	94.88%	0.8548	0.887	0.8248

In fact, the computational cost of deeper models, such as AlexNet, VGGNet, and ResNet50, is high, and they require much more computational time than models that are not as deep. For example, the parameters of the AlexNet and VGGNet architectures are 60 M and 138 M, respectively. However, our proposed model has only 24 M parameters. Our experiments proved that, for our two-class classification task, the generalization of much deeper networks was lower. For instance, the well-known ResNet50 and Inception V3 network architectures do not generalize well to our custom-built dataset. Moreover, deeper networks are more complex and, thus, affect the training time and prediction time.

The deeper networks were much more time consuming for training as well as for prediction (Table 7). The prediction time in the table is in seconds, and the results reflect the complexity of the model. Our model spent less time predicting the entire test set, namely 1.9 s; in contrast, Inception V3 had the longest time of all the models, 8.7 s.

Table 7. Comparing the time required for training and prediction.

Method	Training Time (hh:mm:ss)	Prediction Time for Test Set (s)
Our Model	2:15:00	1.9
Inception V3 [29]	10:20:00	8.7
AlexNet [7]	9:26:40	7.9
ResNet [28]	10:00:00	8.5
VGG16 [8]	3:20:00	2.4
VGG19 [8]	4:43:20	3.1

6. Limitations

The proposed method may make errors in the early stages when the pixel values in the fire and smoke images are very close to those of the background. Our method mainly experiences this problem when the weather is cloudy. In an attempt to overcome this problem, we are currently experimenting with datasets containing satellite imagery of smoke (USTC_SmokeRS), which consist of RGB images from more complex land covers. In our research area, dataset images play a significant role in smoke scene detection.

7. Conclusions

We presented new robust deep learning model architecture for classifying fire and smoke images captured by a camera or nearby surveillance systems. The proposed method is fully automatic, requires no manual intervention, and is designed to be well generalizable for unseen data. It offers effective generalization and reduces the number of false alarms. Based on the proposed fire detection method, our contributions include the following four main features: the use of dilation filters, a small number of layers, small kernel sizes, and a custom-built dataset, which was used in our experiments. This dataset is expected to be a useful asset for future research that requires images of fire and smoke. However, we are far from concluding that this is the best solution for this task, because all experiments were conducted on our custom dataset. We verified our method experimentally by conducting several experiments to demonstrate that employing a dilation operator and a small number of layers can boost the performance of the method by extracting valuable features. Moreover, using a small number of layers and less deep networks would allow the model to be used in devices with low computational power. During the experiments, we assessed the performances and generalizing abilities of well-known CNN architectures in comparison with those of our proposed method. The experimental results proved that the performance of our proposed method on our dataset was slightly superior to that of well-known neural network architectures.

Our future projection is to build a lightweight model with robust detection performance that would allow us to set up embedded devices, which have low computational capabilities.

Author Contributions: This manuscript was designed and written by Y.V.; Y.V. conceived the main idea of this study; Y.V. wrote the program in Python and performed all the experiments; A.A. and Y.I.C. supervised this study and contributed to the analysis and discussion of the algorithm and experimental results. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the MSIT, Korea, under the ITRC support program (IITP-2020-2017-0-01630), the NRF (project number is 2018R1D1A1A09084151) and the Korea Agency for Infrastructure and Transport (Grant 20DEAP-B158906-01) and the Gachon University research fund of 2019 (GCU-2019-0794).

Acknowledgments: I would like to express my sincere gratitude and appreciation to the supervisor, Young Im Cho (Gachon University) for her support, comments, remarks, and engagement over the period in which this manuscript was written.

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

References

1. Lim, J.S.; Lim, S.W.; Ahn, J.H.; Song, B.S.; Shim, K.S.; Hwang, I.T. New Korean reference for birth weight by gestational age and sex: Data from the Korean Statistical Information Service (2008–2012). *Ann. Pediatr. Endocrinol. Metab.* **2014**, *19*, 146–153. [CrossRef]
2. 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]
3. Liu, C.B.; Ahuja, N. Vision Based Fire Detection. In Proceedings of the 17th International Conference on Pattern Recognition (ICPR), Cambridge, UK, 26 August 2004; pp. 134–137.
4. Govil, K.; Welch, M.L.; Ball, J.T.; Pennypacker, C.R. Preliminary Results from a Wildfire Detection System Using Deep Learning on Remote Camera Images. *Remote. Sens.* **2020**, *12*, 166. [CrossRef]
5. Pan, H.; Badawi, D.; Cetin, A.E. Computationally Efficient Wildfire Detection Method Using a Deep Convolutional Network Pruned via Fourier Analysis. *Sensors* **2020**, *20*, 2891. [CrossRef]
6. Namozov, A.; Cho, Y.I. An Efficient Deep Learning Algorithm for Fire and Smoke Detection with Limited Data. *Adv. Electr. Comput. Eng.* **2018**, *18*, 121–128. [CrossRef]
7. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*; Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q., Eds.; Curran Associates Inc.: Red Hook, NY, USA, 2012; pp. 1097–1105.
8. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In Proceedings of the 3rd International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015.
9. 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]
10. Lu, G.; Gilabert, G.; Yan, Y. Vision based monitoring and characterization of combustion flames. *J. Phys. Conf. Ser.* **2005**, *15*, 194–200. [CrossRef]
11. Bheemul, H.C.; Lu, G.; Yan, Y. Three-dimensional visualization and quantitative characterization of gaseous flames. *Meas. Sci. Technol.* **2002**, *13*, 1643–1650. [CrossRef]
12. Ko, B.C.; Cheong, K.-H.; Nam, J.-Y. Fire detection based on vision sensor and support vector machines. *Fire Saf. J.* **2009**, *44*, 322–329. [CrossRef]
13. Jiang, Q.; Wang, Q. Large space fire image processing of improving canny edge detector based on adaptive smoothing. *Proc. Int. CICC-ITOE* **2010**, *1*, 264–267. [CrossRef]
14. Zhang, Z.; Zhao, J.; Zhang, D.; Qu, C.; Ke, Y.; Cai, B. Contour Based Forest Fire Detection Using FFT and Wavelet. In Proceedings of the 2008 International Conference on Computer Science and Software Engineering, Hubei, China, 12–14 December 2008; Volume 1, pp. 760–763.
15. Chen, T.H.; Wu, P.H.; Chiou, Y.C. An early fire-detection method based on image processing. In Proceedings of the International Conference on Image Processing (ICIP), Singapore, 24–27 October 2004; pp. 1707–1710.
16. Celik, T.; Demirel, H.; Ozkaramanli, H.; Uyguroglu, M. Fire detection using statistical color model in video sequences. *J. Vis. Commun. Image Represent.* **2007**, *18*, 176–185. [CrossRef]
17. Chun-yu, Y.; Fang, J.; Jin-jun, W.; Zhang, Y. Video Fire Smoke Detection Using Motion and Color Features. *Fire Technol.* **2009**, *46*, 651–663. [CrossRef]
18. Zaidi, N.; Lokman, N.; Daud, M.; Chia, K. Fire recognition using RGB and YCbCr color space. *ARPN J. Eng. Appl. Sci.* **2015**, *10*, 9786–9790.
19. Google's AlphaGo AI wins three-match series against the world's best Go player. TechCrunch. Available online: https://techcrunch.com/2017/05/23/googles-alphago-ai-beats-the-worlds-best-human-go-player/ (accessed on 25 May 2017).
20. Ciresan, D.; Meier, U.; Schmidhuber, J. Multi-column deep neural networks for image classification. 2012 IEEE Conference on Computer Vision and Pattern Recognition. *arXiv* **2012**, arXiv:1202.02745, 3642–3649. [CrossRef]
21. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Networks (PDF). In Proceedings of the International Conference on Neural Information Processing Systems (NIPS 2014), Montréal, Canada, 8–13 December 2014; pp. 2672–2680.

22. Frizzi, S.; Kaabi, R.; Bouchouicha, M.; Ginoux, J.M.; Moreau, E.; Fnaiech, F. Convolutional neural network for video fire and smoke detection. In Proceedings of the IECON 2016—42nd Annual Conference of the IEEE Industrial Electronics Society, Florence, Italy, 23–26 October 2016; IEEE: Piscataway, NJ, USA; pp. 877–882.
23. Muhammad, K.; Ahmad, J.; Baik, S.W. Early fire detection using convolutional neural networks during surveillance for effective disaster management. *Neurocomputing* **2018**, *288*, 30–42. [CrossRef]
24. Iandola, F.N.; Han, S.; Moskewicz, M.W.; Ashraf, K.; Dally, W.J.; Keutzer, K. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size. *arXiv* **2016**, arXiv:1602.07360.
25. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Computer Society Conference onComputer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
26. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.-C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
27. Ba, R.; Chen, C.; Yuan, J.; Song, W.; Lo, S. SmokeNet: Satellite Smoke Scene Detection Using Convolutional Neural Network with Spatial and Channel-Wise Attention. *Remote. Sens.* **2019**, *11*, 1702. [CrossRef]
28. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.
29. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.
30. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
31. Olga, R.; Jia, D.; Hao, S.; Jonathan, K.; Sanjeev, S.; Sean, M.; Zhiheng, H.; Andrej, K.; Aditya, K.; Michael, B.; et al. ImageNet Large Scale Visual Recognition Challenge. *IJCV* **2015**, *115*, 211–252.
32. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common objects in context. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; DCFS: Tallahassee, FL, USA, 2014; pp. 740–755.
33. René, C.L.; Flynn, M.D.; Vidal, R.; Reiter, A.; Hager, G.D. Temporal Convolutional Networks for Action Segmentation and Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 22–25 July 2017; pp. 156–165.
34. Van Den Oord, A.; Dieleman, S.; Zen, H.; Simonyan, K.; Vinyals, O.; Graves, A.; Kalchbrenner, N.; Senior, A.; Kavukcuoglu, K. Wavenet: A Generative Model for Raw Audio. In Proceedings of the 9th ISCA Speech Synthesis Workshop, Sunnyvale, CA, USA, 13–15 September 2016; p. 125.
35. Kalchbrenner, N.; Espeholt, L.; Simonyan, K.; Oord, A.V.D.; Graves, A.; Kavukcuoglu, K. Neural Machine Translation in Linear Time. *arXiv* **2017**, arXiv:1511.07122v3.
36. CS231n. Convolutional Neural Networks for Visual Recognition. Available online: http://cs231n.github.io/convolutional-networks/#overview (accessed on 26 May 2020).
37. Nair, V.; Hinton, G.E. Rectified linear units improve Restricted Boltzmann machines. In Proceedings of the International Conference on Machine Learning, Haifa, Israel, 21–24 June 2010; pp. 807–814.
38. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *JMLR* **2014**, *15*, 1929–1958.
39. Kiefer, J.; Wolfowitz, J. Wolfowitz Stochastic Estimation of the Maximum of a Regression Function. *Ann. Math. Statist.* **1952**, *23*, 462–466. [CrossRef]