Vol.15, Issue No 2, 2025

Deep Learning Techniques for Predicting Forest-Fire Exposure

[1] Vegesna Yorshita Varma, [2] Patsamatla P Sai Lakshmi, [3] Sangameswara Naga Surya Lakshmi Sailaja, [4] Nanda Siva Surya Manikanta, [5] Mrs. N. Bharathi,

[1,2,3,4]Students, Dept. of CSE, DNR College of Engineering & Technology, Balusumudi, Bhimavaram, India.

[5] Assistant Professor, Dept. of CSE, DNR College of Engineering & Technology, Balusumudi, Bhimavaram, India.

Abstract:-

Devastating natural catastrophes of unfathomable scale are caused annually by hundreds of forest fires throughout the world. Many well-researched solutions are either ready to be tested or are in the process of being developed to address this issue. The fire has been detected with the use of sensors. However, for extensive forest areas, this is obviously not an option. Using state-of-the-art technology, we provide a novel method for fire detection in this article. More specifically, we suggested an AI platform.

Using either still photos or video data from the cameras, computer vision systems can identify and detect smoke and fire. One way to determine the fire's intensity is via a "convolution neural network," a deep learning technique. Because of this, forest video surveillance systems will be able to deal with more complicated real-life circumstances. Methodology, datasets, and the separation of data into a "train set" and a "test set" determine the precision.

Index Terms:-Convolutional Neural Networks, deep learning, OpenCV, picture categorization, and fire detection

1.INTRODUCTION

Forests play a crucial role in maintaining Earth's delicate ecological balance. Regrettably, forest fires are often only discovered after they have spread over a considerable region, making their control and extinction very difficult, if not impossible, to achieve. Devastating loss and permanent damage to the ecosystem and climate are the results. Forest fires contribute 30% of atmospheric carbon dioxide (CO_2) and release massive volumes of smoke and carbon dioxide (CO_2), which harm the ecology. The standard practice is to put a stop to illicit logging.

Through the constant recording of forest noise, the system aims to detect potential risks by analyzing portions of the recorded signals and deciding on the nature of each segment [1].

As quickly as possible, get enough fire apparatus and trained operational personnel to the scene of the incident. In addition, there has to be a well-developed system in place to ensure that fire extinguishers are readily available, regularly serviced, and monitored for any signs of fire. Various detection systems, including early detection, remote sensing, logistics, training through simulation, and firefighting vehicles, are part of an integrated approach to forest fire detection and suppression that takes into account wildfire risks, area size, and human presence. Nowadays, IoT (Internet of Things) systems rely heavily on Wireless Sensor Networks (WSNs). Not only do these systems have wide-ranging uses, but they also bring innovative ideas to the environmental monitoring area. The Raspberry Pi Model 3, digital and analog sensors, and signal processing algorithms form the basis of an intelligent forest environment monitoring system. While ambient noise is examined, sensors are used to track variables including soil humidity, temperature, gas concentrations, and more. The building blocks of a functional circuit and sensor modules are integrated in forest fire automation systems. These systems work together to monitor the conditions of a specific forest. The majority of researchers(33%) use WSN for tracking applications, while nearly half(48%) use it for data exchange inside their system and almost half(43%) use it for data transfer between sensor nodes. [4] In order to enhance the accuracy of both training and classification, a robust AdaBoost (RAB) classifier is suggested. [5] With a total of 650,000 neurons and 60 million parameters, the neural network is built up of five convolutional layers, some of which are followed by max-pooling layers, and three fully-connected layers culminating in a 1000-way softmaxn. We used a newly-developed regularization approach for fully-connected layers termed "dropout" to decrease overfitting, and it worked well. Predictions of forest fires have been made using a variety of machine learning approaches, including neural networks, decision trees, and regression, among others. It is unusual to use a semi-supervised rule-based classification approach to determine whether a forest zone is highly active (HA), moderately active (MA), or poorly active (LA)[8]. Early detection and prevention of fire propagation are crucial in preventing forest fires from becoming uncontrolled and spreading over large areas. Combating these catastrophes calls for an all-encompassing, multi-faceted strategy that allows for constant situational awareness and rapid response. Using state-of-the-art technology, we provide a novel method for fire detection in this article. Of special note is the AI platform that we suggested. The computer vision techniques that can identify and locate smoke and fire using either still pictures or video input from the cameras. One way to determine the fire's intensity is by using a "convolution neural network," a deep learning technique.

2. RELATED WORK

2. 1 A lot of work in traditional fire detection has gone into identifying the most important aspects of fire photos. In order to determine if a fire has changed, Chen [7] used a rule-based technique that relies on an RGB and HSI color model that is based on the difference between successive frames. The YCbCr color model was suggested by Celik and Demirel [5] as a general rule-based flame pixel categorization that could distinguish between luminance and chrominance components. Furthermore, Wang [8] used an HSI color model to identify the potential fire region from a picture, and then they computed the flame color dispersion to pinpoint the exact location of the fire. Nevertheless, a number of environmental variables, including illumination and shadow, may compromise color-based fire detection systems. In order to identify fires using supplementary information like the surface, border, and extent of the affected region to color, Borges and Izquierdo [9] used the Bayes classifier. In order to identify fires in specific areas, Mueller [10] suggested a neural network-based approach that makes use of optical flow. This technique can differentiate between static objects and those in motion by combining two optical flow models. Moreover, a multi-expert system was suggested by Foggia [11] that integrates the findings of a fire's form, color, and mobility characteristics. Color alone isn't enough to prevent false positives, but form, texture, and optical flow may help. However, these methods aren't great at reflecting the spatial and temporal information relevant to fire habitats, and they need domain knowledge of flames in collected images—which is crucial for investigating hand-crafted features. Furthermore, the majority of technologies that rely on the traditional approach to fire detection rely only on still images or consecutive pairs of frames. Consequently, they fail to take into account the longer-term dynamic behavior of fires, focusing instead on the short-term dynamics.

2.2 DEEP LEARNING-BASED APPROACH

Recent years have seen widespread use of deep learning in fields as varied as picture object identification and classification, voice recognition, and NLP. Fire detection based on deep learning has been the subject of several research aimed at improving performance. Compared to the traditional computer vision-based fire detection method, the deep learning methodology differs in a number of important respects. One advantage is that the traits aren't hand-picked by a human but rather learned by the network itself via extensive and varied training. So, instead of focusing on finding the right handmade features, you should focus on building the right network and getting the training data ready. One other distinction is that the detector/classifier may be acquired by training the features in the same neural network concurrently. Consequently, with an effective training algorithm, the importance of the correct network architecture increases. In his CNN-based fire detection network, Sebastien [12] suggested training a Multilayer Perceptron (MLP)-type neural net classifier using the features all at once. A CNN-based cascaded fire detection approach was also suggested by Zhang et al. [13]. Their approach begins with a full-image test using a global image-level classifier; if a fire is identified, a fine-grained patch classifier is used to pinpoint the exact locations of the patches. One such system that Muhammad et al. [14] suggested using a CNN fire detector that has been fine-tuned is a fire surveillance system. Drawing inspiration from the Squeeze Net [15] design, this architecture is a CNN that efficiently detects fires, locates them, and understands their semantics at the scene.

A CNN unit's activation may be seen as a feature containing a broad region of context information due to its wide receptive field in the CNN's deep layer. One other benefit of using CNN-learned features for fire detection is this. Locating objects has also been an issue, despite CNN's much better categorization performance as compared to conventional computer vision approaches. The suggested approach uses an object detection model to identify SRoFs and non-fire objects; the former group comprises things like smoke and flames, while the latter group consists of

things that are unrelated to the fire. The items that aren't related to the fire might cause more false alarms because of changes in shadow and light, and they're likely to pick up on things like sunsets, red cars, and red clothing. Even though it is not limited to object identification, we use the Faster R-CNN model to identify fire items. Convolutional neural network (CNN) feature extractors, a localizer with a classifier, and finally the deep object detector are the typical components of a single-stage or multi-stage implementation. Hence, our object identification approach incorporates a feature extractor that can acquire additional contextual information and has a considerably larger receptive field than the identified SRoF region. While convolutional neural networks (CNNs) perform well, recursive-type neural networks (RNNs) easily capture the dynamic behavior of fire. Hochreiter and Schmidhuber [16] presented LSTM, an RNN model that addresses the issue of RNN vanishing gradients. LSTM's memory cells store internal states and repeated behavior, allowing it to collect temporal characteristics for decision making. Unfortunately, decision-making relies on long-term dynamic behavior, which is sometimes difficult to depict because to the restricted number of recursions. Decisions based on LSTM's long-term behavior must, therefore, be carefully considered. A recent study by Hu et al. [17] used LSTM for fire detection. In this method, CNN features are retrieved from optical fluxes of successive frames and then stored in an LSTM network over time.
Fusion of consecutive temporal characteristics is what ultimately leads to a choice. They avoid directly utilizing RGB frames in favor of computing the optical flow, which is used to prepare the input of CNN.

This research paper incorporates the best features of previous works that have reviewed real-time fire detection methods using convolutional neural networks on video sequence frames. It also incorporates proximity detection of fires using USB cameras, which fuses training and test data of fire signatures for early detection and alerts notification. Additionally, it provides navigational aid to the scene of the fire so that the fire rescue team can respond appropriately. Using cameras and video analysis is an innovative approach that gets beyond the issues that were brought up in the fire detection systems study.

3. SYSTEM DESIGN AND DEVELOPMENT

3.1. SYSTEM OVERVIEW

Figure 1 shows a high-level perspective of the fire detection system's software implementation and hardware module architecture. The parts of the fire alarm system that are physically present, as seen in figure 2. Computers may now function as videophones or videoconferencing stations thanks to webcams, which record video and connect to computers or networks using video connections (often a USB port). In addition to security monitoring and video file recording, webcams may be utilized with a variety of computer video telephony applications.

Overarchingly, it uses a Convolutional Neural Network (CNN), a Deep Learning technique for fire detection, and communicates with an open CV module over a USB connection. The whole thing is powered by an Arduino Uno.
At regular intervals, the microcontroller reads data from the sensors and passes it on to the CNN program. Management information systems (MISs) notify the building's inhabitants and the closest fire station of a potential fire if the system detects fires. In the event that the data link is unable to transmit the message, it will transmit it over the short messaging service (SMS). The hardware parts of the fire detection system are the USB camera and the arduino microcontroller board. The software parts, which really power the system, are what make up the CNN fire detection algorithm.
The software subsystem is the non-physical component of the fire detection unit. It is responsible for taking web camera inputs, analyzing them to see whether they indicate a fire, and then sending out warnings when a fire is detected. One such library is OpenCV, which stands for "Open Source Computer Vision." Its primary goal is to facilitate image processing and real-time computer vision. All operations pertaining to images are mostly performed using it. With the help of pixels, machines can observe everything and turn that vision into statistics.

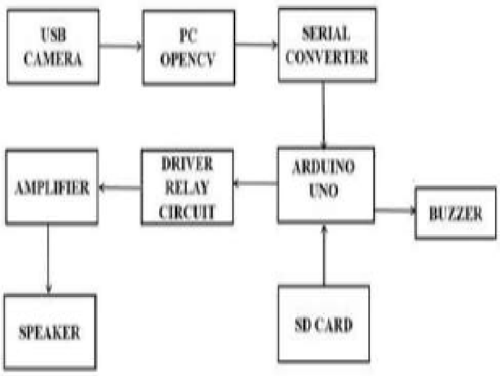

Figure1. Block Diagram of Fire detection system

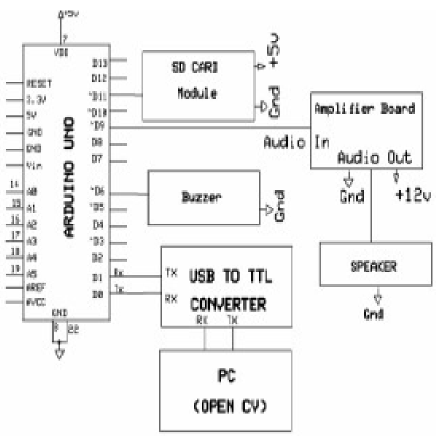

Figure 2. Circuit diagram of Hardware circuit

3.1.2.CONVOLUTIONAL NEURAL NETWORKS:

A type of Deep Learning algorithm, a Convolutional Neural Network (CNN) can distinguish between different objects and aspects in an input image by assigning them learnable weights and biases. When compared to other classification techniques, ConvNets need much less preparation. While filters in primitive techniques are hand-engineered, CNN can learn these filters and attributes with enough training. Here is the construction of CNN, as shown in Figure 3. There are a number of different three-dimensional planes that make up the network's layers. With its many neurons per 3-dimensional plane, convolutional neural networks (CNNs) are well-suited to process picture input. The CNN's input layer, which is a three-dimensional matrix, is where the picture data should be stored. To execute the convolution operation and compute the dot product between the receptive field and the filter, a portion of the image is linked to the Convo layer, which is known as the feature extractor layer. In order to decrease the spatial volume of the input picture after convolution, a pooling layer is used in between two convolution layers. Both the "Filter" and "Stride" hyperparameters are present (S). The components of a fully linked layer include neurons, biases, and weights. It bridges the gap between neurons in different layers. Through training, it is able to categorize images into various groups. The last CNN layer is the Softmax or Logistic layer. There it is, at the very bottom of the FC layer. Binary classification is handled by logistic, while multiclassification is handled by softmax.

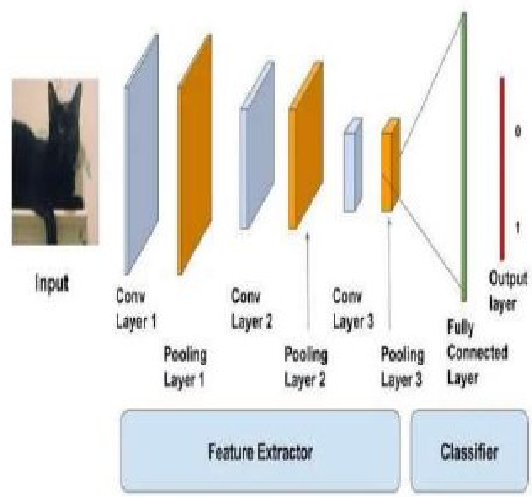

Fig3. Structure of CNN

4. SYSTEM ARCHITECTURE SYSTEM IMPLEMENTATION AND TESTING

To create feature maps, a convolution process applies input data to many kernels of varying sizes. The feature maps are then sent into the subsampling or pooling algorithm, which selects the most activations from a narrow neighborhood. These procedures are crucial for attaining translation invariance to a certain extent and lowering the dimensionality of feature vectors.

The fully-connected layer is an additional crucial component of the CNN pipeline. It is here that the input data is represented into high-level abstractions. Neurons in the fully connected and convolution layers learn and modify their weights during training to better reflect the input data, making them one of the three basic activities.

There are software and hardware parts to the system architecture. The fire detection system's hardware components include the software components that will be installed. An integral component of the hardware system is a surveillance camera unit, which will keep a constant eye on the premises and transmit the footage to a central server. This will greatly improve the system's ability to identify fire incidents and send out alarm notifications.

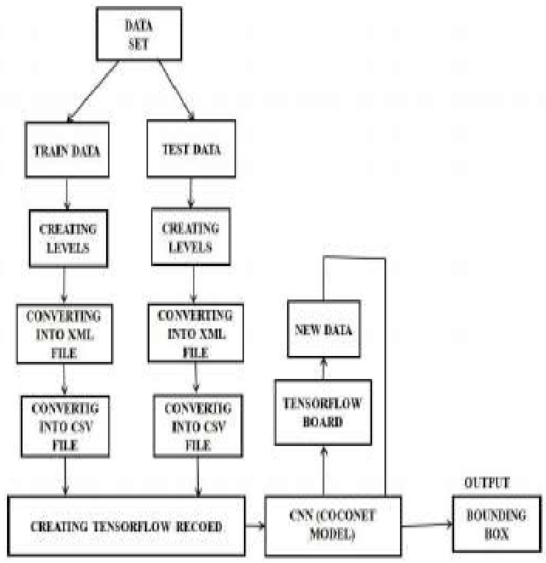

Figure 4: Flow diagram for fire detection software

The purpose of developing the object detector is to educate a reliable classifier. We need an abundance of images that ought to exhibit profound dissimilarity from one another. So, they should all be in various lighting situations, with varied backdrops and random objects. Figure 1 shows the various fire classifications of the samples.

Figure 5: Sample images for classes.

The majority of the photographs should be placed in the object detection/images/train directory, while a smaller percentage should be placed in the object detection/images/test directory. We need picture labeling software of some kind to categorize our data. An excellent tool for classifying images is labeling. The "Create RectBox" button may be used to generate the bounding box. It is necessary to click "Save" after making the bounding box and adding annotations to the picture. Every single picture in the test and training folder has to go through this procedure again. Now that the photos have been identified, we can start generating TFRecords. These will be used to train the object detector. We will use two scripts from Data Tran's raccoon detector to generate the TFRecords. We may utilize the code we have generated to test our newly constructed object detector. Once the video stream is sent to the server, the CNN model is applied to the data. The warning system is activated in the event that a fire is detected.

5.SYSTEM IMPLEMENTATION AND TESTING

More than a thousand photos showcasing both fire and non-fire scenarios were used to train the convolutional neural network. The information came from picture databases accessible online. Figure 5 displays several example photos that were used by the convolutional neural networks.

Datasets are often divided into training and testing sets when neural networks are being trained. In order to derive an inference from the model, the system was fed test video streams as input.

The classifier sorts the input data into two categories: "fire" and "no fire." As a consequence, the classifier takes into account the class that has the highest likelihood score. Tensor Flow, developed by Google, was used to create the classifier module. When it comes to numerical calculation utilizing data flow graphs, Google has you covered with their open-source software package called Tensor Flow. A classifier with 94% accuracy was obtained after training the network on the data. Preprocessing the video stream and extracting frames from it are the first steps in the classification process. To ascertain the state of the region, the retrieved frames/images are categorized.

6.RESULTS AND DISCUSSIONS

This work's overarching goal is to lay out a procedure for constructing a full-fledged fire detection unit that can be easily implemented on an embedded device. Consequently, it's necessary to use a test dataset that contains real-world fire emergency photographs captured using low-cost hardware, such as the Arduino Uno, a microcontroller board based on the ATmega328P, in order to ensure accurate results. The tests conducted on the classifier module revealed that the video classifier functioned well. The classifier confidence level was chosen to prevent false alarms from being activated. Accordingly, the confidence must be higher than or equal to the threshold in order for the alert to be activated. Accurately detecting a fire in the live video feed and promptly sounding an alarm is the goal. The classifier's performance was improved by eliminating superfluous nodes in the TensorFlow module using the "optimize_for_inference" script. To further improve the model's performance, the script performs normalization operations on the convolutional weights, among other optimization procedures.

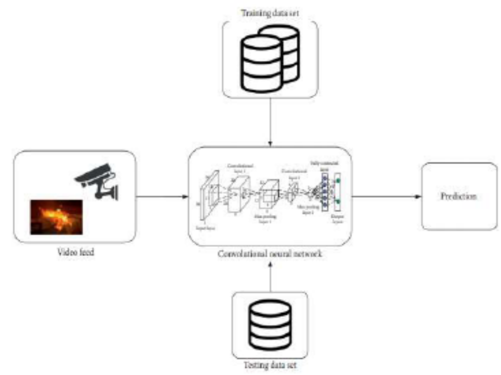

Figure 6: Video processing unit on a dedicated server.

7.CONCLUSION AND RECOMMENDATION

We introduce a Convolutional neural network (CNN) trained on a varied dataset and constructed entirely from scratch. The end goal of this project is to create an IoT-enabled fire detection system that can decrease the issues of delayed and erroneous triggering and successfully replace the present physical sensor-based systems. The newly-introduced neural network may operate at 24 frames per second on an inexpensive embedded device such as the Arduino Uno, a microcontroller board based on the ATmega328P. The accuracy of the model's results on both a regular fire dataset and a custom-built test dataset that mimics the camera quality of an Arduino-connected device, which includes both tough real-world fire and non-fire photos. In addition, the detecting device may provide the user visual feedback and a fire warning in the event of a fire emergency thanks to the Internet of Things feature. Despite the fact that this study enhanced the accuracy of flame detection, there is still a large number of false alarms and further research is needed in this area. Furthermore, it is possible to intelligently adapt existing flame detection frameworks to detect fires. Because of this, forest video surveillance systems will be able to deal with more complicated real-life circumstances.

References:

[1] World fire statistics CTIF—international association of fire services for safer citizens through skilled firefighters, 2019, https://www.ctif.org/worldfire- statistics.

[2] Fire outbreaks cost Ghana over GHC1 million in 2012 general news 2012 https://www.ghanaweb.com/GhanaHomePage/

NewsArchive/Fire-outbreaks-cost-Ghana-over-GHC1- million-in2012-253460

[3] S.-J. Chen, D. C. Hovde, K. A. Peterson, and A. W. Marshall, "Fire detection using smoke and gas sensors," Fire Safety Journal, vol. 42, no. 8, pp. 507–515, 2007.

[4] Z. Tang, W. Shuai, and L. Jun, "Remote alarm monitor system based on GSM and ARM," Procedia Engineering, vol. 15, pp. 65–69, 2011.

[5] R. C. Luo and K. L. Su, "Autonomous firedetection system using adaptive sensory fusion for intelligent security robot," IEEE/ASME Transactions on Mechatronics, vol. 12, no. 3, pp. 274–281, 2007.

[6] K. L. Su, "Automatic fire detection system using adaptive fusion algorithm for fire fighting robot," in Proceedings of the 2006 IEEE International Conference on Systems, Man and Cybernetics, pp. 966–971, Taipei, Taiwan, October 2006.

[7] Chen, T.H.; Wu, P.H.; Chiou, Y.C. An early firedetection method based on image processing. In Proceedings of the International Conference on Image Processing (ICIP), Singapore, 24–27 October 2004; pp. 1707–1710.

[8] Wang, T.; Shi, L.; Hou, X.; Yuan, P.; Bu, L. A new fire detection method based on flame color dispersion and similarity in consecutive frames. In Proceedings of the 2017 Chinese Automation Congress (CAC), Jinan, China, 20–22 October 2017; pp. 151–156.

[9] Borges, P.V.K.; Izquierdo, E. A probabilistic approach for vision-based fire detection in videos. IEEE Trans. Circuits Syst. Video Technol. 2010, 20, 721–731. [CrossRef]

[10] Mueller, M.; Karasev, P.; Kolesov, I.; Tannenbaum, A. Optical flow estimation for flame detection in videos. IEEE Trans. Image Process. 2013, 22, 2786–2797. [CrossRef] [PubMed]

[11] 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]

[12] 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, 2016; pp. 877–882.

[13] Zhang, Q.; Xu, J.; Guo, H. Deep convolutional neural networks for forest fire detection. In Proceedings of the 2016 International Forum on Management, Education and Information Technology Application, Guangzhou, China, 30–31 January 2016; Atlantis Press: Paris, France, 2016.

[14] Muhammad, K.; Ahmad, J.; Lv, Z.; Bellavista, P.; Yang, P.; Baik, S.W. Efficient Deep CNN-Based Fire Detection and Localization in Video Surveillance Applications. IEEE Trans. Syst. Man Cybern. Syst. 2018, 99, 1–16.

[15] Convolutional Neural Networks Based Fire Detection in Surveillance Videos Khan Muhammad 1 , Jamil Ahmad1,Irfan Mehmood 2.Seungmin Rho 3 .Sung WookBaik1,IEEEAccessNo.2016R1A2B4011712).Digital Object Identifier10.1109/ACCESS.2018.2812835