Grenze International Journal of Engineering and Technology, June Issue

Utilizing Convolutional Neural Networks (CNNs) for the Detection of Smoke and Fire through CCTV Camera Surveillance

Pankaj Pratap Singh[1], Nimish Grover[2], Prashansa Sharma[3] and Pooja Singhal[4]
[1-4]IMS Engineering College, Ghaziabad, India
Email: pankajpratapsinghcs@gmail.com, grovernimish2003@gmail.com, sharmaprashansa1@gmail.com, poojasinghal8171@gmail.com

***Abstract*— Fires frequently cause large losses in terms of both property and human life. Thus, there is an urgent need for fire detection systems that are easily accessible, reasonably priced, accurate, and quick. The results of earlier attempts to create workable solutions have been inconsistent. A significant obstacle is striking a balance between model size and system performance, with the latter allowing portability to devices such as Raspberry Pi. This study proposes FireDetector, a neural network designed with performance and scalability in mind. FireDetector specifically seeks to (i) surpass current methods in accuracy and (ii) stay small enough for hardware deployment. Positive findings are obtained from evaluations using newly introduced bespoke fire datasets and benchmark datasets.**

I. INTRODUCTION

The necessity for trustworthy fire alarms in infrastructure and buildings is highlighted by the rise in fire incidents in recent years. Reducing the frequency of fire incidents requires accurate and timely detection to reduce the amount of damage that results. Current systems, however, have flaws, most notably high false positive rates. These days, mainstream methods rely on tangible sensors like flame, smoke, and thermal detectors. However, sensor-dependent systems don't always have reliable fire detection; for example, smoke detectors can set off false alarms if they can't tell the difference between smoke and fire. Conversely, heat-and flame-based technologies can need a large amount of escalation before a reliable detection can occur, which could lead to detrimental delays. One possible substitute to improve system dependability is to use visual fire detection techniques.

Vision-based research topics like Image Processing and Computer Vision have reaped a good deal of benefits from the advancements in various artificial intelligence fields. In certain computer vision tasks, such as image categorization, a variety of deep learning (DL) [1] models have been able to easily outperform human level performance. These technology advancements have also been beneficial to the visual-based fire detection strategy. The use of visual-based fire detection systems is more cost-effective, accurate, resilient, and reliable than hardware-based alarm systems. Deep learning-based techniques have gradually supplanted manual visual fire detection methods, which perform worse in terms of accuracy and false triggers. These methods perform better in terms of fluctuating metrics. The capacity of deep learning-based techniques to automatically extract features from the raw photos is credited with this improved performance. In contrast, because the characteristics in the input

Grenze ID: 01.GIJET.10.2.508_2

photos have to be manually retrieved, the handmade techniques need to be handled with greater care. Thus, a more robust and dependable fire alarm system might be created by fusing these more trustworthy visual-based fire detection approaches with the traditional sensor-based procedures. Thus, the goal of this work is to introduce an upgraded smoke and fire detection unit that integrates several cutting-edge technical concepts such as Convolutional Neural Networks (CNN) and the Internet of Things (IoT) and is dependable, reducing the false triggering issue [2]. We created a proprietary neural network model trained on a composite dataset in order to achieve efficient real-time fire detection performance while maintaining sufficient frame rates utilizing the Raspberry Pi 3B (1.2GHz Broadcom BCM2837 64bit CPU, 1 GB RAM system [3]). The suggested model is tested using a variety of fire data sets, including a collection of benchmark data from earlier research as well as cumulative real-world occurrences [4]. The system also detects the difference between fire and smoke. False positive activations can be decreased by setting off separate smoke and fire alarms, respectively.

However, there are a lot of challenges with visual smoke detection. Color values of pixels are usually analyzed during video processing. Therefore, it can be difficult to distinguish smoke from other similarly grey elements in real time [5]. Moreover, smoke identification is hampered by dark conditions due to camera capture limitations. Therefore, our system uses a smoke sensor to detect smoke for improved overall performance. By doing this, conventional systems that use smoke detectors for fire detection can reduce the number of false alarms while doing away with the need for separate fire and smoke detectors. Smoke sensors offer effective and reasonably priced smoke detection capabilities. Numerous uses exist for this type of intelligent fire and smoke detection unit, including early emergency alerts, alerting fire departments to a fire so they can arrive at the scene quickly, activating automatic fire suppression systems, and more [6].

The main contributions of this work are:

- In contrast to current deep learning solutions that relyon larger CNNs, we present a shallow neural network-based system for fire identification that performs real-time classification at higher frame rates.
- We create a unique training set of composite fire images by merging various samples from various artifacts into a small yet sufficiently informative dataset that can be used for reliable network learning.
- In order to compare the approaches, we also put together a difficult video data collection of simulated fire incidents.
- Our integrated fire detection unit, which can replace conventional sensor-driven systems and reduce false and delayed alarms, has a working prototype available.

This paper is organized as follows. In Section II, previous research on handmade and deep learning methods for fire identification is reviewed. The details of the dataset are expounded upon in Section III, while our proposed methodology is explained in length in Section IV. In Section V, we present the experimental data; in Section VI, we analyze the effectiveness of our method. With final remarks and a discussion of possible future possibilities, Section VII comes to a close.

II. PAST RESEARCH WORK

Because there are so many fire incidents every day, it is crucial for current technology to have a suitable system for detecting fires in order to minimize the damage that can be done [7]. Creating methods that took advantage of flame color and motion cues was the main emphasis of early fire detection research. A technique for flame identification utilizing chromatic and dynamic fire and smoke patterns was presented by Thou-Ho et al. [8]. In addition to using two color spaces to identify fire from smoke, Celik et al. [9] used fuzzy logic techniques to provide robust non-fire object discrimination. Thou-Ho et al. employed the RGB space, while Turgay et al. [9] changed the strategy to produce more broadly applicable fire detection criteria by adopting YCbCr. Turgay et al.'s approach, however, had a limited detection range and high false positive rates. In addition to color, certain methods, such as those used by Rafiee et al. [10], took motion signatures into account by examining both static and dynamic fire/smoke properties. False negatives continued to be a problem as background objects often mimic fire pixel color properties. An automated edge detection approach for flame identification was proposed by Qiu et al. [11]. While using fire dynamic signatures, Rinsurongkawong et al. [12] found difficulty while dealing with backdrops that had pseudo-fire objects. In order to overcome this constraint, [13] used a dual optical flow estimator to separate elements that were fire and those that weren't. A multi-sensor system named Safe from Fire (SFF) was created by Mobin et al. [14] to enable separate fire and smoke detection. But the reliance on sensors raised system expenses. One major disadvantage of hand-engineered fire detection approaches is that they require human feature extraction from raw photos, even though they are computationally efficient for deployment on hardware such as Raspberry Pi. This makes the approach laborious and ineffective for big datasets. On the other hand, as compared to traditional image processing, deep learning techniques may automatically extract prominent visual information, increasing efficiency and reliability. However, the majority of deep learning systems require a significant amount of processing power, both during training and for post-deployment inference. The real-world feasibility of fire detection is greatly limited by its dependency on power- hungry gear, such as personal computers. The solution must be as large and reasonably priced as conventional detectors. There are now several deep learning approaches available for identifying fires. By fine-tuning AlexNet [16], a CNN that has already been trained, to identify fire patches, Zhang et al.[15] carried out forest fire detection. A CNN method comparing the basic VGG16 [18] and Resnet50 [19] architectures was also provided by Sharma et al. [17].

Nevertheless, the large model sizes and parameter counts of these networks hinder the implementation of fire systems in the real world on low-cost hardware, even with accuracy advances. The flaws in Foggia's dataset restrict the performance of the model. There is a significant amount of picture similarity despite the fact that there are 14 fire and 17 non-fire videos with many frames. This limits models to a small set of relevant photos. Furthermore, the training set employed in [20] is unbalanced, consisting of 6188 non-fire and 1844 fire images. Results from such an uneven dataset could be skewed.

More importantly, these networks are not well suited for embedded vision applications due to their enormous number of layers and large on-disk size. We find that a shallow network with high frame rate and good accuracy is required, which is indicated by the difficult training dataset selection and the excessively large models. This enables effective implementation on inexpensive embedded devices. Thus, taking inspiration from our earlier work, we blended samples from Foggia's and Sharma et al.'s datasets with photos from Google and Flickr to produce a more varied and difficult training set [17]. Our solution to the problems of over- dimension and over-layering is a neural network that we created from scratch, named "FireDetector." Furthermore, we made our dataset and model publicly available to help other researchers enhance fire detection. Additional information about our dataset and network are in the following sections.

III. Dataset Description

Current fire datasets are not diverse, according to analysis. Although there are 31 fire and non-fire movies in Foggia et al.'s dataset [4], it is too homogeneous for reliable model training. Diversity is limited by resemblance across multiple frames, despite being expansive. By using difficult video grabs and online photos of fires and non-fires with backgrounds resembling fires from Google and Flickr, we were able to construct a more diversified dataset. We augment the datasets of Foggia and Sharma [17] with samples from our training set. After adding to Sharma's data and choosing extra photos at random, we had a final count of 1,124 fire and 1,301 non-fire photos. Even with the reduced amount, it offers more variety.

We concentrated on building more realistic fire images in our test dataset because real deployment requires strong performance in these kinds of settings. We saved the realistic examples only for testing because our training set was already quite varied. The entire test set consists of 16 non-fire films (6,747 frames), 46 fire videos (19,094 frames), and 160 more difficult non-fire photos.

We created the final test set for evaluation by randomly sampling frames from various videos in order to increase diversity. As was covered in Section VI, the model performed admirably. Example training and test images are shown in Figures 1 and 2, respectively.

Fig. 1. Few images from our training dataset.

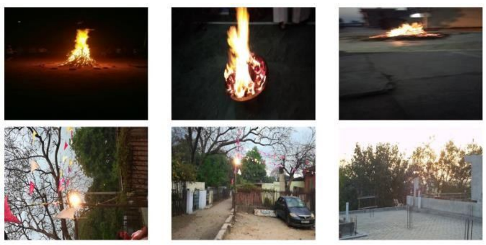

Fig. 2. Few images from our test dataset.

IV. Proposed Approach

Previous fire detection techniques, such as those found in references [17], only improved big CNNs like VGG16, ResNet, GoogLeNet, SqueezeNet, and MobileNetV2. Overly large training sizes and layers are a significant disadvantage of just fine-tuning such heavy models. This makes it more difficult for low-cost electronics, like

Raspberry Pi, to operate in real time. In order to be deployed as fire systems in real- world settings such as shopping centers, residences, hotels, etc., quick frame rates on low-cost devices are essential. The focus on effective embedded deployment is warranted because all of these strategies aim to address real-world fire detection units.
Therefore, unlike expensive, impractical devices, low-cost commercial hardware allows economically viable real-world deployment. We present FireDetector, a low-weight neural architecture with good real-time performance that may be used for embedded and mobile fire detection. The network runs on less powerful, more economical single board computers, such as the Raspberry Pi 3B, at a very satisfactory frame rate of more than 24 frames per second.
We also suggested the following flowchart (Fig. 3) to help with the overall model creation process. In order to assist with the general data flow in the model, we have also included our data flow diagram (Fig. 4) for our own CNN model.
A limited number of dense layers, including an output "softmax" layer, and convolution layers will make up the suggested neural network.
The neural network will have the fewest possible layers in order to operate on lightweight devices, and it will be simple to install on any kind of system, from high-end desktop configurations to general-purpose computers. Since surveillance cameras don't require a high-end PC, the model was created with minimal specifications in mind, ensuring optimal performance in all circumstances.

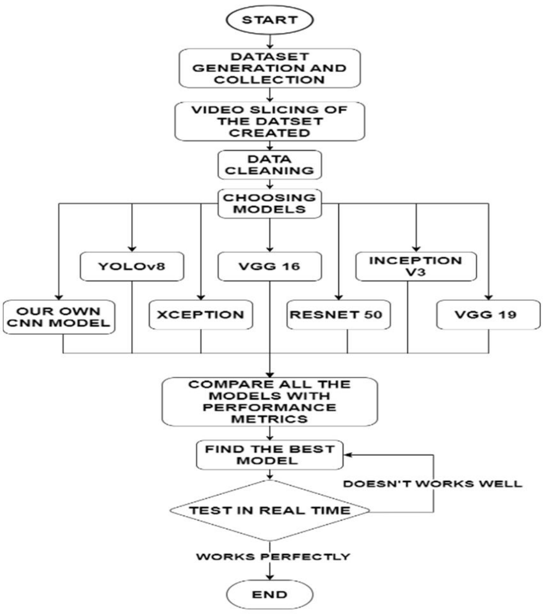

Fig. 3. Flow Chart for overall model creation process.

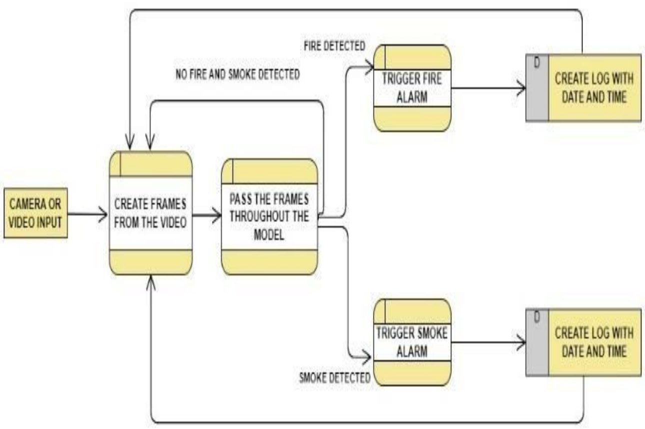

Fig. 4. Data Flow Diagram for our model.

A. Architecture

The suggested network's whole architectural diagram is displayed (Fig. 5). The following architecture adheres to all the specifications kept in mind to accurately detect smoke and fire in a low-spec PC at a pace of about 24 FPS. Fig. 5 displays the whole architecture layout of the suggested network, demonstrating that model has 14 layers in total. Each of the three convolution layers is connected to a dropout layer and pooling. Except for the last layer, which uses Softmax as its activation function, each of these layers uses Relu as its activation function. The total number of trainable parameters turns out to be 646,818 (size on disk ~7.45 MB).

V. Results

An embedded, deployable fire detection system is the goal. Thus, it is necessary to assess real-world imagery captured by typical security cameras. There isn't a diversified fire dataset meeting in existence right now. The data from Foggia et al.[4] lacks diversity but includes some genuine photos of fire emergencies. We still benchmarked against it for comparison's sake. Our test set consists of independently collected videos that depict difficult real-world fire and non- fire situations. By putting the model to the test in real-world scenarios, this helps to better demonstrate the integrated fire detection unit's practicality. The performance obtained on both these datasets is shown. Accuracy, precision, recall, and F-measure are the four metrics we used to give a thorough and trustworthy study.

Furthermore, considering the restricted diversity of Foggia's dataset, performance is better. The training curves are shown here to better exemplify the reliability of the model. The training and validation loss curves from the training procedure are displayed in Fig. 6. The matching training and validation accuracy curves for the same process are shown in Fig. 7.

The performance gained on all models, including our own, is compared in the table that follows (Table 1), which includes metrics like validation accuracy, validation loss, training accuracy, and training accuracy. We can categorically state that our model performs the best out of all of them based on these factors, and real-time testing also demonstrates how well our model performs because it is accurate and lightweight, having been trained on a proprietary dataset that includes real-world movies. For equitable results delivery, every model trained for this study used the same dataset.

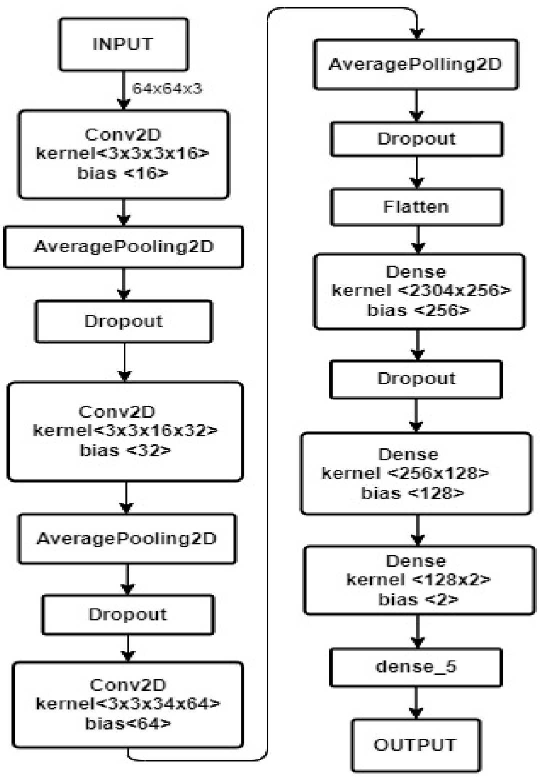

Fig. 5. Architecture of Proposed Network

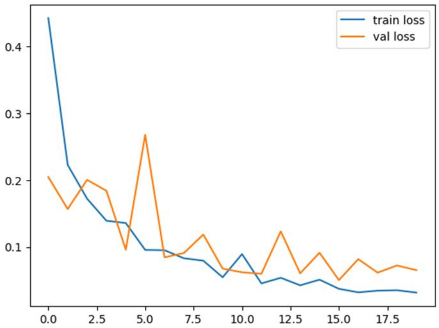

Fig. 6. Training and validation curves for model loss.

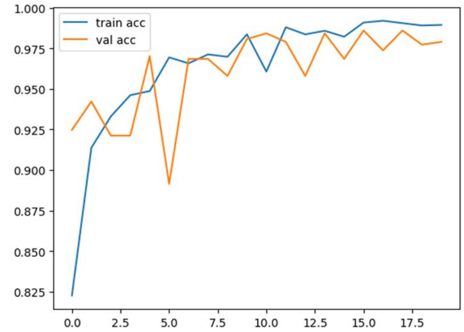

Fig. 7. Training and validation curves for model accuracy.

TABLE I. COMPARISON OF ALL MODELS ON OUR CUSTOM DATASET

S.No	Model	Training Accuracy	Validation Accuracy	Training Loss	Validation Loss
1.	VGG 16	98.45%	93.67%	2.67%	3.89%
2.	VGG 19	99.12%	94.58%	1.16%	2.34%
3.	RESNET 50	80.15%	85.64%	55.82%	34.50%
4.	XCEPTION	97.16%	95.72%	5.60%	41.55%
5.	INCEPTION v3	97.45%	91.85%	1.93%	3.76%
6.	YOLO v8	98.27%	95.22%	2.55%	4.66%
7.	FireDetector	99.24%	98.90%	0.31%	0.65%

VI. DISCUSSION ON SIMILAR WORK

In this section, FireDetector and other cutting-edge fire detection techniques are contrasted. The primary benefit of FireDetector on is its small size, which results from employing a shallow network with few parameters. Granted, higher performance is reported in some literature. But "better" relies a lot on datasets and resources. Due to the utilization of enormous CNNs in works like [17], big on-disk models with ~4-5 fps on embedded hardware are required. On the other hand, FireDetector leverages more varied training data and specialized from-scratch architecture to acquire strong capabilities. This approaches human vision cognition by enabling real-time performance up to 24 frames per second. FireDetector thus strikes a good compromise between high-framerate embedded deployment and precision.

VII. CONCLUSION

We introduce FireDetector, a very light-weight neural network trained on a variety of data sets and created from scratch for fire detection. The ultimate objective is to replace physical sensors and handle their delayed or false triggers FireDetector functions well at 24 frames per second. Tests using a standard dataset and difficult real-world images that resemble CCTV cameras demonstrate high F1, recall, accuracy, and precision. We will next improve performance in light of the diversity of datasets. The goal of this endeavor is to create a reliable integrated fire detector that can be monitored remotely.

REFERENCES

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," nature, vol. 521, no. 7553, p. 436, 2015.

[2] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (iot): A vision, architectural elements, and future directions," Future generation computer systems, vol. 29, no. 7, pp. 1645–1660, 2013.

[3] "Raspberry Pi 3 Model B." https://www.raspberrypi.com/products/raspberry-pi-3-model-b/ Accessed: 2023-11-03.

[4] P. Foggia, A. Saggese, and M. Vento, "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, vol. 25, no. 9, pp. 1545–1556, 2015.

[5] T.-H. Chen, Y.-H. Yin, S.-F. Huang, and Y.-T. Ye, "The smoke detection for early fire-alarming system base on video processing," in 2006 International Conference on Intelligent Information Hiding and Multimedia, pp. 427–430, IEEE, 2006.

[6] "Applications for Fire Alarms and Fire Safety." https://www.vent.co.uk/ventilation-controls/ . Accessed: 2023-11-05.
[7] "Fire Accidents." https://urlzs.com/7mTM Accessed: 2023-11-05.
[8] T.-H. Chen, P.-H. Wu, and Y.-C. Chiou, "An early fire- detection method based on image processing," in 2004 International Conference on Image Processing, 2004. ICIP'04., vol. 3, pp. 1707–1710, IEEE, 2004.
[9] T. C¸ elik, H. Ozkaramanlı, and H. Demirel, "Fire and smoke detection ¨ without sensors: Image processing-based approach," in 2007 15th European Signal Processing Conference, pp. 1794–1798, IEEE, 2007.
[10] Rafiee, R. Dianat, M. Jamshidi, R. Tavakoli, and S. Abbaspour, "Fire and smoke detection using wavelet analysis and disorder characteristics," in 2011 3rd International Conference on Computer Research and Development, vol. 3, pp. 262–265, IEEE, 2011.
[11] T. Qiu, Y. Yan, and G. Lu, "An autoadaptive edge- detection algorithm for flame and fire image processing," IEEE Transactions on instrumentation and measurement, vol. 61, no. 5, pp. 1486–1493, 2012.
[12] S. Rinsurongkawong, M. Ekpanyapong, and M. N. Dailey, "Fire detection for early fire alarm based on optical flow video processing," in 2012 9th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, pp. 1–4, IEEE, 2012.
[13] M. Mueller, P. Karasev, I. Kolesov, and A. Tannenbaum, "Optical flow estimation for flame detection in videos," IEEE Transactions on image processing, vol. 22, no. 7, pp. 2786– 2797, 2013.
[14] Mobin, M. Abid-Ar-Rafi, M. N. Islam, and M. R. Hasan, "An intelligent fire detection and mitigation system safe from fire (sff)," Int. J. Comput. Appl, vol. 133, no. 6, pp. 1–7, 2016.
[15] Q. Zhang, J. Xu, L. Xu, and H. Guo, "Deep convolutional neural networks for forest fire detection," in 2016 International Forum on Management, Education and Information Technology Application, Atlantis Press, 2016.
[16] Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in Advances in neural information processing systems, pp. 1097–1105, 2012.
[17] J. Sharma, O.-C. Granmo, M. Goodwin, and J. T. Fidje, "Deep convolutional neural networks for fire detection in images," in International Conference on Engineering Applications of Neural Networks, pp. 183– 193, Springer, 2017.
[18] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014.
[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778, 2016.
[20] K. Muhammad, J. Ahmad, and S. W. Baik, "Early fire detection using convolutional neural networks during surveillance for effective disaster management," Neurocomputing, vol. 288, pp. 3042, 2018.