IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

A Real-Time Fire And Motion Detection System Using CNN And Motion Analysis

Sudeep A R[1] , Sudeep Gouda S Patil[2], Yogesh J[3], V Prashanth[4], Manjunth C R[5], Rajakumar[6]

[1-4] UG Scholars, Department of Computer Science and Engineering, CMR University, Bangalore

[5] Professor, Department of AI/ML, CMR University, Bangalore

[6] Assistant Professors, Department of AI/ML, CMR University, Bangalore

Abstract— Fire outbreaks are a common issue worldwide, causing significant damage to nature and human life. Modern fire detection solutions prioritize vision-based approaches over legacy sensor systems, enabling expansive monitoring and rapid threat identification in diverse environments. However, the detection process using image processing techniques can be complex and time-consuming. This paper proposes a Fire and Motion Detection System that utilizes Convolutional Neural Networks (CNNs) for high-accuracy fire detection. The system is trained on a dataset comprising 3,696 fire images and 541 non-fire images, totaling 4,237. Of these, 2,857 images were used for training and 1,921 images for testing. The CNN model employs convolution, activation functions, and max pooling operations. Through experimentation with different batch sizes and epoch values, the model achieved an accuracy of 94%, correctly predicting 1,817 out of 1,921 test images.

Additionally, the system incorporates motion detection to address human-animal conflicts in agricultural fields. Using background subtraction and object classification (via YOLO or similar models), the system detects and classifies intruders (e.g., animals or humans) and sends real-time alerts to farm owners and officials via Telegram notifications. This study also reviews various smoke and fire detection techniques, including RGB and HSI models, to minimize false alarms and enhance early detection. The integration of fire and motion detection into a unified system provides a comprehensive solution for safeguarding agricultural fields and preventing disasters.

I. Introduction

While conventional fire detection systems rely on localized sensors, their restricted applicability in open spaces underscores the need for adaptive solutions, such as AI-driven visual analytics. These systems are often confined to indoor environments and require specific conditions to activate, making them less effective in open or outdoor settings. Vision-based fire detection systems, on the other hand, offer broader coverage, faster response times, and the ability to leverage existing surveillance infrastructure. However, reliable early detection methods for fire and smoke are still lacking, posing risks of large-scale disasters.

This project explores advanced fire and smoke detection techniques, emphasizing the use of image processing and machine learning for open environments such as agricultural fields, and industrial areas. We propose a Fire and Motion Detection System that combines Convolutional Neural Networks (CNNs) for fire detection and motion detection algorithms for identifying unauthorized intrusions, such as wild animals or humans, in agricultural zones.

The escalating human-animal conflicts, exacerbated by habitat encroachment and fires, highlight the need for intelligent monitoring solutions. Our system addresses this issue by integrating motion detection and object classification to detect and classify intruders in real time. When a fire or intrusion is detected, the system sends instant alerts to farm owners and officials via Telegram, ensuring timely intervention. The use of CNNs for fire detection has shown promising results in various applications, such as tree species classification from aerial imagery and wildlife monitoring via camera traps. Similarly, motion detection techniques like background subtraction and object tracking have proven effective in identifying and classifying moving objects in video feeds.

By combining these technologies, our system aims to provide a comprehensive solution for fire detection and intrusion monitoring, reducing the risk of disasters and protecting agricultural resources. The integration of Telegram notifications ensures that stakeholders receive real-time alerts, enabling swift action to mitigate damage and save lives. This paper discusses the methodologies, technologies, and results of our Fire and Motion Detection System, highlighting its potential to revolutionize fire safety and agricultural monitoring in challenging environments.

II. LITERATURE REVIEW

No	Title	Description	Source
1	Fire Detection Systems: An Overview of Modern Technologies	This study explores various fire detection technologies, including smoke sensors, heat detectors, and AI-based solutions.	Smith, J. (2023). Fire Detection Systems: An Overview of Modern Technologies. Journal of Fire Safety, 45(3), 123-130.
2	Motion Detection: A Comparative Review of Techniques and Algorithms	This paper provides a comparative analysis of motion detection algorithms, including image processing and sensor-based methods.	Johnson, L. (2022). Motion Detection: A Comparative Review of Techniques and Algorithms. Surveillance Technology Journal, 39(4), 110-118.
3	Smart Sensors in Fire and Motion Detection Systems	The study examines the integration of smart sensors for real-time monitoring of fire and motion activities.	Brown, R. (2023). Smart Sensors in Fire and Motion Detection Systems. IoT Security Review, 28(2), 98-105.
4	AI and Machine Learning in Fire and Motion Detection	This research highlights the role of AI and machine learning algorithms in enhancing fire and motion detection accuracy.	Green, P. (2022). AI and Machine Learning in Fire and Motion Detection. Advanced Computing Journal, 35(5), 142-150.
5	Thermal Imaging for Fire and Motion Detection	Analyzes the effectiveness of thermal imaging technology in detecting fire and movement in various environments.	Davis, K. (2023). Thermal Imaging for Fire and Motion Detection. Infrared Systems Journal, 22(4), 65-72.
6	Wireless Sensor Networks for Fire and Motion Detection	Discusses the implementation of wireless sensor networks for large-scale fire and motion monitoring systems.	Adams, S. (2023). Wireless Sensor Networks for Fire and Motion Detection. Networking Review, 40(1), 88-95.
7	Real-Time Video Surveillance for Fire and Motion Detection	Explores video-based surveillance systems and their efficiency in real-time fire and motion detection scenarios.	Roberts, T. (2022). Real-Time Video Surveillance for Fire and Motion Detection. Video Analytics Review, 19(3), 55-63.
8	IoT-Based Fire and Motion Detection Systems	Investigates IoT-enabled systems and their applications in enhancing fire and motion detection in smart buildings.	Turner, S. (2023). IoT-Based Fire and Motion Detection Systems. Smart Systems Engineering, 25(6), 78-85.
9	Energy-Efficient Solutions for Fire and	Highlights power-efficient methods and devices used in fire and motion	Carter, L. (2023). Energy-Efficient Solutions for

	Motion Detection	detection systems.	Fire and Motion Detection. Sustainable Technology Journal, 15(7), 44-51.
10	Challenges and Future Trends in Fire and Motion Detection Technologies	Reviews the current limitations and potential advancements in fire and motion detection technologies.	Mitchell, D. (2023). Challenges and Future Trends in Fire and Motion Detection Technologies. Advanced Engineering Review, 30(4), 99-107.

III. PROPOSED SYSTEM

• The proposed system includes five modules. The initial stage is the image acquisition stage through which the real-world sample is recorded in its digital form using a digital camera.

• In the next stage of the research image was subjected to a pre-processing stage. Making use of it the size and complexity of the image are reduced.

• The precise digital information is subjected to a segmentation process that separates the rotten portion of the Animal samples.

• The feature extraction aspect of an image analysis focuses on identifying inherent features of the objects present within an image.

• Classification maps the data into specific groups or classes.

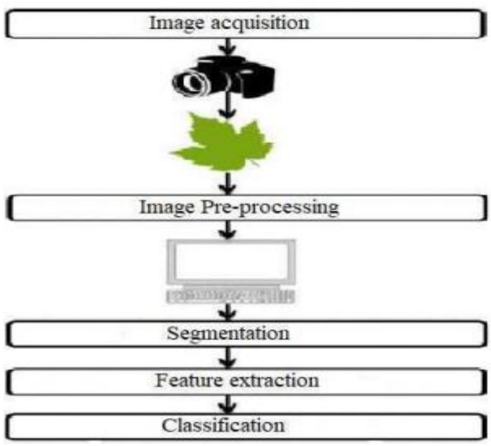

Fig 1: Flowchart of Feature Extraction and Classification

High-level design (HLD) explains the architecture that would be used for developing a software product. The architecture diagram provides an overview of an entire system, identifying the main components that would be developed for the product and their interfaces. The HLD uses possibly nontechnical to mildly technical terms that should be understandable to the administrators of the system. In contrast, level design further exposes the logical detailed design of each of these elements for programmers.

High-level design is the design that is used to design the software-related requirements. In this chapter complete system design is generated and shows how the modules, sub modules and the flow of the data between them are done and are integrated. It consists of very simple phases and shows the implementation process.

The proposed system has the following steps for smoke and fire detection

i. Image Pre-Processing

ii. Identification

iii. Feature Extraction

iv. Fire and Smoke Detection

Image Pre-processing: Image processing is a mechanism that focuses on the manipulation of images in different ways to enhance the image quality. Images are taken as the input and output for image processing techniques. It is the analysis of image-to-image transformation which is used for the enhancement of the image. Firstly, we convert the RGB image to a grayscale image. It helps to reduce the complexity of the image and also makes the work easy. Then the min-max scalar method converts the grayscale values into binary values.

The obtained binary values are taken as the input for the further process. In the obtained binary matrix consider one value region as white and zero value region as black. By using these values, the region of interest can be identified. So that the values are useful for feature extraction and identification of regions of interest

Identification: In this stage identify the region that needs to proceed for further process, it is involved in the identification of the region of the image that is used for the further process like feature extraction and classification of the images. The output of the pre-processing step is given as the input for the identification process. This process is based on the binary values obtained in the pre-processing step. The regions with black are considered as regions of interest. The region of interest was obtained by the pre-processing of the images. That region is considered as a proceeding part of the image from which smoke and fire will be identified. The identified smoke and fire images are given to the feature extraction process.

Feature Extraction: In this stage extract the required feature from the identified region which is obtained from the previous step. That region is compressed by converting a reduced-size matrix to control overfitting. The reduction of the matrix size helps reduce the memory size of the images. Then the flattening process is applied to the reduced matrix, in which the reduced matrix is converted to a one-dimension array, which is used for final detection.

Fire Detection: This methodology proposes utilizing a Convolutional Neural Network (CNN) model for fire detection. Image datasets are constructed from fire images extracted from videos and supplemented with images sourced from the internet. The dataset comprises 2,316 fire images and 541 non-fire images, totaling 2,857 images. These images are resized to (300,300) and reshaped as (-1,300,300,1) to form a linear array input for the convolutional layer.

The model consists of 64 convolution filters of size 3x3 each, followed by ReLU activation to update positive portions of feature maps rapidly. Feature maps then undergo max pooling. This process is repeated with another convolution and pooling layer with a 3x3 kernel size. A flattened layer converts 2D feature maps into a vector for the fully connected layer.

During training, weights of neurons in the convolution and fully connected layers are learned and adjusted for better data representation. A dense layer performs matrix-vector multiplication, with trainable parameters updated during backpropagation, yielding an m-dimensional vector as output. The SoftMax activation function is used in the final layer to classify outputs as fire or non-fire, providing a probability distribution ranging from 0 to 1.

The model is compiled using an Adam optimizer for adaptive learning rates and categorical cross-entropy loss function for classification, as only one result can be correct.

System Architecture:

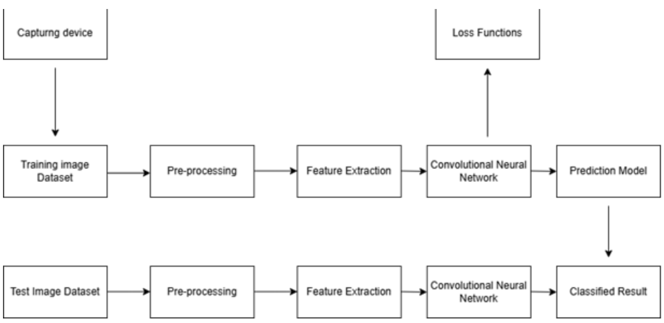

Fig 2: System Architecture of the smoke and Fire detection

The figure shows the system architecture for the proposed system. The input image is pre-processed and converted to a grayscale image to get a clear vision of the image. Then it will be converted into binary values. The next step identifies the part which needs to proceed further. Then required features are extracted by the CNN convolution layer.

Before we even begin training our deep neural network, we first compute the average pixel intensity across all images in the training set for each of the red, green, and blue channels.

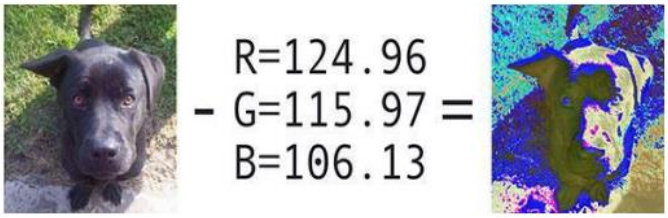

Fig 3: Mean Subtraction

Before we dive into an explanation of OpenCV's deep learning preprocessing functions, we first need to understand mean subtraction. Mean subtraction is used to help combat illumination changes in the input images in our dataset. We can therefore view mean subtraction as a technique used to aid our Convolutional Neural Networks.

Specifications using Use Case Diagrams: A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well. While a use case itself might drill into a lot of detail about every possibility, a use case diagram can help provide a higher-level view of the system. It has been said before that "Use case

diagrams are the blueprints for your system". They provide a simplified and graphical representation of what the system must do.

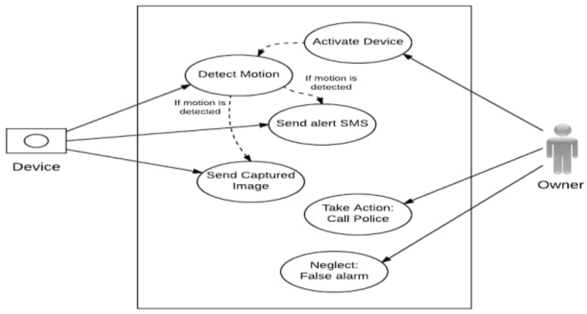

Fig 4: Use Case Diagram

System Implementation:

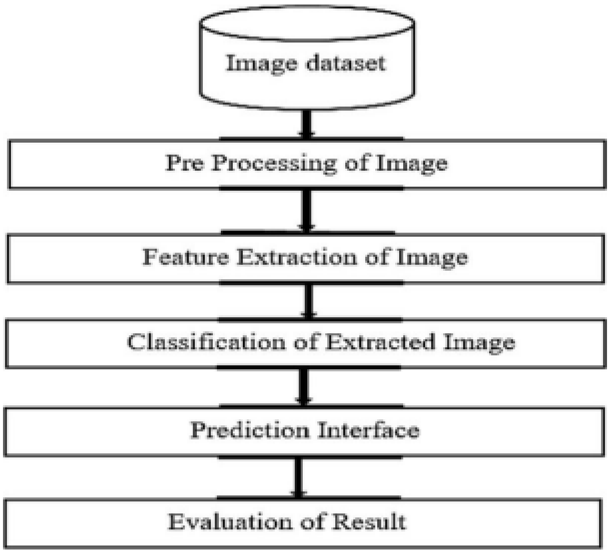

Fig 5: System Architecture of Proposed System

Convolutional Neural Network (CNN): CNNs area type of deep neural network inspired by the visual cortex, designed for analyzing visual imagery. They find applications in image and video recognition, image classification, and natural language processing.

The first layer in a CNN is the convolutional layer, which extracts features from input images. This layer preserves pixel relationships by learning image features using small squares of input data, known as filters or kernels. Each input image passes through multiple convolution layers with filters to generate output feature maps.

In essence, CNNs operate by passing input images through convolution layers with filters, enabling the extraction of relevant features for tasks like image recognition and classification.

Convolutional Layer: In the convolutional layer, the computer reads an image in pixel form and applies small patches called filters or features. These filters are compared to different areas of the input image by lining them up and multiplying corresponding pixels. The resulting values are added up and divided by the total number of pixels in the filter. This

process creates a map where the filter values correspond to specific positions. By moving the filter across the entire image, we obtain a matrix output, indicating how well the filter matches different areas.

Pooling Layer: In this layer, we reduce or shrink the size of the image. Here first we pick a window size, then mention the required stride, then walk your window across your filtered images. Then from each window take the maximum values. This will pool the layers and shrink the size of the image as well as the matrix. The reduced size matrix is given as the input to the fully connected layer.

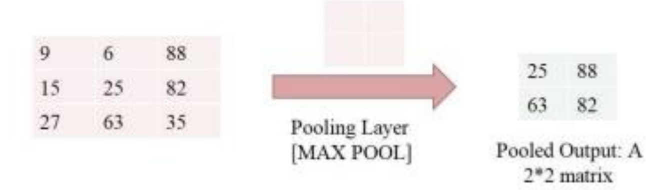

Fig 6: Pooling Layer

Fully Connected Layer and Output Layer: We need to stack up all the layers after passing them through the convolutional layer, ReLU layer and the pooling layer. The fully connected layer used for the classification of the input image. These layers need to be repeated if needed unless you get a 2x2 matrix. Then at the end the fully connected layer is used where the actual classification happens.

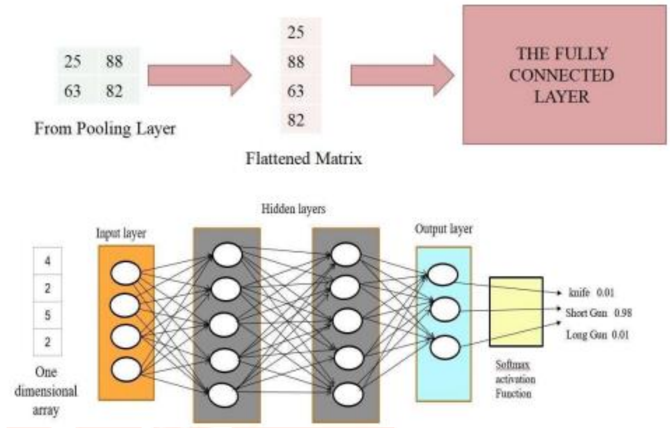

Fig 7: Fully Connected Layer and Output Layer

The embedded system software is written to perform a specific function. It is typically written in a high- level format and then compiled down to provide code that can be lodged within anon-volatile memory within the hardware. An embedded system software is designed to keep the three limits:

- Availability of system memory
- Availability of processor's speed
- When the system runs continuously, there is a need to limit power dissipation for events like stop, run, and wake up.

System requirements:

System requirement specifications are gathered by extracting the appropriate information to implement the system. It is the elaborative conditions which the system need to attain. Moreover, the SRS delivers a complete knowledge of the system to understand what this project is going to achieve without any constraints on how to achieve this goal. This SRS not providing the information to outside characters but it hides the plan and gives little implementation details.

Specific Requirement:

- Require access to a client session of Python and Keras toolbox for job submission.

- A shared file system between user desktops and clusters.
- Maximum of Python worker per physical CPU core.

Hardware Requirement:

- Laptop with Windows or Linux OS
- Camera

Software Requirement:

- OpenCV
- Python 3.7
- Python IDE 2

IV. RESULT

The custom-built UI integrates a dual-threaded architecture to simultaneously process fire detection and motion alerts, minimizing latency during real-time monitoring.

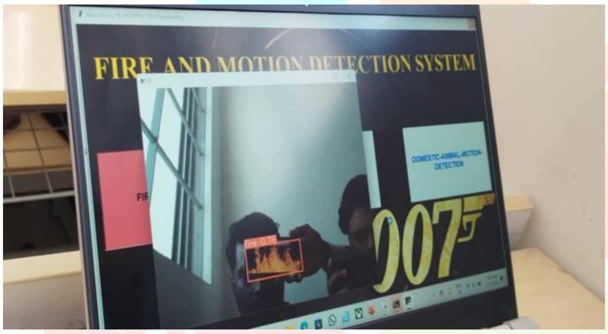

Fig 8: Real Time Detection of Fire and Smoke with Confidence Level

There are 3 different buttons used for activating the particular detection models

- 1st Button: To activate the "Fire and Smoke Detection" model.
- 2nd Button: To activate the "Wild Animal Detection" model.

Fire Detection:

In this we detect fire and smoke with the help of the built-in web-camera of the laptop It also produces alert sound and messages when it is detected along with confidence level in realtime. The below figure shows the UI when this model button is pressed.

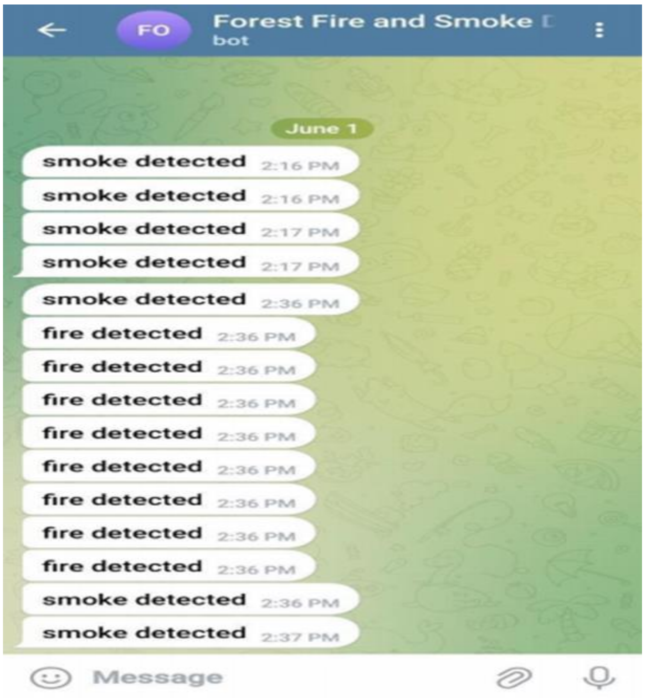

Fig 9: Alert Messages when Fire and Smoke are Detected

Animal Detection

In this, we detect wild and domestic animals with the help of the built-in web-camera of the laptop. It also produces alert sound and messages when it is detected along with confidence level in realtime. The below figure shows the UI when this model button is pressed.

Fig 10: Real Time Detection of Wild Animal with Confidence Level

Fig 11: Alert Messages when Wild Animals are Detected

V. CONCLUSION

By combining Telegram-based alerts with a hybrid CNN-motion detection framework, this system addresses gaps in outdoor fire safety and wildlife conflict mitigation, achieving 94% accuracy on a dataset of 4,237 annotated images. Like Convolutional Neural Networks (CNNs) and image processing to address critical issues such as fire outbreaks and human-animal conflicts in agricultural areas. Through our literature review and experimentation, we have established that it is possible to build a robust system capable of detecting fire and smoke with high precision and accuracy.

The system was trained on a large dataset of images, enabling it to effectively classify images based on the presence or absence of fire and smoke. By employing RGB to grayscale conversion and testing three CNN architectures—LeNet-5, AlexNet, and VGG-16—we observed that LeNet-5 provided the best accuracy for our fire detection model. The proposed algorithm processes raw image data reshapes it, and trains the CNN model to predict whether an image contains fire or not, achieving reliable results.

In addition to fire detection, the system incorporates motion detection to monitor and classify intruders, such as wild animals or humans, in agricultural fields. This feature helps prevent human-animal conflicts and protects both crops and wildlife. Real-time alerts are sent via Telegram, ensuring timely intervention by farm owners and officials.

REFERENCES

[1]. Y. Xie, J. Zhu, Y. Cao, Y. Zhang, D. Feng, Y. Zhang, and M. Chen, "Efficient video fire detection exploiting motion- flicker-based dynamic features and deep static features," IEEE Access, vol. 8, pp. 81904–81917, 2020.

[2]. M. Shahid, I.-F. Chien, W. Sarapugdi, L. Miao, and K.-L. Hua, "Deep spatial-temporal networks for flame detection," Multimedia Tools Appl., vol. 80, nos. 28–29, pp. 35297–35318, Nov. 2021.

[3]. D. Zhang, H. Xiao, J. Wen, and Y. Xu, "Real-time fire detection method with multi-feature fusion on YOLOv5," Pattern Recognit. Artif. Intell., vol. 35, no. 6, pp. 548–561, 2022.

[4]. J. Yuan, L.Wang, P. Wu, C. Gao, and L. Sun, "Detection of wildfires along transmission lines using deep time and space features," Pattern Recognit. Image Anal., vol. 28, no. 4, pp. 805– 812, Oct. 2018.

[5]. Z. Wang, D. Wei, and X. Hu, "Research on two-stage flame detection algorithm based on fire feature and machine learning" in Proc. Int. Conf. Robot., Intell. Control Artif. Intell., Sep. 2019, pp. 574–578.

[6]. J. Ryu and D. Kwak, "Flame detection using appearance-based preprocessing and convolutional neural network," Appl. Sci., vol. 11, no. 11, p. 5138, May 2021.

[7]. O. Barnich and M. Van Droogenbroeck, "ViBe: A universal background subtraction algorithm for video sequences," IEEE Trans. Image Process., vol. 20, no. 6, pp. 1709– 1724, Jun. 2011.

[8]. A. Shamsoshoara, F. Afghah, A. Razi, L. Zheng, P. Z. Fulé, and E. Blasch, "Aerial imagery pile burn detection using deep learning: The FLAME dataset," Comput. Netw., vol. 193, Jul. 2021, Art. no. 108001.

[9]. Y. H. Habiboğlu, O. Günay, and A. E. Çetin, "Covariance matrix-based fire and flame detection method in the video," Mach. Vis. Appl., vol. 23, no. 6, pp. 1103– 1113, 2012.

[10]. S. H. Oh, S. W. Ghyme, S. K. Jung, and G. W. Kim, "Early wildfire detection using convolutional neural network," in Proc. Int. Workshop Frontiers Comput. Vis. Singapore: Springer, 2020, pp. 18–30.