Deep Learning – Based Fire and Smoke Detection System with MobileNet Architecture

[1]Shruti Kolte, [2]Anurag Mahakalkar, [3]Shreyash Arghode, [4]Geyesh Barsagade, [5]Abhishek Kongare

[1]Assistant Professor, [2,3,4,5]Students
Department of Artificial Intelligence,
Priyadarshini J L College of Engineering
Nagpur, India.

Abstract- **Wildfires, or similar events in other places, are among the most common yet unfavorable occurrences brought on by climate change and rising temperatures. Thus, sophisticated yet user-friendly systems are required, ones that at the very least make it possible to employ modern tools and solutions efficiently. In order to guarantee the security and safety of diverse settings, fire and smoke detection are essential jobs. In this research, we use deep learning techniques to propose a comprehensive solution for smoke and fire detection. The project is being developed in Python, making use of the MobileNet architecture's potent capabilities. Accurately identifying fire and smoke instances in many scenarios—including pictures, videos, and real-time webcam feeds—is the primary goal of this study. The excellent accuracy shows that the model can reliably detect fire, smoke, and typical occurrences in a variety of scenarios. The suggested solution offers real-time analysis of photos, videos, and live webcam feeds in addition to multipurpose detection. This adaptability guarantees that the solution may be applied to a variety of situations, including emergency response management, fire alarm systems, and surveillance systems**.

Key words: **Convolutional Neural network, Deep Learning, Image Classification, Fire Detection, MobileNet Architecture.**

INTRODUCTION

These days, a lot of fire incidents happen often and result in significant financial damages. The creation of fire and smoke detection systems, which are crucial for early warning and disaster avoidance in a variety of situations, including commercial buildings, residential dwellings, and industrial facilities, is one important use of deep learning. This introduction will highlight the importance and advantages of this method by giving a brief overview of a state-of-the-art smoke and fire detection system that uses the MobileNet architecture. Incidents involving fire and smoke can have disastrous results, including fatalities, property damage, and environmental risks. The accuracy and reactivity of traditional fire detection systems are constrained because they frequently rely on rule-based algorithms or human involvement. A potential remedy that makes use of neural networks' capacity to quickly and precisely detect fire and smoke is deep learning-based systems. Deep learning methods have proven remarkably effective in applications involving image recognition and categorization. Convolutional Neural Networks (CNNs) are perfect for fire and smoke detection because they are especially well-suited for visual data interpretation.

These devices provide speedy response and mitigation by swiftly analyzing pictures or video streams to spot possible fire or smoke patterns. MobileNet is a deep learning architecture created especially for embedded systems and cellphones that have limited resources. Its efficiency and speed, combined with competitive accuracy, are its main advantages. Real-time applications can benefit from MobileNet's depthwise separable convolutions, which lessen computational load. Because of these features, MobileNet is an obvious choice for implementing smoke and fire detection systems on edge and mobile devices. To get around these limitations, systems that use video to detect fire are employed. There is a noticeable trend toward adopting new technology as digital cameras and the capacity to handle videos become more advanced. It monitors all activity both inside and outside the door. It is possible to create a competitively-edge, cost-effective video-based fire detection system by utilizing the existing security cameras. There are many benefits when comparing this kind of technology to conventional detecting methods. For instance, using this kind of detection is less expensive and easier to deploy than using antiquated techniques. Second, because it can monitor a greater area and doesn't require any kind of circumstance to activate the devices, a vision-based fire detection system responds faster than other conventional fire detection systems.

LITERATURE SURVEY

Wangda Zhao (2020)[1] Using the faster-RCNN, R-FCN, SSD, and YOLO v3 advanced object identification CNN models, a novel picture fire detection method was developed. The proposed algorithms are able to locate fire in various settings and automatically extract complex picture fire properties. Additionally, the author claimed that CNN-based algorithms outperform traditional algorithms in terms of accuracy. All CNN models combined, the YOLO v3-based algorithm detects fire at a rate of 83.7 percent and 28 frames per second, respectively, making it the most accurate method.

K. Muhammad, J. Ahmad, I. Mehmood, S. Rho & S. W. Baik (2018)[2] recommended a Convolutional Neural Network (CNN) architecture for low-cost fire detection in video surveillance. The proposed model primarily focused on computational complexity and detection precision. The main source of inspiration for the model was the Google Net architecture, which has lower computing complexity than other networks with significant computational expenses, such as AlexNet. The author claims that the proposed framework works better on fire datasets and is suitable for practical applications such as fire detection in CCTV security systems.

Yuming Li, Wei Zhang, Yanyan Liu & Yao Jin (2022)[3]Using the anchor less structure and convolutional neural network MobileNetV3, a fast and efficient fire detection model is created. The suggested approach performs better in two areas. First, the suggested technique will increase network speed because it is tiny enough to be easily implemented on mobile devices with visuals. The accuracy of the model has been evaluated using two publicly accessible fire datasets in addition to self-constructed datasets. With a maximum speed of 29.5 feet per second, the suggested framework could be able to detect fires in real time, which makes it suitable for use in actual applications.

Arpit Jadon, Akshay Varshney, Mohammad Samar Ansari (2020)[4] To overcome the challenges with fire detection, the author presented the MobileNetV2 architecture convolutional neural network model. The author presents a more transparent data handling method along with a unique MobileNetV2 architecture that performs better than existing solutions and is computationally feasible for implementation on less powerful hardware. The efficacy of this model in comparison to existing convolutional neural network models was evaluated using the metrics Accuracy, Precision, Recall, and F-Measure on two datasets. The recommended model has the highest accuracy of any (99%), at 0.99.

Myeongho Jeon, Han-Soo Choi, Junho Lee, Myungjoo Kang (2021)[5] The author proposed a framework that enhances the existing Convolutional neural network-based fire image classification model by highlighting the various sizes of flames in photos. The author proposed employing a feature-squeeze block to incorporate feature maps of various scales in the final forecast. The 13516maps' features are compressed both spatially and channel-wise by the feature-squeeze block, making efficient use of the multi-scale prediction data possible. Using the given methods, the experiment yielded an F1-score of 97. 89% and a false positive rate of 0.0227%.

Qingjie Zhang, Jiaolong Xu, Haifeng Guon (2016, January)[6] A deep learning method was proposed by the study's researcher to identify forest fires. The author trained both a fine-grained patch fire classifier and a full picture classifier in a mixed deep convolutional neural network (CNN). Here, the fire was discovered by the author using a cascade approach. The suggested fire patch detector achieves detection accuracy of 97% and 90% on training and test datasets, respectively.

METHODOLOGY

A) Models

The suggested model analyzes the user-provided image to identify whether or not fire is present. Two main categories of models have been put forth.

1) Data Pre-processing
2) Image Classification

1) Data Pre-processing

Preparing unprocessed data for use with machine learning models is known as data pre-processing. It is the most crucial and initial stage in the development of a machine learning model. When working on a machine learning project, clean, well-prepared data is not always available. Additionally, data needs to be prepped and cleaned before beginning any operation. We therefore make use of data pre-processing services.

STEPS IN DATA PRE-PROCESSING IN MACHINE LEARNING

i) Acquire the dataset:

The first step in machine learning data pre-processing is the gathering of the dataset. The necessary dataset must be gathered before you can start building and refining Machine Learning models. The creation of this dataset will involve gathering data from multiple sources and integrating it into an appropriate manner. A dataset's format varies depending on its intended use.

ii) Import all the libraries:

The most widely used and highly recommended library among data scientists globally is Python. Specialized data pre-processing operations can be carried out using the Python libraries. Importing the required libraries is the second step in the pre-processing of machine learning data. Machine learning frequently uses the following Python libraries for preprocessing data:

a) NumPy - NumPy is the most widely used Python scientific computing tool. As such, it's employed to incorporate any kind of mathematical function into the code. Programmers using NumPy may also work with large multidimensional arrays and matrices.
b) Pandas - Pandas are excellent open-source Python tools for data manipulation and analysis. It is widely used for data import, gathering, and maintenance. It comes with quick and easy-to-use Python data structures and data analysis tools.
c) Matplotlib - A Python 2D charting library called Matplotlib can be used to make a wide range of charts. It can generate publication-quality numbers in a range of interactive and hard copy formats.
iii) Dividing the datasets into the test set and the training set:
We divide our datasets into a training set and a test set for machine learning data preparation. This is an important step in the pre-processing of the data because it improves our machine learning model's functionality. Suppose we trained our machine learning model on one dataset and tested it on another. After then, our model will have trouble understanding the connections between the models. The model's performance will deteriorate if it is properly trained and has a high training accuracy, but then we introduce new datasets. We therefore try our hardest to create a machine learning model that performs well on both training and test datasets.
iv) Data Augmentation:
Data augmentation techniques create several replicas of a real dataset in order to artificially increase its size. To solve data scarcity and inadequate data diversity, data augmentation approaches are used in computer vision and natural language processing (NLP) models. Machine learning models can benefit from data augmentation strategies. An experiment shows that a deep learning model with picture augmentation outperforms a deep learning model without picture augmentation in training and accuracy for image classification tasks, as well as in validation loss and accuracy.
There are various methods for augmenting data:
Adding noise: If a picture is hazy, adding noise could make it stand out. When the image is referred to as "salt and pepper noise," it seems to consist of white and black specks.
Cropping: Selecting a portion of the image, cropping it, then resizing it to its original dimensions,
Flipping: The image has been reversed both vertically and horizontally. While rearranging the pixels, flipping the image preserves its features. There are some photos when there's no use for vertical flipping.
Rotation: From 0 to 360 degrees, the image is rotated by a few degrees. Every rotating image in the model will be distinct.
Scaling: The picture has been scaled both inward and outward. An object can be made smaller or larger in a new image by scaling it from the original.
Translation: The neural network searches the entire image for the image since it is shifted along the x- or y-axis.
Brightness: The brightness of the original image is changed, making the new one either lighter or darker. The model can recognize photos in different lighting conditions thanks to this technique.
Zooming: The image is arbitrarily zoomed and new pixels are added in the Zooming Augmentation technique.

2) Image Classification:

Image categorization is the process of assigning labels to photos so they can be placed into one of several predetermined categories. There are an endless number of categories into which one image might be divided. It takes time to manually evaluate and categorize photos, especially when there are many of them. For this reason, it would be advantageous to automate the process with computer vision.
CNNs, or convolutional neural networks, are widely employed in deep learning picture classification. Not every node in the layer below receives the output from the nodes in CNNs' hidden layers. Through the use of deep learning, machines are able to identify and extract information from photos.
For image classification, we use MobileNet Architecture, a deep learning model based on convolutional neural networks. MobileNet Architecture is an image recognition architecture that has demonstrated accuracy levels greater than 78.1 percent on the ImageNet datasets. A collection of depth-separable convolutional layers is the basis of MobileNets.
Each depth wise separable convolution layer consists of a point wise and a depth wise convolution. A MobileNet contains 28 layers if the depthwise and pointwise convolutions are counted separately. The 4.2 million parameters that comprise a typical MobileNet can be further minimized by appropriately modifying the width multiplier hyper-parameter.

B) MobileNet Architecture

Convolutional neural network (CNN) architecture known as MobileNet was created to effectively classify images and recognize objects on mobile devices with constrained computational power. It is made with the intention of being quick and lightweight without sacrificing accuracy. Depthwise is a new kind of convolutional layer The significantly faster and smaller CNN design known as MobileNet uses separable convolution. Because of their compact size, these models are considered to be very helpful for implementation on embedded and mobile devices. Thus the term MobileNet.
1) Convolution that is separable by depth:
1.1 Convolution by depth

1.2 Convolution by point
2) Dimensions of MobileNet
2.1 Multiplayer in Width
2.2 Multiplayer with Resolution-Aware
1) Convolution that is separable in depth
The two layers that comprise the depth-wise separable convolution are the point-wise and depth-wise convolutions. Essentially, the first layer filters the input channels, and the second layer combines them to create a new feature.
1.1) Depthwise Convolution: In this stage, each input channel is convolved independently using a 3x3 filter, for instance. In light of this, the input tensor is subjected to 'n' independent 3x3 convolutions, one for each of the 'n' input channels. This process helps to extract attributes specific to a given channel.
1.2) Pointwise Convolution: Following depthwise convolution, a 1x1 convolution (pointwise convolution) is employed. Convolution helps create connections and interactions between channels by merging data from several channels.
2) Parameters of MobileNet
The fundamental MobileNet architecture has two unique global hyperparameters that drastically reduce the computational cost, despite being extremely straightforward and requiring little processing power.
There are two types of multiplayer games: width and resolution-wise.
2.1) Width Multiplier: We add a very basic parameter α called width multiplier in order to design these smaller and less computationally expensive models. The function of the width multiplier α is to evenly thin a network at every layer.
$DK \cdot DK \cdot \alpha M \cdot DF \cdot DF + \alpha M \cdot \alpha N \cdot DF \cdot DF$
2.2) Multiplier of Resolution: Resolution Multiplier: The second hyper-parameter to reduce the computational cost of a neural network is a resolution multiplier. The internal representation of each layer is then decreased by the same multiplier when we apply this to the input image. Actually, we achieved this by changing the input resolution.
$DK \cdot DK \cdot \alpha M \cdot \rho DF \cdot \rho DF + \alpha M \cdot \alpha N \cdot \rho DF \cdot \rho DF$

BLOCK DIAGRAM

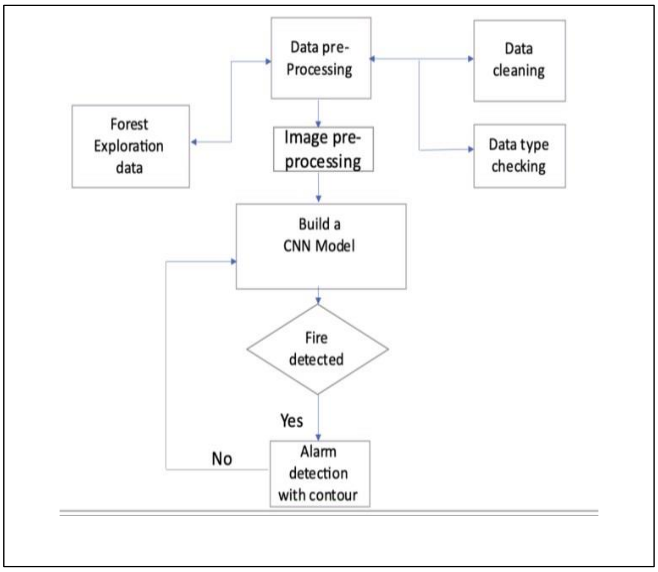

Figure 1: Block Diagram of Project

A) Image Acquisition
It is the procedure for locating an image online. This can be accomplished with hardware systems such as encoders, sensors, cameras, and so on. Using this technique, an image is transferred from a camera to a computer and converted to binary code. The 8 Megapixel JPG image compression standard is the picture format that is being used. During the day, the information is collected in the morning, midday, and evening. More data may be obtained from 100 sample photographs to build a system that performs better.
B) Pre-processing
A stage of image development and enhancement is called pre-processing. Here are steps in this phase:
Read Image:
The libraries are first imported, and after putting the path to the picture dataset in a variable, a method is built to load image folders into arrays.
Resize Image:
Following that, a function is created, which only takes images as an argument. A basic size should be created for all images that are fed into algorithms since certain images taken by cameras and submitted to algorithm fluctuate in size.
Edge Detection:

Edge detection is the process of locating an image's edges. The background, color, and other subtleties are all present in the original image, but the other image merely displays the image's boundaries. These edges are computed by determining the difference in light between picture pixels.
Histogram Equalization:
To increase contrast in photos, a computer technique known as histogram equalization is applied to image processing. It does this by effectively spreading out the most common intensity values and therefore expanding the image's intensity range.

C) Feature Extraction

When less processing power is needed without sacrificing significant or pertinent data, the feature extraction strategy is advantageous. Moreover, feature extraction may facilitate the analysis process by reducing the quantity of redundant data. The computer's efforts to generate variable combinations (features) and the reduction of data will speed up the training method's knowledge and generalization phases.

D) Convolutional Neural Network Architecture

A CNN's typical architecture consists of fully linked, pooling, and convolutional layers.
Layer of Convolution:
This layer performs a dot product between two modules: the limited region of the input patch is represented by the other grid, and the kernel is a set of learnable properties.
Layer of Pooling:
The pooling layer replaces the network's output at some points by computing an aggregate statistic from the surrounding outputs. This helps reduce the spatial size of the representation, which in turn reduces the amount of computation and weights required.
Completely Networked Layer:
This displays the kernel-shaped output that the pooling layer produced.

SCREEN SHOTS

1) We upload the images for detect the fire and smoke after training data set.

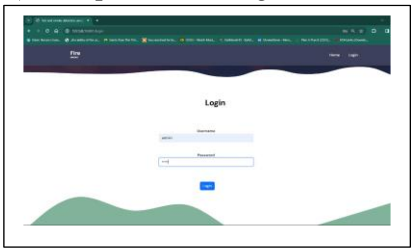

Figure 2: Login Page

Figure 3: Detect Fire

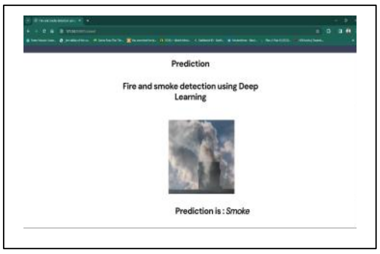

Figure 4: Detect Smoke

Figure 5: Detect Normal

2) Now, detect the fire and smoke through live situation.

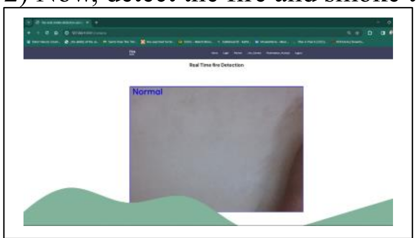

Figure 6: Detect Normal through Live Camera

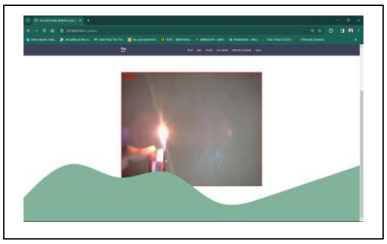

Figure 7: Detect fire through Live Camera

EXPERIMENT ANALYSIS

The two ensemble models that have been given were made using the trained model, which was trained on a thousand photos. The optimal weights are used to extract deep features in the proposed investigation. Two lightweight ensemble models have been provided through the application of the transfer approach. The two model ensembles that have been suggested use NCA as a feature selection paradigm. The method provides the fire's exact location and the approximate amount of water needed to put it out as quickly as possible.

Figure 8: Performance Analysis

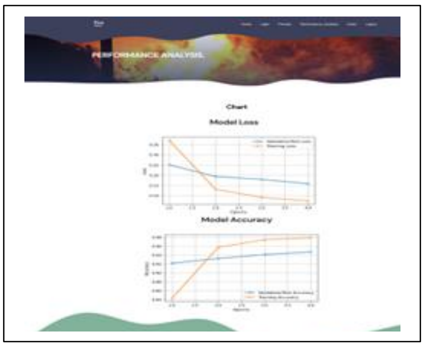

Figure 9: Chart View of Project

CONCLUSION

To sum up, our project on smoke and fire detection utilizing deep learning methods and the MobileNet architecture has advanced computer vision and safety systems significantly. Over the project's duration, we have met the main goals stated at the beginning and accomplished a number of significant benchmarks.

The primary strength of our system is its capacity to deliver extremely precise and dependable fire and smoke detection across a range of media formats, such as still photos, video streams, and live webcam feeds. The system can be used in a variety of situations, including emergency response management, fire alarm systems, and surveillance systems, due to its flexibility and adaptability.

REFERENCES:

1. Wangda Zhao, (2020). Image fire detection algorithms based on convolutional neural networks. Case Studies in Thermal Engineering, val. 19, 100625.
2. K. Muhammad, J. Ahmad, I. Mehmood, S. Rho., & S. W. Baik (2018). Convolutional neural network-based fire detection in surveillance videos. IEEE Access, vol. 6, 18174-18183
3. Yuming Li, Wei Zhang, Yanyan Liu & Yao Jin (2022). A visualized fire detection method based on a convolutional neural network beyond anchor. Applied Intelligence,Volume 52, Issue 11, Sep 2022pp 13280–13295.
4. Arpit Jadon, Akshay Varshney, Mohammad Samar Ansari, (2020). The low-complexity high-performance deep learning model for real-time low-cost embedded fire detection systems. Procedia Computer Science, vol. 171, 418-426.
5. Myeongho Jeon, Han-Soo Choi, Junho Lee, Myungjoo Kang (2021). Multi-scale prediction for fire detection using convolutional neural network. Fire Technology, 57(5), 2533-2551.
6. Qingjie Zhang, Jiaolong Xu, Haifeng Guo (2016, January). Deep convolutional neural networks for forest fire detection. In Proceedings of the 2016 international forum on management, education and information technology application. Atlantis Press.
7. Mohit Dua, Mandeep Kumar, Gopal Singh Charan, Parre Sagar Ravi (2020, February). An improved approach for fire detection using deep learning models. In 2020 International Conference on Industry 4.0 Technology (I4Tech) (pp. 171-175). IEEE.
8. Z. Jiao, Y. Zhang, J. Xin, L. Mu, Y. Yi, H. Liu, & D. Liu (2019, July). A deep learning-based forest fire detection approach using UAV and YOLOv3. In 2019 1st International conference on industrial artificial intelligence (IAI) (pp. 1-5). IEEE.
9. F. Saeed, A. Paul, Karthi kumar& A. Nayyar (2020). Convolutional neural network-based early fire detection. Multimedia Tools and Applications, 79(13), 9083-9099.International Journal of All Research Education and Scientific Methods (IJARESM),ISSN: 2455-6211, Volume 11, Issue 9, September-2023, Available online at: www.ijaresm.comPage | 1719
10. J. Gotthans, T. Gotthans, & R. Marsalek (2020, April). Deep convolutional neural network for fire detection. In 2020 30th International Conference Radio elektronika (RADIOELEKTRONIKA) (pp. 1-6). IEEE.
11. S. Frizzi, R. Kaabi, M. Bouchouicha, J. M. Ginoux, E. Moreau, & F. Fnaiech(2016, October). Convolutional neural network for video fire and smoke detection. In IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society (pp. 877-882). IEEE.
12. M. Iqbal, C. Setianingsih, & B. Irawan (2020, August). Deep Learning Algorithm for Fire Detection. In 2020 10th Electrical Power, Electronics, Communications, Controls, and Informatics Seminar (EECCI) (pp. 237-242). IEEE