Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 171 (2020) 418–426

Procedia Computer Science

www.elsevier.com/locate/procedia

Third International Conference on Computing and Network Communications (CoCoNet'19)

Low-Complexity High-Performance Deep Learning Model for Real-Time Low-Cost Embedded Fire Detection Systems

Arpit Jadon[a,*], Akshay Varshney[a], Mohammad Samar Ansari[b]

[a] *Z.H.College of Engineering and Technology, Aligarh Muslim University, Aligarh 202002, India*
[b] *Software Research Institute, Athlone Institute of Technology, Athlone N37 HD68, Ireland*

Abstract

Correct and timely detection of fires has been an active area of research. Both shallow learning (with manual feature engineering), as well as deep learning (with its promise of automatically extracting meaningful representations from the data) approaches, have been used to solve fire detection problems. Most deep learning systems outperform the hand-crafted algorithms for fire detection, particularly due to the enormous potential offered by Convolutional Neural Network and its variants. The design problem is further compounded when the model is intended to be deployed on a low-computationally-intensive portable and mobile hardware. This requirement calls for a model which has a suitably small size on disk (translating to a lesser number of parameters to be estimated). Although some MobileNet based solutions are available which are superior to their counterparts (both in terms of increased accuracy as well as reduced complexity), there is still scope for improvement. The present work endeavors to demonstrate this by proposing a modified MobileNetV2 architecture and a better transparent data handling strategy that is capable of outperforming the existing solutions while being computationally viable for deployment on less able hardware.

Keywords: Convolutional Neural Networks; Deep Learning (DL); Neural Networks; Computer Vision; Image Processing; Fire Detection

1. Introduction

Almost all buildings such as large and small-scale industries, households, public places like sports stadiums, shopping malls, and movie theaters, are prone to accidents due to fire. As per the National Fire Protection Association, a fire department in USA has to respond to a fire call every 24 seconds [27]. According to another source [20], the rate of fire accidents in India has consistently increased over the past and has had a disruptive impact on the businesses running in the country. According to the FICCI-Pinkerton India risk survey conducted in 2017 [9], fire accidents have moved up by three ranks from the previous year rankings, to hold the fifth position in terms of the risk to businesses.

* Corresponding author. Tel.: +91-8266062042.
E-mail address: arpitjadon@zhcet.ac.in

1877-0509
10.1016/j.procs.2020.04.044

In addition, there have been a cumulative of 18, 450 reported major fire accident cases in India in 2015 with a massive number of 17, 700 people facing death and 1, 193 people being injured. All these accidents point towards the need for a reliable technology capable of efficient fire detection at an early stage so that timely fire suppression measures can be employed to prevent major damages.

Present fire detection systems are not entirely reliable because the mainstream fire detectors like thermal, smoke, and flame detectors rely on the presence of substantial heat, smoke, and radiations from flame in order to get triggered [1]. Also, their operation often leads to delays in fire detection thereby essentially rendering them as consequentially ineffective fire suppression system. In addition, with such fire detectors, there are inherent problems of false triggering. For instance, smoke detectors may be triggered even when the smoke has a source other than fire. Moreover, these methods of fire detection also lack in providing visual feedback of the fire spot to the central system.

These concerns regarding the present fire detecting systems call for more robust and reliable fire detection techniques. One such technique that is capable of enhancing the reliability and efficiency of fire detection, along with solving the problems associated with conventional fire detection systems, is visual-based fire detection technique. Such visual-based techniques are also used in diverse range of problems wherever detection or classification by humans or sensors becomes difficult or inefficient, for instance, in automatic fish recognition [21], segmentation of brain tumor in MRI images [28], and resistor value identification from resistor images [17] etc. Over the past decade, similar visual based fire detection techniques have been proposed by various researchers. Initially the hand-engineered approaches based on image processing and computer vision (CV) were proposed and later years saw increasing performance with better techniques finally leading to the deep learning based approaches benefited by the rapid advancements in the field of machine learning (ML) and artificial intelligence (AI).

1.1. Motivation

In some recent works, deep learning (DL) based fire detection techniques [23], [25], [36], [40] are reported to have better performance as compared to hand-engineered approaches like [32], [30], [3], [39]. The primary reason behind this significant improvement in performance is that the handcrafted approaches involve manual extraction of relevant features from images, while typical DL approaches are capable of extracting features from the images contained in the dataset, there by making them an efficient method of image classification. However, the requirement of hardware with high computational capacity render such solutions unsuitable for deployment on embedded devices. The reliability and accuracy of a field-deployable fire detection technique is one aspect, while the potential of the technique to run on economically feasible hardware is another crucial aspect. The latter aspect is important because a fire detection technique that cannot be deployed on low-cost hardware makes it harder to be utilized in the form of an end product.

1.2. Contribution

This paper focuses on a DL solution for fire detection which has both the following characteristics: (i) It exhibits improved fire detection performance over existing solutions (ii) It is also capable of being deployed on commercially available low-cost, moderate-performance hardware (e.g. Raspberry Pi) (iii) It is suitable for real-time applications.

Our aim is to achieve a better performance than previous approaches while keeping the frame rate obtained on the hardware device to a value which is reasonable enough for real-time fire detection. We have kept the performance of the model at the first priority while keeping frame rate as a satisficing factor. We also noticed the absence of a diverse fire dataset. Toward that end, we also introduce our own dataset compiled from multiple sources including some challenging fire and non-fire videos which we ourselves shot and compiled. In order to help other researchers to capitalize on our work, we also plan to open-source our dataset in the near future. Our method is inspired by the MobileNet V2 variant of the Convolutional Neural Network (CNN) [35], which is a suitable CNN baseline for embedded and mobile vision applications.

This paper is organized as follows, in section 2 we have discussed about the past research that has been conducted on computer vision and deep learning based fire detection, section 3 introduces our proposed approach, which is in turn followed by the brief description of our dataset and the improvements we have made, compared to previous approaches in terms of diversifying the dataset, subsequent sections include the description of the results with a brief discussion, and in the final section, we conclude.

2. Related Work

When the problem of visual fire detection was initially addressed, mostly handcrafted image processing algorithms were used [4], [2], [31], [29], [33], [22], [18], [10], [13]. Chen et al. [4] used the chromatic and dynamic values of the fire pixels to create a set of rules, albeit with low accuracy results. Celik et al. [2] used the `YCbCr` approach and determined rules to detect fire. Although this technique was good for flame detection, the false detection rate was high. Rafiee et al. [31] used the static and dynamic properties of the fire and smoke, but with limited accuracy. Qiu et al. [29] proposed an auto-adaptive edge detection algorithm for fire. Rudz et al. [33] employed the segmentation approach, which comprised of two steps, $k-$means clustering followed by filtering. Mueller et al. [22] used the optical flow technique. Kong et al. [18] used logistic regression and temporal smoothing. Foggia et al. [10] used several different classifiers and combined the result to accurately determine the fire in real-time. Habiboglu et al. [13] used the covariance method for a SVM based classification process. However, this method does not work well if the fire is small and/or distant.

Although the above approaches are capable of running on low-cost embedded hardware for real-time functioning, the task of manually extracting the features is very tedious and error-prone. In addition, with these methods it becomes extremely difficult to perform well on various special cases like fires of varying colors, or a moving fire like object complementing the fire. Therefore, considering that the Convolutional Neural Networks (CNNs) can provide the advantage of automatic feature extraction from raw images, CNN based deep learning approaches for fire detection present a better performance and are simpler in comparison to manual feature extraction based visual techniques.

The work that inspired the research community to use CNNs in CV was 'AlexNet' [19], after winning the 2012 ImageNet Large Scale Visual Recognition Competition (ILSVRC) [34], [7]. Zhang et al. [40] proposed a method based on detecting fire patches using fine-tuning with a modified AlexNet. However, since the model had a significantly large size (~238MB), it was not suitable for real-time fire detection. Frizzi et al. [12] also worked on CNN based fire detection but did not use any benchmark fire dataset. Sharma et al. [36] proposed a DL-based fire detection method utilizing Resnet50 (~98MB [6]) [14] and VGG16 (~528MB [6]) [37] as baseline architectures. Although the dataset (651 images) used for fine-tuning these networks was very versatile as it involved images that are very difficult to classify, and the accuracy attained was significant, the large size of both VGG16 and Resnet50 rendered them unsuitable for mobile and embedded vision applications. A recent work that provided a complete comparison (on a comprehensive fire dataset worthy of being called a benchmark dataset) was done by Muhammad et al. [23]. In their work on CNN-based fire detection, they utilized AlexNet as the baseline architecture and fine-tuned it on the vast video dataset of Foggia et al. [10]. They also showed the performance of their trained model on another challenging dataset of Chino et al. [5]. Though there was an improvement in the performance on Foggia's and Chino's dataset as compared to previous handcrafted approaches, the large size of AlexNet was still a problem. Muhammad et al. in another paper [25] tried to address the problem of high computational power required with AlexNet by taking 'GoogleNet' [38] as the pre-trained model for fine-tuning. Although the model was relatively small as compared to AlexNet [23], the size was still large enough to prevent the model from running on low-cost embedded hardware at a sufficient frame rate to detect fire in real-time.

To further address the problem of bulky trained CNN-based image classifiers, Muhammad et al. [24] used SqueezeNet [16] as the baseline architecture and fine-tuned the modified version of the SqueezeNet using the same datasets as they used in their previous two papers [23],[25]. Though the results show significant accuracy and capability of the model to run at 4 frames per second for real-time fire detection on the Raspberry Pi 3B [11], the very small number of parameters in the model restricts its performance. It is therefore evident that the performance can still be improved and approximately the same frame rate can be achieved on the same hardware with a CNN architecture that has a size and number of parameters intermediate between GoogleNet and SqueezeNet.

The most recent work is [26] where they utilized MobileNet$V2$ [35] as the baseline architecture and obtained better performance than all previous approaches. However, unlike their previous works, the training dataset they used for fine-tuning MobileNet$V2$ also contained Chino's [5] dataset, which only consists of 226 very challenging images. Thus, the performance presented in the paper on Chino's dataset is not a true representation of the their model's actual performance on this dataset. To overcome this shortcoming, we do *not* use the Chino's dataset as a part of our training dataset. Instead we emphasized on collecting more realistic fire scenario images. However, testing on Chino's dataset is important because of the huge diversity of the images in this dataset and also, following the past trend, for smooth comparison, it is mandatory that Chino's dataset must be refrained from use in training. Moreover, the dataset Muhammad et al. used for training consisted of 1844 fire and 6188 non-fire images. Therefore, the dataset was not balanced and better results could be obtained with a balanced dataset.

There are a few other limitations of the methods in [23], [25], [24]. In these works, Foggia's dataset constituted a major portion of the training samples. This dataset contains videos that comprise of around 62,690 frames, and is challenging if a few frames were acquired from each of the 31 videos. However, in these works, all the frames from each video were used for training and testing purposes. Thus, the accuracy obtained on Foggia′s dataset was not a true representation of how well the model would perform in the real-world situations as the model was trained on the images from the same dataset, and the dataset includes a large number of frames that look exactly similar. In addition, there was no mentioned use of any technique in the papers to enhance the versatility of the training dataset, thereby making it difficult to ascertain that the model will exhibit the same performance on the real-world images as was shown on the Foggia's dataset.

We utilize a model that has architecture similar to MobileNet$V2$, trained on a balanced training dataset capable of detecting fire in real-world like scenarios. We also introduce our own dataset, which consists of 25,656 fire images shot in challenging real-world situations and 6,907 non-fire images taken in the presence of fire like objects in the surroundings. We have sampled some images from this dataset for training our model. The results obtained in terms of various metrics are better than the previous methods, and also the frame rate obtained for real-time fire detection on low-cost embedded hardware is sufficient for accurate real-time fire detection.

3. Proposed Approach

3.1. CV for Mobile and Embedded Applications

When the system lacks computational power, large-sized deep CNNs cannot be used, because they have to accomplish many mathematical operations to reach to their final output, which burdens the hardware to its extreme computational capability and simultaneously makes these deep CNNs unsuitable for mobile and embedded vision based applications. Some deep networks, which have a very large number of parameters, are VGG-16, AlexNet, ResNet, etc. Thus, in order to accomplish image classification on mobile and embedded devices, one has to reduce the total number of parameters so that it could run efficiently on devices with lesser computational power. Some networks that were designed to do these tasks were the MobileNets [15], [35], GoogleNet [38], and SqueezeNet [16]. These networks have small size, and are being frequently used in varying levels of embedded and mobile vision applications. Thus, these CNNs are suitable for use in devices which do not have huge computational capabilities without compromising significantly on the accuracy of the results. There are two versions of MobileNet viz. MobileNet$V1$ (4.2 million parameters, typical disk size of 21 MB) and MobileNet$V2$ (3.4 million parameters, typical disk size of 13 MB). Also, as pointed out in [26], MobileNet$V2$ proves to be faster than the MobileNet$V1$ for computer vision tasks while showing an improved performance.

MobileNet$V1$ uses an efficient architecture that utilizes depth-wise separable convolutions in order to build the lightweight deep neural networks suitable for mobile and embedded vision applications. This CNN also uses two hyper-parameters, width multiplier and resolution multiplier, which contribute to a reduction in the size of MobileNets with improved efficiency. MobileNet$V2$ is an advancement over MobileNet$V1$ and its network design is actually based on MobileNet$V1$. It is specifically designed for computer vision applications in mobile and resource-constrained environments. It retains the same accuracy as that of the other state of the art mobile models while reducing the number of operations and memory required to a significant extent. The main contribution in this model was the introduction of a novel layer module, called the inverted residual with linear bottleneck module.

3.2. Framework and Approach

Our model for mobile and embedded application oriented fire detection is inspired by the MobileNet$V2$ architecture, and is presented in Fig. 1. In a typical DL network, the initial few layers are used for the detection of simple low-level features while the deeper layers are capable of detecting more specific high-level complex features. Therefore, we freezed the parameters of the initial layers in MobileNet$V2$ and removed the last layer from the model.

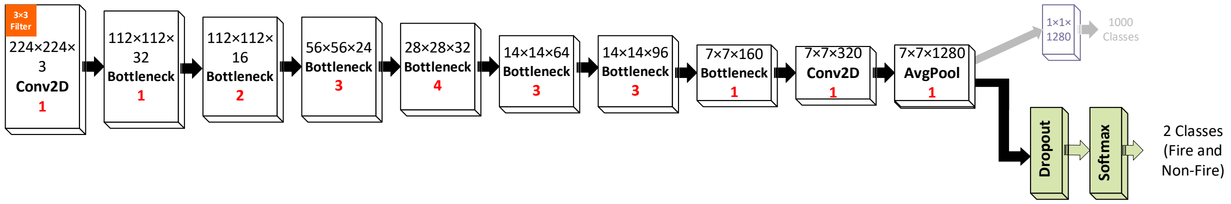

Fig. 1. Architecture of the deep learning model used in this work. The greyed-out portion shows the layer(s) removed from the MobileNetV2 model, and the green blocks show the added layer(s). Numbers in red denote the number of times the block is repeated. Detailed information about the internals of the different layers (except the ones added in this proposal, shown in green), expansion factors, and strides, can be obtained from [35].

Subsequently, we added a dropout layer prior to the final classification layer in order to prevent overfitting. This dropout layer was followed by a softmax classification layer for our two classes. In this way, we fine-tuned the model for our two (fire and non-fire) classes using the training dataset, which is discussed in the next section. A major portion of our work involved developing a better data handling strategy, which is also mentioned in the following section. The results obtained on different datasets are presented in Section 5.

4. Datasets: Features and Suggested Improvements

We have followed a more rigorous dataset distribution strategy than what was followed in previous works [23], [25], [24], [10], for better performance in diverse scenarios and for a more realistic comparison. The Foggia's dataset [10] consists of 31 videos out of which, 14 videos contain frames with fire (except a couple of videos, which only contain fire in a subset of frames in the video), while the rest of the 16 videos do not contain fire, and are shot in normal environment. This dataset is challenging with respect to an inter-video context as it contains videos shot in smoky, foggy surroundings, with fire-colored objects in the video, and variably distant (far and close) fires from the camera, etc. But when it comes to comparing the diversity within the frames obtained from a single video, the dataset seems to lack diversity. Thus, although the number of images (62690) in the dataset appears to be large, the similarity between a lot of images within the dataset makes it a not so versatile dataset.

In the previous works [23], [25], [24] on CNN based fire detection, this dataset has been used for both training and testing distribution, thus, resulting in the spiking of testing accuracy on the Foggia's dataset, which obviously doesn't reflect the efficiency of the methods on real-world scenarios. In [26], although the training dataset is claimed to be taken such that it covers uncertain environments as well, the dataset is imbalanced (1844 fire images and 6188 non-fire images). Therefore, good performance on Foggia's dataset with this model does not truly represent the possible outcome in real-world scenarios. In addition, since in this work, Chino's dataset is also used as part of the *training* set, the test performance on this dataset is not a true indicator of the model's performance in realistic fire scenarios. This is attributed to the very small size of Chino's dataset (226 images only). Some images from the dataset are shown in Figure 2.

In addition, Foggia's dataset contains 6,311 frames with fire in it (excluding the frames from those two videos, which don't have fire frames in partial sections of the video), and the number of frames without fire is 51,489. In order to make our dataset more versatile, we only took a certain number of frames from each non-fire video in order to break the homogeneity among the non-fire frames that is bound to creep in due to the presence of a lot of similar images in this dataset. Thus, we only took 36,271 non-fire frames out of these 51,489 frames. Then we used data augmentation in order to increase the number of fire frames as well as to make the fire class image dataset more versatile. As a part of augmentation, we used various techniques like rotation, flipping, zooming, and shifting, etc. resulting in the increase of number of fire images from 6,311 to 36,884. We also used the dataset used by Sharma et al. [36] for training purpose, as this dataset although small, is extremely diverse with 541 non-fire images having a fire like background. The dataset also contains 110 fire images which are taken in a challenging environment. Few images from this dataset are also shown in the Figure 2. We also augmented the fire class images of this dataset to resonate a more balancing situation between the two classes.

Further, driven by the lack of available diverse fire datasets that can simulate real-world situations more closely, we tried to accumulate our own fire dataset, which contains mostly the fire class images, as these are hard to capture

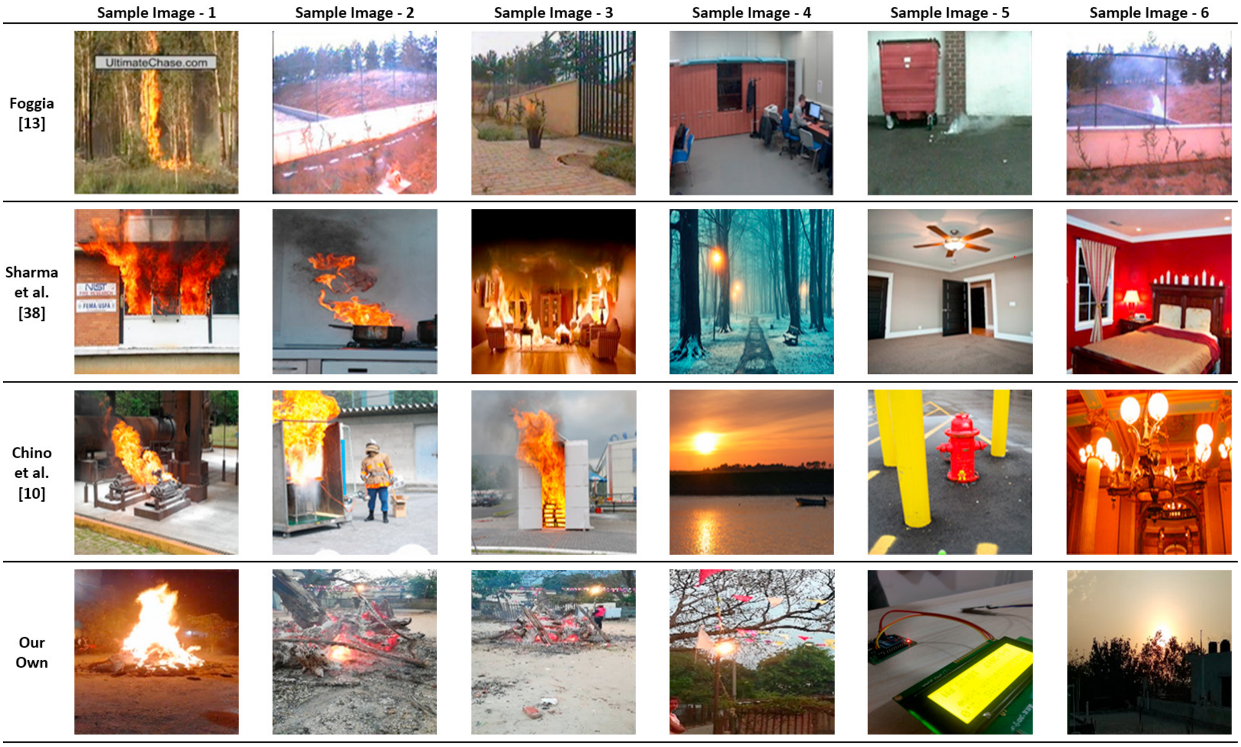

Fig. 2. Sample images from the various datasets.

in comparison to non-fire images. Our dataset also includes some challenging non-fire class images. The dataset[1] contains a total of 77 videos with 25,656 frames belonging to fire class and 16 videos with a total of 6,747 frames belonging to non-fire class along with 160 more individually clicked non-fire images. We have shown some images from this dataset in Figure 2. For our training dataset, we sampled some images from these videos, again to maintain diversity in the dataset. Also, it is worthwhile to note that unlike [26], we have not used Chino's dataset for training purpose. Out of all these images we had, we used 20% images for training and remaining 80% images for testing, a breakdown followed in most recent works [23], [26], [25], [24], [10]. Our final training dataset included 8,481 images each in fire and non-fire classes.

5. Results

We performed extensive evaluations of the proposed architecture on different datasets. The results of performance evaluation on dataset 1 (Foggia's dataset) in terms of the various metrics such as accuracy, false positives, false negatives, precision, recall, and F-measure are shown in Table 1. Table 2 presents the evaluation results of the same model on dataset 2 (Chino's dataset) on the same metrics. Table 3 and Table 4 show the comparison of our approach with previous handcrafted approaches in terms of the different metrics chosen. It is to be noted that the choice of different metrics in Table 3 and Table 4 is done to offer a valid comparison among different approaches. It needs to be mentioned that Chino's dataset (dataset 2) is extremely challenging and versatile, making fire detection on this dataset very difficult. However, our model has been able to perform well on this dataset in comparison to past methods, as shown by Table 2 and Table 4.

Further, as can be seen from Table 1, our model performs extremely well on dataset 1 in terms of accuracy, false positives, precision, and F-measure. The recall value is also quite high and the false negatives percentage is also quite

[1] Dataset available at: https://bit.ly/2QsrWhM

Table 1. Comparison with previous deep learning approaches on dataset 1 (Foggia's).

Methods	Accuracy (%)	False Positives(%)	False Negatives (%)	Precision	Recall	F-Measure
Our Approach	**99.17**	**0**	0.80	**1**	0.98	**0.99**
Muhammad et al. [26]	95.86	**0**	0.14	**1**	**0.99**	**0.99**
Muhammad et al. [24]	94.61	0.06	1.24	0.99	0.98	0.98
Muhammad et al. [25]	93.66	**0**	1.09	**1**	0.97	0.98
Muhammad et al. [23]	94.27	6.78	**0.08**	0.93	**0.99**	0.96

Table 2. Comparison with previous deep learning approaches on dataset 2 (Chino's).

Methods	Accuracy (%)	False Positives(%)	False Negatives (%)	Precision	Recall	F-Measure
Our Approach	**91.15**	**4.42**	4.42	**0.92**	0.92	**0.92**
Muhammad et al.[26]	92.04	9.34	6.72	0.90	0.93	0.92
Muhammad et al. [24]	89.82	18.69	2.52	0.83	0.97	0.90
Muhammad et al. [25]	84.96	24.29	6.72	0.79	0.93	0.85
Muhammad et al. [23]	88.05	23.36	**1.68**	0.80	**0.98**	0.88

Table 3. Comparison with previous hand-crafted approaches on dataset 1 (Foggia's).

Methods	False Positives(%)	False Negatives(%)	Accuracy(%)
Our Approach	**0**	0.80	**99.17**
Foggia et al. [10]	11.67	**0**	93.55
Lascio et al. [8]	13.33	**0**	92.86
Habibuglu et al. [13]	5.88	14.29	90.32
Rafiee et al. (RGB) [31]	41.18	7.14	74.20
Rafiee et al.(YUV) [31]	17.65	7.14	87.10
Celik et al.[2]	29.41	**0**	83.87
Chen et al. [4]	11.76	14.29	87.10

Table 4. Comparison with previous hand-crafted approaches on dataset 2 (Chino's).

Methods	Precision	Recall	F-Measure
Our Approach	**0.92**	**0.92**	**0.92**
Chino et al. [5]	0.51	0.65	0.57
Rudz et al. [33]	0.63	0.45	0.52
Rossi et al. [18]	0.39	0.22	0.28
Celik et al. [2]	0.55	0.54	0.54
Chen et al. [4]	0.75	0.15	0.25

low. Results on dataset 2 are also very encouraging as our method outputs the best performance in terms of accuracy, precision, false positives, and F-measure metrics. The model also gives a good value of recall while keeping the false negatives to sufficiently low value. A comparison with the previously introduced hand-crafted approaches also show the superior performance of our model in terms of all six metrics as evident from Table 3 and Table 4.

Regarding Table 2, it needs to be noted that the Chino's dataset is quite small (only 226 images) and [26] have used at-least a portion of this dataset for training purpose. Therefore, a direct comparison between our results and those in [26] on Chino's dataset is not valid, as we have *not* used any image from Chino's dataset for training our network, as was also done in [23], [25], [24], which is a valid strategy as being a very versatile dataset, the test performance on this dataset can closely reflect the performance in difficult real-world situations.

Table 5. Frame rate comparison with previous DL and hand-crafted approaches.

Methods	Frame Rate (fps)	Hardware Used
Our Approach	5	Raspberry Pi 3B
Muhammad et al. [26]	5	Raspberry Pi 3B
Muhammad et al. [24]	4	Raspberry Pi 3B
Foggia et al. [10]	3	Raspberry Pi B

Finally, in Table 5 we have shown the results of our methods in comparison to previous DL and hand-crafted approaches based on the frame rate obtained on the Raspberry Pi 3B hardware. A frame rate of 5 frames per second was obtained with our trained network while detecting the fire on Raspberry Pi 3B, which points towards the feasibility of our network for embedded and mobile vision applications. This frame rate is good enough to detect fire while running on any embedded or mobile device in real-time.

6. Conclusion

Various efficient visual based fire detection systems have been developed so far. CNN based fire detection methods have proven to be simple while simultaneously showing superior performance in detecting fire as compared to the hand-engineered approaches. However, most CNN based approaches have the big disadvantage of being computationally hungry even when only the trained model is deployed to the machine for real-time fire detection. This problem is further worsened when the hardware in consideration is a low-cost embedded or mobile device with very low on-board computational power. There have been some research efforts toward that direction, and the present work is one such effort. We proposed an embedded and mobile devices friendly efficient fire detection method. The results obtained on various standard fire datasets with our method were superior to all the existing methods in terms of most evaluation metrics like accuracy, precision, recall, and f-measure. Our method is inspired by the CNN architecture named MobileNet$V2$. In a future work, we would work to increase the frame rate obtained on low-cost embedded hardware, and also expand our dataset in order to replicate more real-world like situations.

References

[1] Artim, N., 1999. An Introduction to Fire Detection, Alarm, and Automatic Fire Sprinklers. https://www.nedcc.org/free-resources/preservation-leaflets/3.-emergency-management/3.2-an-introduction-to-fire-detection,-alarm,-and-automatic-fire-sprinklers. [Last accessed on: 01 August 2019].

[2] Celik, T., Demirel, H., 2009. Fire detection in video sequences using a generic color model. Fire Safety Journal 44, 147–158.

[3] Çelik, T., Özkaramanlı, H., Demirel, H., 2007. Fire and smoke detection without sensors: Image processing based approach, in: 2007 15th European Signal Processing Conference, IEEE. pp. 1794–1798.

[4] Chen, T.H., Wu, P.H., Chiou, Y.C., 2004. An early fire-detection method based on image processing, in: 2004 International Conference on Image Processing, 2004. ICIP'04., IEEE. pp. 1707–1710.

[5] Chino, D.Y., Avalhais, L.P., Rodrigues, J.F., Traina, A.J., 2015. Bowfire: detection of fire in still images by integrating pixel color and texture analysis, in: 2015 28th SIBGRAPI Conference on Graphics, Patterns and Images, IEEE. pp. 95–102.

[6] Chollet, F., 2018. Deep learning models. https://github.com/fchollet/deep-learning-models/releases. [Last accessed on: 01 August 2019].

[7] Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L., 2009. Imagenet: A large-scale hierarchical image database .

[8] Di Lascio, R., Greco, A., Saggese, A., Vento, M., 2014. Improving fire detection reliability by a combination of videoanalytics, in: International Conference Image Analysis and Recognition, Springer. pp. 477–484.

[9] FICCI, 2017. FICCI Pinkerton India Risk Survey 2017. http://ficci.in/pressrelease-page.asp?nid=2806. [Last accessed on: 01 August 2019].

[10] Foggia, P., Saggese, A., Vento, M., 2015. Real-time fire detection for video-surveillance applications using a combination of experts based on color, shape, and motion. IEEE TRANSACTIONS on circuits and systems for video technology 25, 1545–1556.

[11] Foundation, R.P., 2019. Raspberry pi 3 model b. URL: https://www.raspberrypi.org/products/raspberry-pi-3-model-b/. [Last accessed on: 01 August 2019].

[12] Frizzi, S., Kaabi, R., Bouchouicha, M., Ginoux, J.M., Moreau, E., Fnaiech, F., 2016. Convolutional neural network for video fire and smoke detection, in: IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society, IEEE. pp. 877–882.

[13] Habiboğlu, Y.H., Günay, O., Çetin, A.E., 2012. Covariance matrix-based fire and flame detection method in video. Machine Vision and Applications 23, 1103–1113.
[14] He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778.
[15] Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H., 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 .
[16] Iandola, F.N., Han, S., Moskewicz, M.W., Ashraf, K., Dally, W.J., Keutzer, K., 2016. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. arXiv preprint arXiv:1602.07360 .
[17] Jadon, A., Varshney, A., Varshney, N.G., Ansari, M.S., 2018. Simple and efficient non-contact technique for resistor value estimation, in: 2018 International Conference on Communication and Signal Processing (ICCSP), IEEE. pp. 0938–0941.
[18] Kong, S.G., Jin, D., Li, S., Kim, H., 2016. Fast fire flame detection in surveillance video using logistic regression and temporal smoothing. Fire Safety Journal 79, 37–43.
[19] Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks, in: Advances in neural information processing systems, pp. 1097–1105.
[20] Lahiry, S., 2018. Why fire accidents are unabated. https://www.downtoearth.org.in/blog/natural-disasters/fire-safety-59773. [Last accessed on: 01 August 2019].
[21] Meng, L., Hirayama, T., Oyanagi, S., 2018. Underwater-drone with panoramic camera for automatic fish recognition based on deep learning. IEEE Access 6, 17880–17886.
[22] Mueller, M., Karasev, P., Kolesov, I., Tannenbaum, A., 2013. Optical flow estimation for flame detection in videos. IEEE Transactions on image processing 22, 2786–2797.
[23] Muhammad, K., Ahmad, J., Baik, S.W., 2018a. Early fire detection using convolutional neural networks during surveillance for effective disaster management. Neurocomputing 288, 30–42.
[24] Muhammad, K., Ahmad, J., Lv, Z., Bellavista, P., Yang, P., Baik, S.W., 2018b. Efficient deep cnn-based fire detection and localization in video surveillance applications. IEEE Transactions on Systems, Man, and Cybernetics: Systems , 1–16.
[25] Muhammad, K., Ahmad, J., Mehmood, I., Rho, S., Baik, S.W., 2018c. Convolutional neural networks based fire detection in surveillance videos. IEEE Access 6, 18174–18183.
[26] Muhammad, K., Khan, S., Elhoseny, M., Ahmed, S.H., Baik, S.W., 2019. Efficient fire detection for uncertain surveillance environment. IEEE Transactions on Industrial Informatics .
[27] NFPA, 2019. U.S. fire problem. "https://www.nfpa.org/News-and-Research/Data-research-and-tools/US-Fire-Problem. [Last accessed on: 01 August 2019].
[28] Pereira, S., Pinto, A., Alves, V., Silva, C.A., 2016. Brain tumor segmentation using convolutional neural networks in mri images. IEEE transactions on medical imaging 35, 1240–1251.
[29] Qiu, T., Yan, Y., Lu, G., 2012. An autoadaptive edge-detection algorithm for flame and fire image processing. IEEE Transactions on instrumentation and measurement 61, 1486–1493.
[30] Qureshi, W.S., Ekpanyapong, M., Dailey, M.N., Rinsurongkawong, S., Malenichev, A., Krasotkina, O., 2016. Quickblaze: early fire detection using a combined video processing approach. Fire technology 52, 1293–1317.
[31] Rafiee, A., Dianat, R., Jamshidi, M., Tavakoli, R., Abbaspour, S., 2011. Fire and smoke detection using wavelet analysis and disorder characteristics, in: 2011 3rd International Conference on Computer Research and Development, IEEE. pp. 262–265.
[32] Rossi, L., Akhloufi, M., Tison, Y., 2011. On the use of stereovision to develop a novel instrumentation system to extract geometric fire fronts characteristics. Fire Safety Journal 46, 9–20.
[33] Rudz, S., Chetehouna, K., Hafiane, A., Laurent, H., Séro-Guillaume, O., 2013. Investigation of a novel image segmentation method dedicated to forest fire applications. Measurement Science and Technology 24, 075403.
[34] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al., 2015. Imagenet large scale visual recognition challenge. International journal of computer vision 115, 211–252.
[35] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C., 2018. Mobilenetv2: Inverted residuals and linear bottlenecks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4510–4520.
[36] Sharma, J., Granmo, O.C., Goodwin, M., Fidje, J.T., 2017. Deep convolutional neural networks for fire detection in images, in: International Conference on Engineering Applications of Neural Networks, Springer. pp. 183–193.
[37] Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556.
[38] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. Going deeper with convolutions, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1–9.
[39] Toulouse, T., Rossi, L., Celik, T., Akhloufi, M., 2016. Automatic fire pixel detection using image processing: a comparative analysis of rule-based and machine learning-based methods. Signal, Image and Video Processing 10, 647–654.
[40] Zhang, Q., Xu, J., Xu, L., Guo, H., 2016. Deep convolutional neural networks for forest fire detection, in: 2016 International Forum on Management, Education and Information Technology Application, Atlantis Press.