Enhancing Fire Detection with YOLOv10: Advanced Techniques for Flame and Smoke Recognition

Author 1: B. Bhanu charan

Email: bhanucharan1810@gmail.com

Author 2: A. Praveen Reddy

Email: praveenreddyannem0621@gmail.com

Author 3: A. Abhika Rani

Email: 22wj1a0512@gmail.com

Guide: Mr. V. Devasekhar

Organization: Guru Nanak Institutions, India

Keywords: Fire Detection, YOLOv10, Flame and Smoke Recognition

ABSTRACT

This project introduces an advanced method for detecting flames and smoke using the YOLOv10 algorithm, aimed at improving fire detection systems for enhanced safety and prevention. The proposed system incorporates significant advancements to address challenges such as cluttered backgrounds, low visibility, varying fire intensities, and overlapping objects. By leveraging an enhanced feature extraction mechanism, the system captures finer details of flames and smoke. It also includes a sophisticated attention mechanism to prioritize critical fire-related areas in an image while suppressing irrelevant background information. To further enhance detection accuracy, the system uses an improved bounding box regression method, ensuring precise localization of flames and smoke, even in dense or challenging environments. These improvements make the system more robust, adaptable, and capable of handling high-resolution images and videos efficiently. Experimental results demonstrate that the YOLOv10-based system significantly outperforms previous methods, such as YOLOv5s, in terms of precision, recall, and mean average precision (mAP). Additionally, the system processes data in real-time, making it suitable for large-scale applications in industries, public spaces, and residential areas. By ensuring early and reliable detection of flames and smoke, the proposed system offers a practical solution for improving fire safety and prevention measures.

CHAPTER-1

INTRODUCTION

1.Introduction :

The growing need for effective fire detection systems has become increasingly important as fires pose significant threats to both human life and property. Traditional fire detection systems, such as smoke alarms and heat detectors, have limitations in detecting flames and smoke in complex or large-scale environments, especially under varying visibility and fire intensity conditions. This project introduces an advanced fire detection system based on the YOLOv10 (You Only Look Once) algorithm, which leverages state-of-the-art deep learning techniques to enhance the detection and localization of flames and smoke in real-time.

YOLOv10 has been chosen as the backbone for this system due to its ability to detect multiple objects within an image simultaneously, with high accuracy and speed, making it ideal for dynamic, real-world applications. The system uses enhanced feature extraction, an attention mechanism, and an improved bounding box regression method to address challenges such as cluttered backgrounds, overlapping objects, and varying fire conditions. These improvements ensure that the system can reliably detect flames and smoke even in complex, low-visibility, or densely packed environments. The proposed system not only provides accurate fire detection but also offers real-time processing, which is essential for immediate alerts in case of a fire outbreak. With applications spanning industrial plants, public spaces, residential areas, and even large-scale forests, this system aims to provide enhanced fire safety and prevention. The project's primary goal is to develop a robust, scalable, and efficient fire detection solution that can significantly reduce the risk of fire-related damages and save lives by enabling quicker responses to fire incidents. the proposed system also incorporates advanced machine learning techniques such as transfer learning and hyperparameter optimization to further enhance its accuracy and efficiency. By fine-tuning a pre-trained YOLOv10 model, the system can adapt to the specific challenges of flame and smoke detection, even with a relatively smaller dataset. The use of data augmentation techniques also ensures that the model remains robust under various environmental conditions, such as changing lighting, weather, or backgrounds. These enhancements not only improve detection performance but also make the system highly adaptable and scalable, allowing it to be deployed in diverse settings ranging from small residential areas to large industrial complexes. Ultimately, the goal of this project is to create a comprehensive, adaptable, and reliable fire detection system that provides both accuracy and speed, ensuring that fire incidents are detected early and mitigating the risks associated with late fire detection.

1.2 SCOPE OF THE PROJECT

The scope of this project revolves around enhancing the accuracy, reliability, and efficiency of flame and smoke detection through the YOLOv10 algorithm. The system is designed for real-time processing, enabling the early detection of fire hazards in environments such as industrial facilities, residential areas, public spaces, and forests. By addressing challenges such as cluttered backgrounds, low visibility, varying fire intensities, and overlapping objects, the proposed system ensures high detection performance even in complex conditions. The integration of advanced feature extraction and attention mechanisms allows the system to focus on critical fire-related areas while minimizing background noise. Additionally, the system's improved bounding box regression method ensures precise localization of flames and smoke, even in dense or challenging environments. This makes the system adaptable for large-scale applications, including smart city infrastructure and industrial safety. The scope of this project extends to improving fire safety measures by providing early, reliable detection, enabling quicker responses to prevent fire spread, reduce damage, and save lives. Moreover, the system can be further integrated with other safety systems like automated fire suppression or emergency response mechanisms. Ultimately, the project aims to offer a robust, scalable solution that enhances fire detection capabilities and contributes to enhanced fire safety across various sectors.

1.3 OBJECTIVE

The objective of this project is to develop an advanced fire detection system that utilizes the YOLOv10 algorithm to accurately detect and localize flames and smoke in real-time. The system aims to enhance detection performance by incorporating advanced techniques such as improved feature extraction, attention mechanisms, and enhanced bounding box regression methods to address challenges like cluttered backgrounds, varying fire intensities, and overlapping objects. Additionally, the project focuses on making the system robust enough to function effectively in complex environments with low visibility or dense surroundings. Through the use of transfer learning and data augmentation, the system will be optimized for real-time applications, ensuring fast and reliable detection. The goal is to create a scalable and adaptable solution that can be deployed in diverse settings, such as industrial plants, residential areas, public spaces, and wildfire management. Ultimately, the project seeks to improve fire safety by providing early and accurate fire detection, enabling quicker responses to prevent fire-related damages and saving lives.

1.1 EXISTING SYSTEM:

The existing system in this project utilizes the YOLOv5s algorithm for detecting flames and smoke in images and videos. YOLOv5s is a widely used and efficient object detection model known for its high speed and accuracy in real-time applications. In the context of fire detection, YOLOv5s is capable of detecting various objects in the scene, including flames and smoke, and localizing them within the image by drawing bounding boxes around them.

The existing system leverages YOLOv5s' ability to detect and classify objects in a single forward pass, making it suitable for applications that require real-time fire detection, such as surveillance systems in industries, residential areas, and public spaces. It uses a standard feature extraction process, relying on the model's pre-trained weights to perform object detection. YOLOv5s is capable of handling complex environments, but it may struggle with certain challenges, such as cluttered backgrounds, varying fire intensities, and low visibility, which can reduce its accuracy in detecting flames and smoke.

In terms of performance, YOLOv5s is relatively fast and efficient, but the system could face limitations when dealing with dense or challenging environments where objects overlap or when the fire or smoke is not clearly visible. Despite these challenges, YOLOv5s provides a solid foundation for fire detection applications, offering a good balance of speed and accuracy. However, the need for more advanced techniques to improve detection performance in complex real-world conditions led to the proposal of YOLOv10 as an enhancement to address these limitations and improve overall detection accuracy and reliability.

1.4.1 EXISTINGSYSTEM DISADVANTAGES:

- Lower Detection Accuracy: YOLOv5s may miss smaller or partially obscured flames and smoke, leading to false negatives, whereas YOLOv10 provides higher precision in complex environments.
- Limited Feature Extraction: YOLOv5s struggles with capturing fine details of flames and smoke, while YOLOv10 enhances feature extraction for better detection in cluttered scenarios.
- Inability to Prioritize Critical Features: YOLOv5s processes the entire image equally, while YOLOv10 uses an attention mechanism to focus on fire-related areas, improving accuracy.
- Bounding Box Regression Limitations: YOLOv5s may provide less accurate bounding box localization, whereas YOLOv10's improved method ensures better detection, even in dense environments.
- Performance in Dense Environments: YOLOv5s may fail in detecting flames and smoke in crowded or obstructed scenes, while YOLOv10 excels in these situations.
- Real-Time Detection Limitations: YOLOv5s may struggle with high-resolution images in real-time, whereas YOLOv10 is optimized for faster processing and real-time detection.

1.5 LITERATURE SURVEY

Title: Enhancing geometric factors in model learning and inference for object detection and instance segmentation

Author: Z. Zheng, P. Wang, D. Ren, W. Liu, R. Ye, Q. Hu, and W. Zuo,

Year: 2022.

Description: The paper titled "Enhancing geometric factors in model learning and inference for object detection and instance segmentation" by Z. Zheng, P. Wang, D. Ren, W. Liu, R. Ye, Q. Hu, and W. Zuo, published in the IEEE Transactions on Cybernetics (Aug. 2022), explores the incorporation of geometric factors into the learning and inference processes of object detection and instance segmentation models. The authors address the challenge of improving the performance of object detection and segmentation models by considering geometric factors such as the spatial relationships and shapes of objects. Traditional object detection models, while successful, often fail to account for the rich geometric information inherent in the objects themselves, which can lead to inaccuracies in localizing or segmenting objects, especially in complex or cluttered scenes. To overcome this, the paper proposes new methods for incorporating geometric factors into both model learning and inference. The authors argue that by explicitly modeling and leveraging geometric cues such as object shape, orientation, and relative positioning, the detection and segmentation models can achieve better accuracy, particularly in scenarios where objects exhibit complex or unusual geometries. The paper demonstrates how these geometric factors enhance the feature extraction process and contribute to more precise bounding box predictions and better segmentation results. In the context of instance segmentation, where it is crucial to distinguish between different object instances even when they are close to or overlapping with each other, geometric information plays a significant role in ensuring that boundaries are correctly identified. This methodology can be applied to various real-world scenarios, including autonomous driving, robotics, and video surveillance, where accurate object detection and segmentation are critical.

Title: CSPNet: A new backbone that can enhance learning capability of CNN

Author: C.-Y. Wang, H.-Y. Mark Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh

Year: 2020.

Description: The paper titled "CSPNet: A new backbone that can enhance the learning capability of CNN" by C.-Y. Wang, H.-Y. Mark Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, published in the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), June 2020, introduces a novel convolutional neural network (CNN) backbone known as CSPNet (Cross-Stage Partial Network). The main contribution of this paper is the introduction of CSPNet, which enhances the learning capability of CNNs, specifically improving their efficiency in deep network architectures. Traditionally, as CNN architectures become deeper, the models face challenges in terms of gradient flow, optimization, and learning efficiency. These problems can lead to slower convergence during training and sometimes poorer overall performance on tasks like image classification, detection, and segmentation. CSPNet aims to address these issues by introducing a cross-stage partial network strategy. Instead of utilizing a single stage for feature extraction, it divides the feature maps into multiple groups, with some being passed through and others being processed in a partial manner before merging again. This helps the network focus on important features while avoiding unnecessary computations that may contribute to overfitting or inefficiencies. By splitting the computation across different stages, CSPNet allows for more efficient training and better gradient propagation, which enhances learning in deeper networks. In terms of model performance, CSPNet outperforms traditional CNN architectures by reducing the computational complexity while maintaining or improving accuracy. The model achieves a balance between efficiency and performance, making it particularly suited for real-time applications or scenarios where computational resources are limited.

Title: Coordinate attention for efficient mobile network design

Author: Q. Hou, D. Zhou, and J. Feng

Year: 2021.

Description: The paper titled "Coordinate Attention for Efficient Mobile Network Design" by Q. Hou, D. Zhou, and J. Feng, presented at the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2021, introduces a novel attention mechanism known as Coordinate Attention (CA) to improve the efficiency and performance of mobile networks, particularly in the context of mobile device usage. The main objective of this work is to design an efficient attention mechanism for mobile networks that can strike a balance between computational efficiency and performance. Mobile networks often face resource constraints such as limited computation power, memory, and bandwidth, making it challenging to deploy large-scale models on such devices. Therefore, there is a need for lightweight and efficient models that can perform well on mobile devices while keeping resource consumption in check. The Coordinate Attention mechanism proposed in this paper targets spatial and channel information by leveraging the coordinate system of feature maps. Traditional attention mechanisms, such as the Channel Attention (CA) or Spatial Attention (SA), focus on either spatial or channel-wise attention but often miss capturing the dependencies across both dimensions efficiently. The Coordinate Attention mechanism integrates both spatial and channel dependencies by incorporating the coordinate system of feature maps into the attention process. It reduces computational costs by using an efficient strategy for computing attention, which is crucial for mobile networks. Coordinate Attention works by splitting the input feature maps into two key components: one that handles spatial information and another that captures channel-wise dependencies. These components are then fused through an efficient coordinate-based attention mechanism, allowing the model to focus on the most relevant features while remaining computationally efficient.

Title: Multi-temporal dependency handling in video smoke recognition: A holistic approach spanning spatial, short-term, and long-term perspectives

Author: F. Yang, Q. Xue, Y. Cao, X. Li, W. Zhang, and G. Li

Year: 2024

Description: The paper titled "Multi-temporal Dependency Handling in Video Smoke Recognition: A Holistic Approach Spanning Spatial, Short-term, and Long-term Perspectives" by F. Yang, Q. Xue, Y. Cao, X. Li, W. Zhang, and G. Li, published in Expert Systems with Applications (July 2024), presents an innovative approach for video-based smoke recognition by addressing the challenge of handling multi-temporal dependencies. The paper focuses on improving the recognition of smoke in video sequences, where smoke can exhibit complex and dynamic behaviors over time. Traditional methods for smoke recognition often struggle to handle the temporal dependencies present in videos, particularly when the smoke is in motion, appears intermittently, or is influenced by various environmental factors. This research proposes a holistic approach that spans spatial, short-term, and long-term temporal dependencies to enhance the accuracy and robustness of smoke detection in video streams.

The core idea of the approach is to integrate spatial information (the appearance of smoke at different locations in the video frame), short-term temporal dependencies (the immediate transitions from one frame to the next), and long-term temporal dependencies (the overall progression of smoke over extended periods). By combining these three perspectives, the model is better equipped to understand the evolving characteristics of smoke in videos, leading to more reliable and precise detection.

To address these dependencies, the authors introduce a novel framework that effectively combines convolutional neural networks (CNNs) for spatial feature extraction and recurrent neural networks (RNNs) for temporal modeling. The short-term dependencies are captured using RNNs that process adjacent video frames, while long-term dependencies are managed by integrating memory-based mechanisms to maintain context over extended video segments. This multi-level approach allows the model to capture both the local motion of smoke and its larger-scale temporal changes over time.

Title: Improved fire detection by YOLOv8 and YOLOv5 to enhance fire safety,

Author: M. N. Uddin, M. Sakibul Islam Sakib, S. Nawer, and R. T. Mohona

Year: 2023**.**

Description: The paper titled "Improved Fire Detection by YOLOv8 and YOLOv5 to Enhance Fire Safety" by M. N. Uddin, M. Sakibul Islam Sakib, S. Nawer, and R. T. Mohona, presented at the 26th International Conference on Computer and Information Technology (ICCIT) in December 2023, explores the enhancement of fire detection systems through the use of advanced object detection algorithms, specifically YOLOv8 and YOLOv5.
In this work, the authors aim to improve the accuracy and efficiency of fire detection systems, which play a crucial role in fire safety applications, especially in industrial, commercial, and residential environments. The detection of fire in real-time is vital for initiating prompt emergency responses and minimizing potential damages. The paper compares the performance of two popular versions of the YOLO (You Only Look Once) algorithm—YOLOv8 and YOLOv5—focusing on how they can be applied to the detection of fire in images and videos.

The paper highlights that fire detection often faces challenges such as false positives, difficulty in distinguishing fire from other similar-looking phenomena (like smoke or light reflections), and varying fire intensities across different environments. YOLO-based models, known for their real-time object detection capabilities, are applied in this study to address these challenges. YOLOv5, which is widely used for object detection tasks, provides a robust baseline for comparison. YOLOv8, being a more recent iteration, includes improved feature extraction, better handling of small objects, and enhanced performance in detecting objects in diverse and cluttered backgrounds, making it particularly suitable for fire detection tasks.

1.6 PROPOSED SYSTEM

The proposed system introduces the YOLOv10 algorithm for enhanced flame and smoke detection in images and videos. YOLOv10 is an advanced version of the YOLO series, designed to improve upon its predecessors by incorporating several innovative techniques that address the limitations of previous models like YOLOv5s. This system is specifically designed to handle complex environments with cluttered backgrounds, varying fire intensities, and low visibility, which are common challenges in fire detection scenarios. One of the key improvements in the proposed system is the integration of an enhanced feature extraction mechanism that captures finer details of flames and smoke, improving detection accuracy. Additionally, a sophisticated attention mechanism is employed, which prioritizes critical areas of an image that are likely to contain flames or smoke, while suppressing irrelevant background information. This mechanism ensures that the model focuses on the most important features, improving both speed and precision.

Another notable advancement is the use of an improved bounding box regression method, which ensures more precise localization of flames and smoke, even in dense or challenging environments. The system can accurately identify and localize fire-related objects in images, even when multiple objects overlap or when flames and smoke are partially obscured. The proposed YOLOv10-based system is also designed for real-time detection, which is essential for fire safety applications. By processing high-resolution images and videos quickly and efficiently, the system can provide immediate alerts to users in various settings, such as industrial plants, public spaces, or residential areas. The system is robust and adaptable, capable of working in diverse conditions, including varying light levels, environmental factors, and different fire intensities.

Overall, the proposed system outperforms the existing YOLOv5s algorithm in terms of precision, recall, and mean average precision (mAP), making it a significant improvement for real-time fire detection. It represents a more reliable and adaptable solution for enhancing fire safety and prevention measures across multiple environments.

1.6.1 PROPOSED SYSTEM ADVANTAGES:

- Higher Detection Accuracy: YOLOv10 offers improved accuracy, effectively detecting flames and smoke, even in challenging environments with low visibility or partial obstructions.
- Enhanced Feature Extraction: YOLOv10 captures finer details of flames and smoke, improving its ability to detect fire-related objects in cluttered and dynamic settings.
- Advanced Attention Mechanism: YOLOv10 prioritizes fire-related regions, ensuring faster and more accurate detection by suppressing irrelevant background information.
- Improved Bounding Box Localization: YOLOv10's enhanced bounding box regression method ensures more precise localization of flames and smoke, even in dense or overlapping environments.
- Better Performance in Dense Environments: YOLOv10 excels in detecting flames and smoke in complex, object-rich environments, where YOLOv5s may struggle.
- Optimized Real-Time Detection: YOLOv10 processes high-resolution images more efficiently, ensuring faster real-time detection suitable for large-scale applications in fire safety.

CHAPTER 2

PROJECT DESCRIPTION

2.1 GENERAL:

This project focuses on developing an advanced fire detection system using the YOLOv10 (You Only Look Once) algorithm to accurately detect flames and smoke in real-time. The primary aim is to address the challenges faced by traditional fire detection systems, such as their inability to handle complex environments, low visibility, and varying fire intensities. The YOLOv10 algorithm has been selected due to its capability to detect multiple objects in images or video streams simultaneously, with high accuracy and speed, making it well-suited for real-time fire detection.

The system is designed to identify and localize flames and smoke by incorporating advanced techniques such as enhanced feature extraction, an attention mechanism to focus on critical fire-related areas, and an improved bounding box regression method to ensure precise localization. The integration of transfer learning allows the model to leverage pre-trained knowledge and adapt it for fire detection, even with a smaller dataset, while data augmentation techniques improve the system's robustness under various environmental conditions.

The system's primary goal is to provide early, reliable detection of fires, reducing the risk of damage to property and loss of life. It is intended for use in diverse environments, including industrial plants, residential areas, public spaces, and even large-scale outdoor areas like forests. With real-time processing capabilities, the system ensures quick alerts, allowing for rapid response to fire incidents. The project aims to create a scalable, efficient, and adaptive fire detection solution that enhances fire safety and prevention measures in various contexts.

2.2 METHODOLOGIES

2.2.1MODULES NAME:

Modules Name:

- **Object Detection Engine**
- **Model Adaptation through Pre-trained Networks**
- **Performance Tuning Module**
- **Data Expansion and Augmentation**
- **Model Training and Validation**
- **Real-time Fire & Smoke Detection and Localization**
- **Performance Evaluation and Monitoring:**

2.2.2 MODULES EXPLANATION:

1) Object Detection Engine:

This module forms the backbone of the system and is responsible for detecting and locating flames and smoke within images and video streams. Using the advanced YOLOv10 architecture, this engine classifies the objects and determines their exact position in the image. The ability to detect multiple objects simultaneously, combined with its high speed, makes this module ideal for real-time fire detection applications, ensuring that hazardous situations are identified promptly.

2) Model Adaptation through Pre-trained Networks:

This module utilizes transfer learning to enhance the model's performance by leveraging pre-trained networks. The system starts with a pre-trained YOLOv10 model, which has been trained on large, general datasets. It then fine-tunes the model using a smaller, domain-specific fire dataset, enabling the system to learn fire-specific features efficiently. This approach minimizes the need for large amounts of labeled data and accelerates the training process.

3) Performance Tuning Module:

This module is focused on fine-tuning the model's hyperparameters for optimal performance. Key parameters such as learning rate, batch size, and number of epochs are adjusted to ensure that the model converges quickly and accurately. Optimizing these settings improves the detection precision and ensures that the system can work effectively in diverse conditions, adapting to different types of fire events.

4) Data Expansion and Augmentation:

The data augmentation module is used to artificially increase the size of the training dataset by applying various transformations like rotation, flipping, scaling, and adjustments in lighting or contrast. This process helps the system to become more robust, enabling it to generalize well to new and unseen conditions. By simulating different real-world scenarios, this module ensures that the fire detection system remains effective in varied environments.

5) Model Training and Validation:

This module is responsible for the process of training the fire detection model on labeled datasets and validating its performance on separate test data. The model learns to recognize and classify fire-related features based on the training data. During the validation phase, the model's ability to generalize to new data is assessed, helping ensure that it is not overfitting and can detect fires effectively in real-world conditions.

6) Real-time Fire & Smoke Detection and Localization:

Once trained, this module allows the system to detect flames and smoke in real time. It processes incoming video or image streams and immediately identifies the presence of fire-related hazards. Additionally, it outputs the location of these hazards within the image or video frame, helping emergency response teams or automated systems to react swiftly and accurately in the event of a fire.

7) Performance Evaluation and Monitoring:

The evaluation module focuses on assessing the effectiveness of the fire detection system using standard metrics such as accuracy, precision, recall, and F1-score. It ensures that the system is performing as expected and provides insights into areas for improvement. This module also tracks the system's ability to handle large-scale data in real-time, making sure that it remains efficient and scalable in production environments.

2.3 TECHNIQUE USED OR ALGORITHM USED

2.3.1 EXISTING TECHNIQUE: -

- **YOLOV5s**

The existing algorithm in this project is YOLOv5s, which is a version of the YOLO (You Only Look Once) model designed for real-time object detection. YOLOv5s is known for its ability to perform both classification and localization in a single forward pass, making it highly efficient and fast. It divides an image into a grid and predicts bounding boxes for each object present, along with the corresponding class probabilities for each grid cell. The model uses convolutional layers to extract features at different levels of abstraction, which are then passed through the detection head for object localization and classification. YOLOv5s is optimized for speed, making it a popular choice for real-time applications such as surveillance and industrial monitoring.

However, while YOLOv5s is effective in general object detection tasks, it has limitations when applied to more specialized and complex problems such as fire detection. In cluttered environments or scenarios with low visibility, such as during smoke detection or in images with varying fire intensities, YOLOv5s may struggle to distinguish subtle features, like faint flames or light smoke. This is due to its architecture, which is more focused on processing speed than on high-resolution or detailed feature extraction. Additionally, YOLOv5s lacks an attention mechanism, meaning it cannot emphasize critical regions in an image, leading to lower detection accuracy in situations where the fire or smoke is small, distant, or partially obscured. Another limitation is its reliance on anchor boxes, which may not always capture the right aspect ratios for irregularly shaped fire and smoke objects, leading to errors in bounding box predictions. These factors limit YOLOv5s' performance in detecting flames and smoke under challenging real-world conditions, necessitating the adoption of more advanced methods such as YOLOv10, which can address these challenges more effectively.

2.3.2 PROPOSED TECHNIQUE USED OR ALGORITHM USED:

- **YOLOv10:**

The proposed algorithm in this project is YOLOv10, an advanced version of the YOLO (You Only Look Once) model that enhances the performance of object detection tasks, particularly in fire and smoke detection. YOLOv10 introduces several improvements over previous versions, particularly YOLOv5s, which make it more effective for specialized tasks like detecting flames and smoke in challenging environments. One of the key improvements is the inclusion of an enhanced feature extraction mechanism that captures finer details of flames and smoke, which are often difficult to detect due to their small size, subtle appearance, or varying intensity. This enables the model to distinguish between fire and smoke, even in cluttered or low-visibility conditions.

Another significant enhancement in YOLOv10 is the integration of a sophisticated attention mechanism. This mechanism allows the model to focus on critical areas of an image that are more likely to contain flames or smoke, while suppressing irrelevant background information. This selective attention helps to reduce the impact of noisy or cluttered backgrounds and improves the accuracy of detection, particularly in environments with overlapping objects or other distractions. Additionally, YOLOv10 uses an improved bounding box regression method, which ensures more precise localization of flames and smoke, even when they appear in dense or difficult-to-interpret images.

Furthermore, YOLOv10 is designed to handle high-resolution images and videos more efficiently, making it well-suited for real-time detection in large-scale applications, such as fire safety monitoring in industrial sites, residential areas, or public spaces. With these advancements, YOLOv10 is more robust and adaptable to varied environmental conditions compared to its predecessors. It also outperforms older models, like YOLOv5s, in terms of precision, recall, and mean average precision (mAP), making it an ideal choice for accurate, real-time fire and smoke detection.

CHAPTER 3

REQUIREMENTS ENGINEERING

3.1 GENERAL

We can see from the results that on each database, the error rates are very low due to the discriminatory power of features and the regression capabilities of classifiers. Comparing the highest accuracies (corresponding to the lowest error rates) to those of previous works, our results are very competitive.

3.2 HARDWARE REQUIREMENTS

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It should what the system do and not how it should be implemented.

- PROCESSOR : DUAL CORE 2 DUOS.
- RAM : 4GB DD RAM
- HARD DISK : 250 GB

3.3 SOFTWARE REQUIREMENTS

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the teams and tracking the team's progress throughout the development activity.

- Operating System : Windows 7/8/10
- Platform : Anaconda
- Programming Language : Python
- Front End : VS code

3.4 FUNCTIONAL REQUIREMENTS

A functional requirement defines a function of a software-system or its component. A function is described as a set of inputs, the behavior, Firstly, the system is the first that achieves the standard notion of semantic security for data confidentiality in attribute-based deduplication systems by resorting to the hybrid cloud architecture.

3.5 NON-FUNCTIONAL REQUIREMENTS

The major non-functional Requirements of the system are as follows

Usability

The system is designed with completely automated process hence there is no or less user intervention.

Reliability

The system is more reliable because of the qualities that are inherited from the chosen platform python. The code built by using python is more reliable.

Performance

This system is developing in the high level languages and using the advanced back-end technologies it will give response to the end user on client system with in very less time.

Supportability

The system is designed to be the cross platform supportable. The system is supported on a wide range of hardware and any software platform, which is built into the system.

Implementation

The system is implemented in web environment using Jupyter notebook software. The server is used as the intellignce server and windows 10 professional is used as the platform. Interface the user interface is based on Jupyter notebook provides server system.

CHAPTER 4

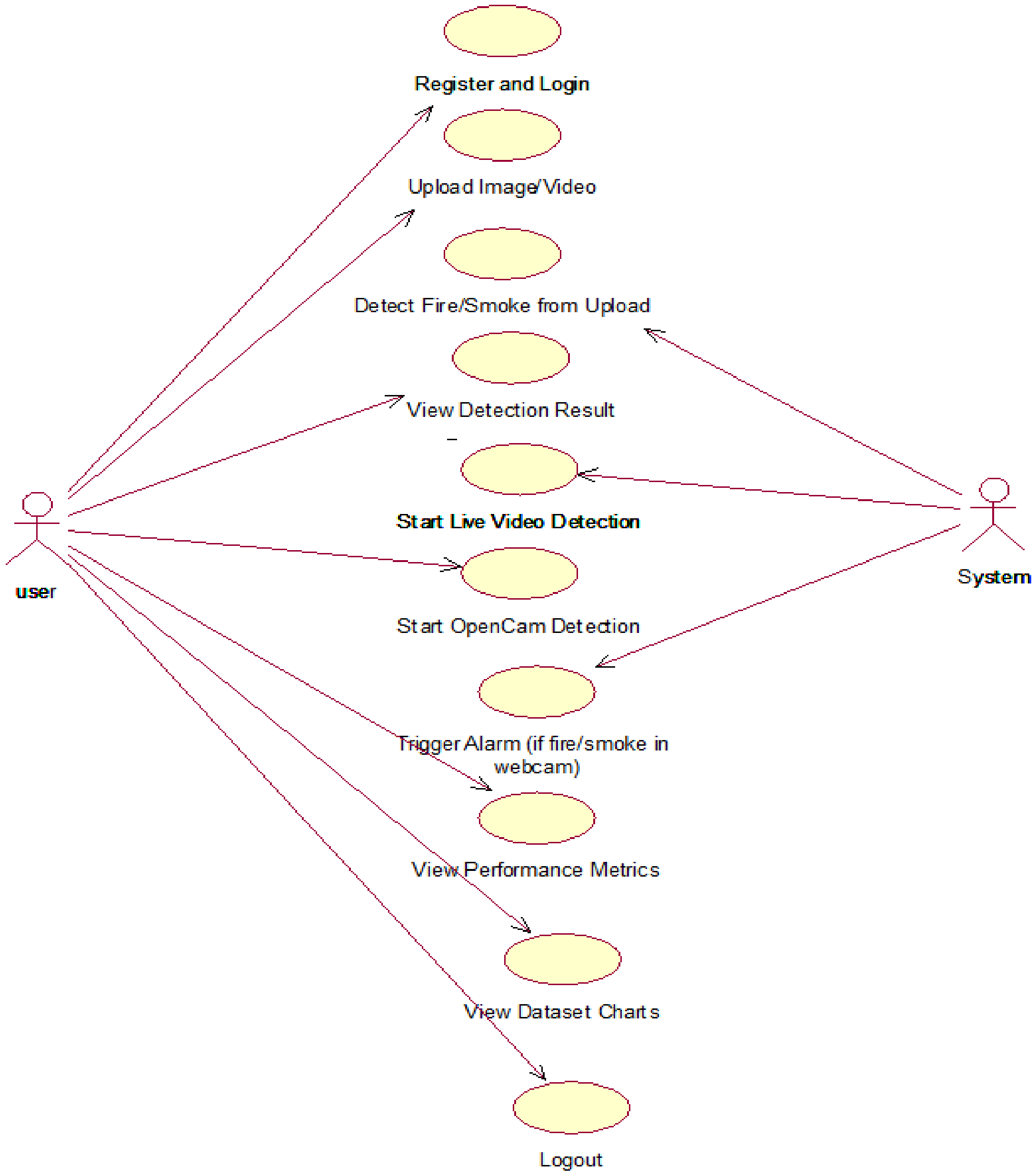
DESIGN ENGINEERING

4.1 GENERAL

Design Engineering deals with the various UML [Unified Modelling language] diagrams for the implementation of project. Design is a meaningful engineering representation of a thing that is to be built. Software design is a process through which the requirements are translated into representation of the software. Design is the place where quality is rendered in software engineering.

4.2 UML DIAGRAMS

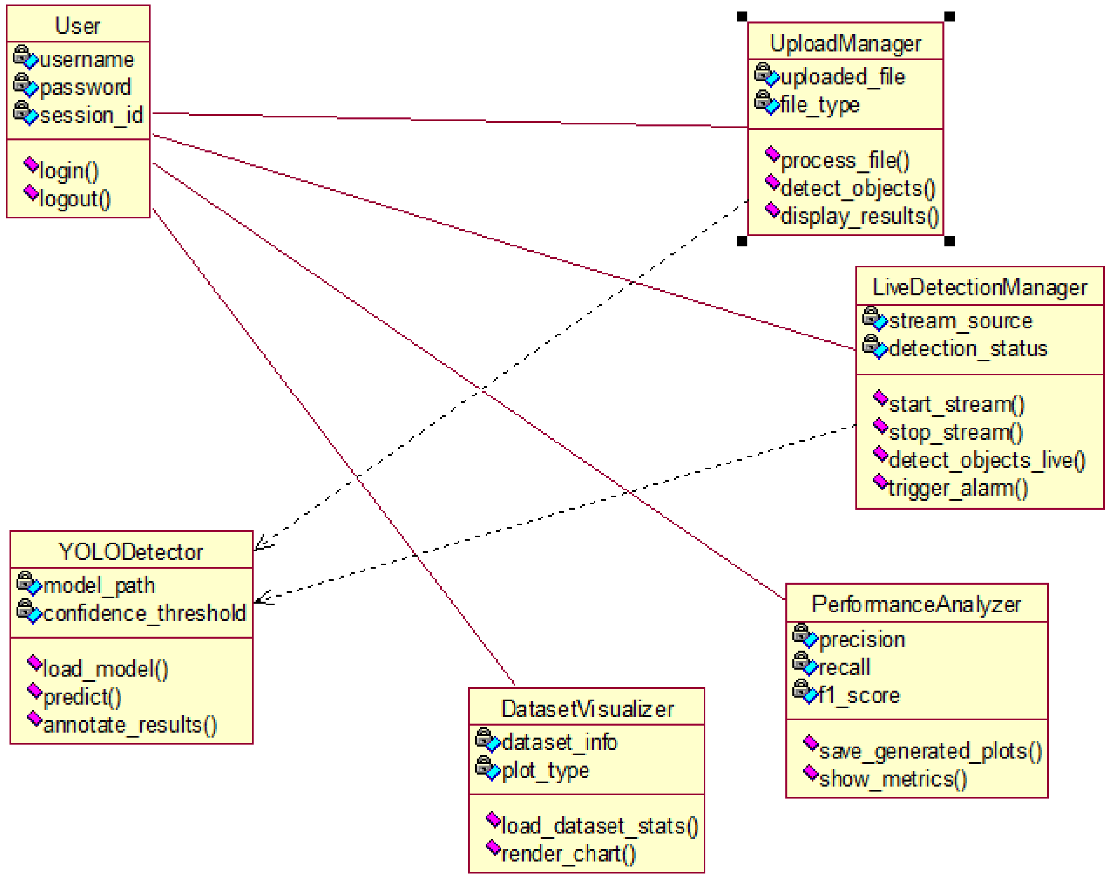
4.2.1 USE CASE DIAGRAM



EXPLANATION:

The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted. The above diagram consists of user as actor. Each will play a certain role to achieve the concept.

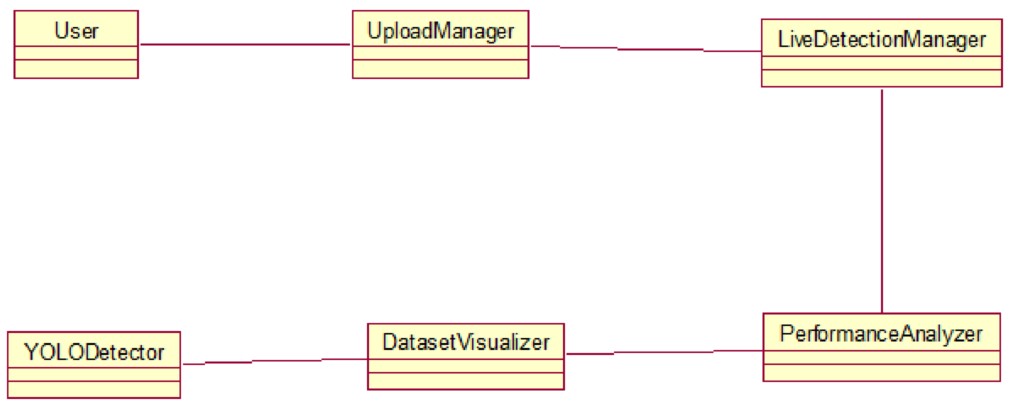
4.2.2 CLASS DIAGRAM



EXPLANATION

In this class diagram represents how the classes with attributes and methods are linked together to perform the verification with security. From the above diagram shown the various classes involved in our project.

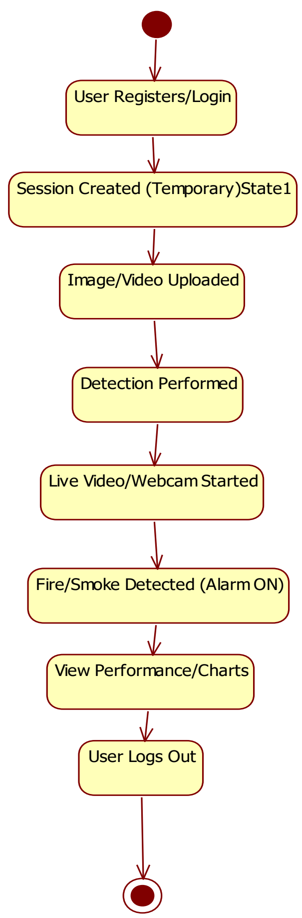
4.2.3 OBJECT DIAGRAM



EXPLANATION:

In the above digram tells about the flow of objects between the classes. It is a diagram that shows a complete or partial view of the structure of a modeled system. In this object diagram represents how the classes with attributes and methods are linked together to perform the verification with security.

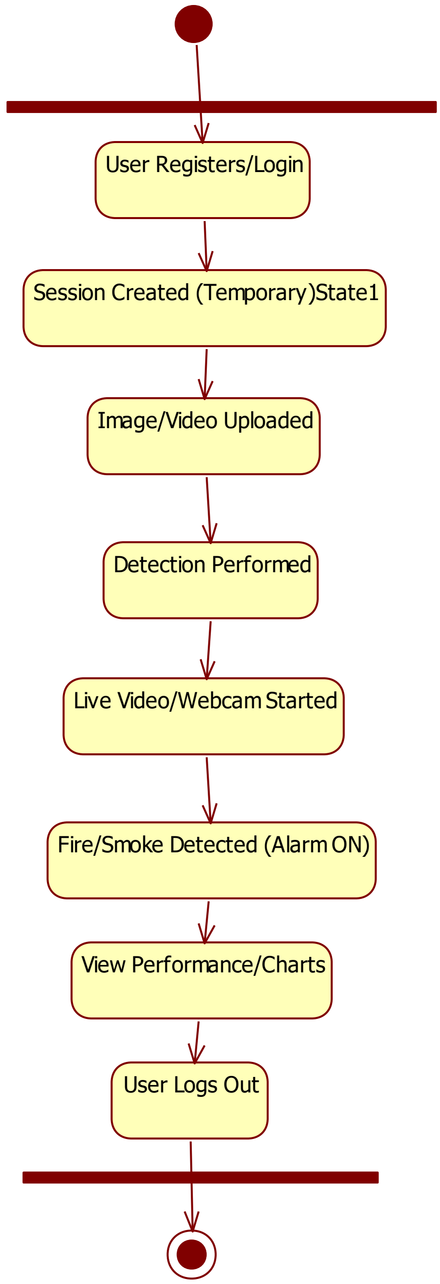
4.2.4 STATE DIAGRAM



EXPLANATION:

State diagram are a loosely defined diagram to show workflows of stepwise activities and actions, with support for choice, iteration and concurrency. State diagrams require that the system described is composed of a finite number of states; sometimes, this is indeed the case, while at other times this is a reasonable abstraction. Many forms of state diagrams exist, which differ slightly and have different semantics.

4.2.5 ACTIVITY DIAGRAM

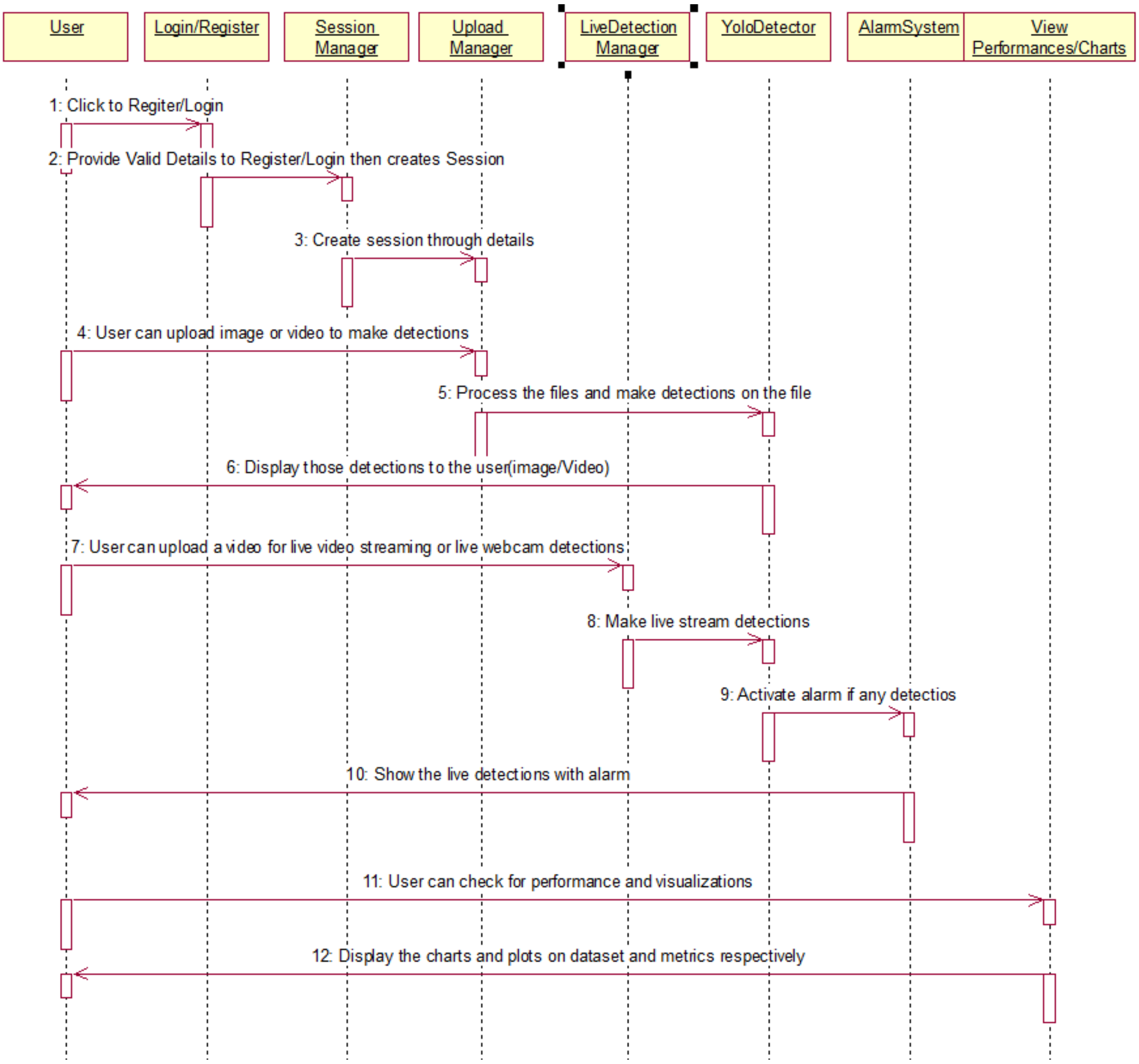


EXPLANATION:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 03 | March - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

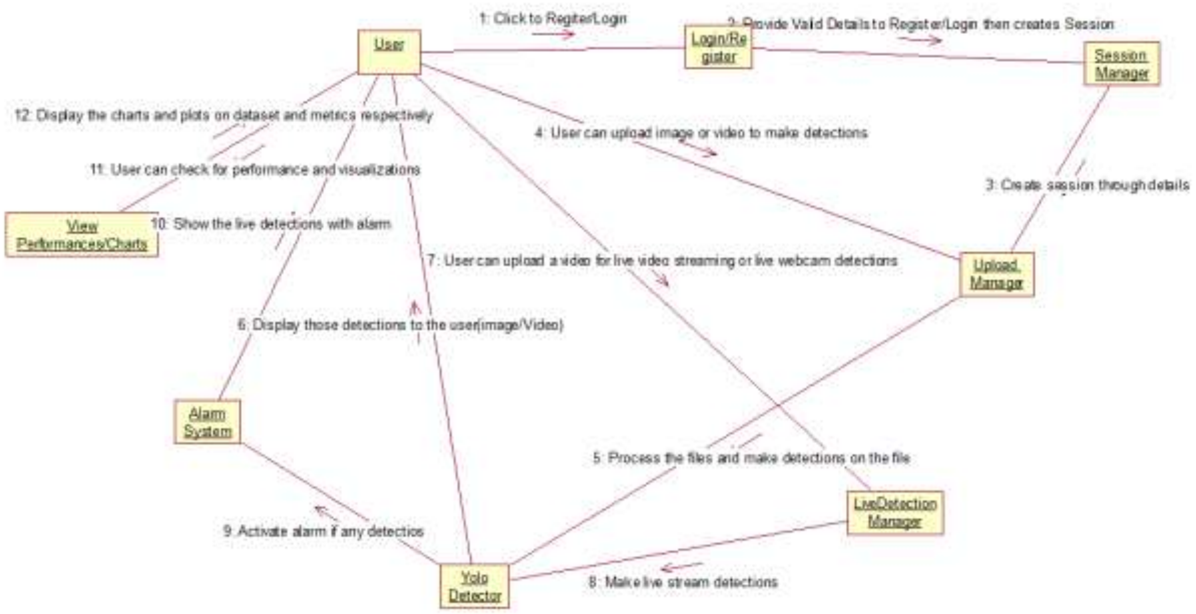
4.2.6 SEQUENCE DIAGRAM



EXPLANATION:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

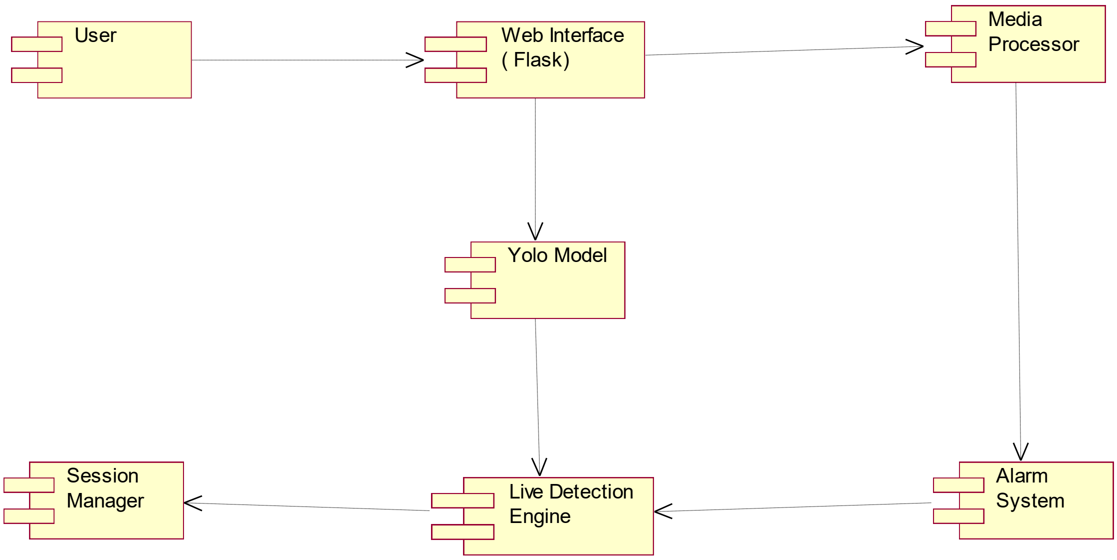
4.2.7 COLLABORATION DIAGRAM



EXPLANATION:

A collaboration diagram, also called a communication diagram or interaction diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). The concept is more than a decade old although it has been refined as modeling paradigms have evolved.

4.2.8 COMPONENT DIAGRAM



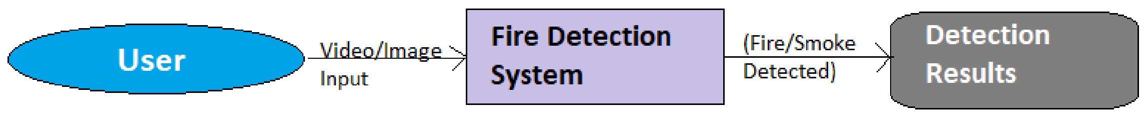
EXPLANATION

In the Unified Modeling Language, a component diagram depicts how components are wired together to form larger components and or software systems. They are used to illustrate the structure of arbitrarily complex systems. User gives main query and it converted into sub queries and sends through data dissemination to data aggregators. Results are to be showed to user by data aggregators. All boxes are components and arrow indicates dependencies.

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 03 | March - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

4.2.9 DATA FLOW DIAGRAM

Level 0



Level 1

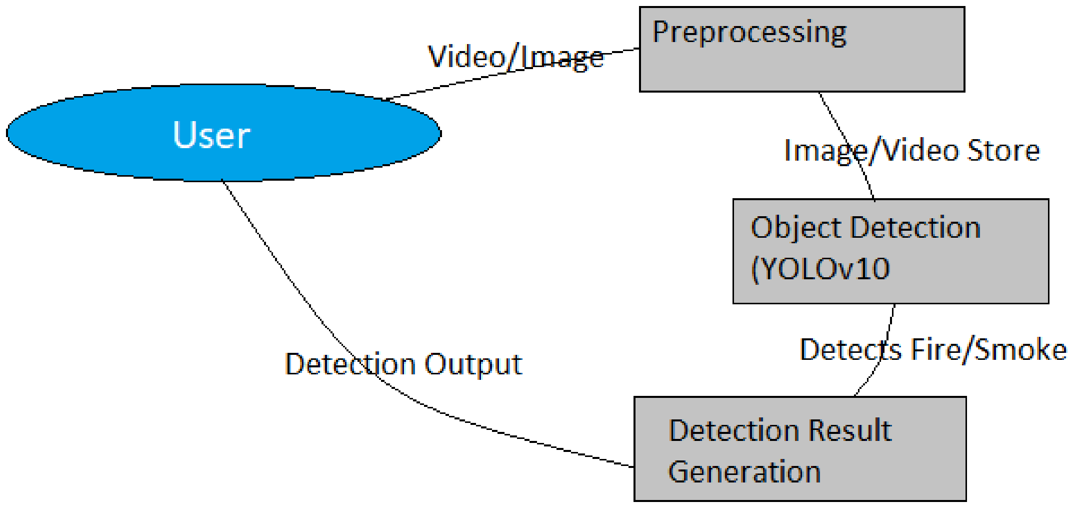


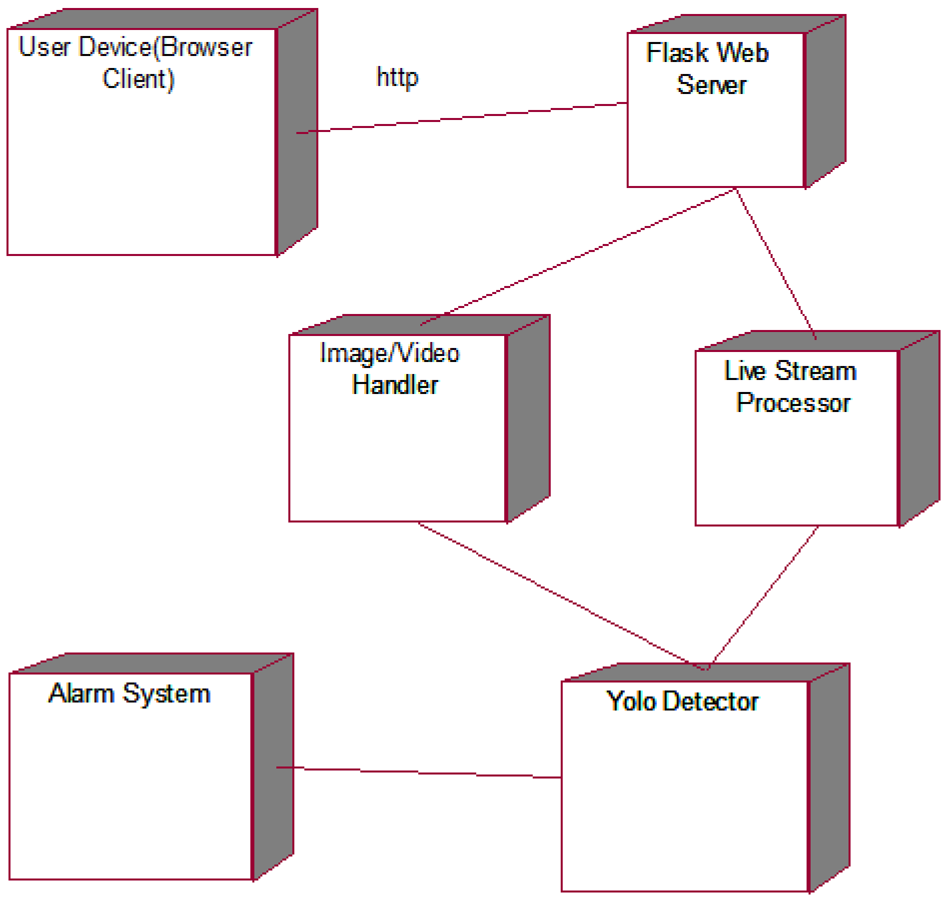
Fig 4.9: Data Flow Diagrams

EXPLANATION:

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. Often they are a preliminary step used to create an overview of the system which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).

A DFD shows what kinds of data will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of processes, or information about whether processes will operate in sequence or in parallel.

4.2.10 DEPLOYMENT DIAGRAM



EXPLANATION:

Deployment Diagram is a type of diagram that specifies the physical hardware on which the software system will execute. It also determines how the software is deployed on the underlying hardware. It maps software pieces of a system to the device that are going to execute it.

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 03 | March - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

SYSTEM ARCHITECTURE:

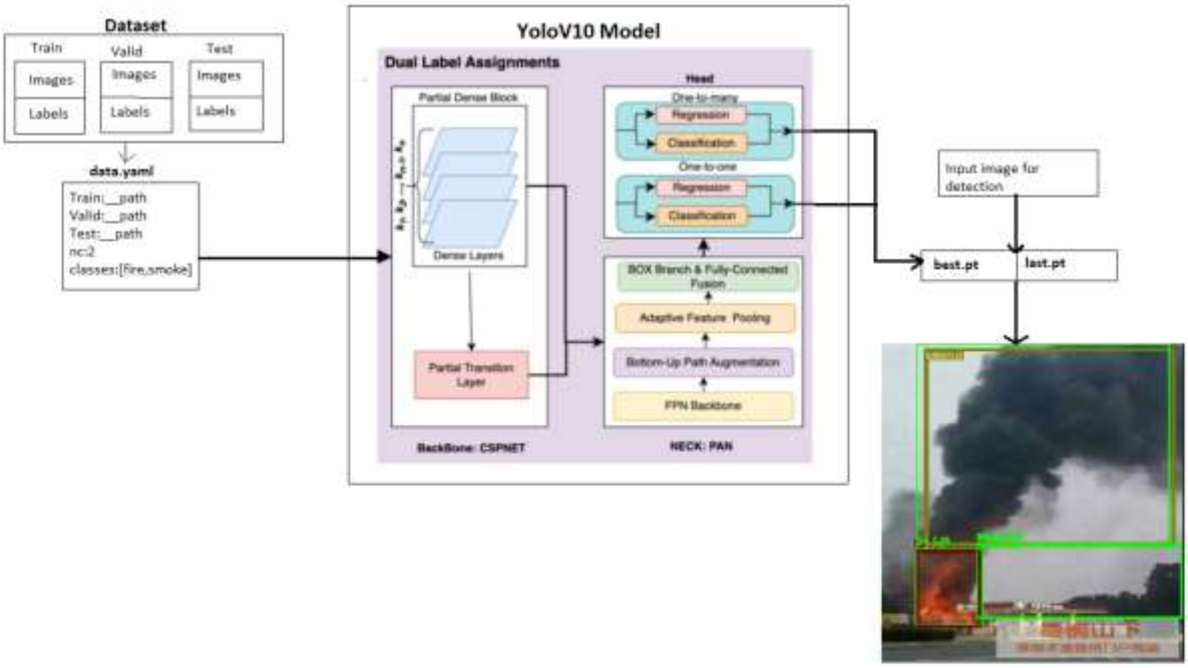


Fig 4.11: System Architecture

CHAPTER 5

DEVELOPMENT TOOLS

5.1 Python

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

5.2 History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

5.3 Importance of Python

- **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive** − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

- **Python is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language** − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

5.4 Features of Python

- **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** − Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.
- **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** − Python provides interfaces to all major commercial databases.
- **GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable** − Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below −

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- IT supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

5.5 Libraries used in python

- numpy - mainly useful for its N-dimensional array objects.
- pandas - Python data analysis library, including structures such as dataframes.
- matplotlib - 2D plotting library producing publication quality figures.
- scikit-learn - the machine learning algorithms used for data analysis and data mining tasks.

Figure : NumPy, Pandas, Matplotlib, Scikit-learn

CHAPTER 6

SOFTWARE TESTING

8.1 GENERAL

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

8.2 DEVELOPING METHODOLOGIES

The test process is initiated by developing a comprehensive plan to test the general functionality and special features on a variety of platform combinations. Strict quality control procedures are used. The process verifies that the application meets the requirements specified in the system requirements document and is bug free. The following are the considerations used to develop the framework from developing the testing methodologies.

8.3Types of Tests

8.3.1 Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program input produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

8.3.2 Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.
Invalid Input : identified classes of invalid input must be rejected.
Functions : identified functions must be exercised.
Output : identified classes of application outputs must be exercised.
Systems/Procedures: interfacing systems or procedures must be invoked.

8.3.3 System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

8.3.4 Performance Test

The Performance test ensures that the output be produced within the time limits,and the time taken by the system for compiling, giving response to the users and request being send to the system for to retrieve the results.

8.3.5 Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

8.3.6 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Acceptance testing for Data Synchronization:

- The Acknowledgements will be received by the Sender Node after the Packets are received by the Destination Node
- The Route add operation is done only when there is a Route request in need
- The Status of Nodes information is done automatically in the Cache Updation process

8.2.7 Build the test plan

Any project can be divided into units that can be further performed for detailed processing. Then a testing strategy for each of this unit is carried out. Unit testing helps to identity the possible bugs in the individual component, so the component that has bugs can be identified and can be rectified from errors.

CHAPTER 7

FUTURE ENHANCEMENT

9.1 FUTURE ENHANCEMENTS:

In the future, the fire detection system can be further enhanced in several ways to improve its accuracy, scalability, and adaptability. One potential enhancement could involve integrating deep learning models that specifically address the detection of other hazardous situations, such as gas leaks or chemical spills, to broaden the system's safety capabilities. By combining fire detection with multi-hazard recognition, the system could provide a more comprehensive safety solution for various environments, including industrial sites and urban areas. Additionally, incorporating edge computing could significantly reduce processing time and improve real-time performance. By processing data locally on edge devices, the system could provide instant fire alerts without relying on cloud servers, which would be especially beneficial in remote or low-connectivity areas. Another enhancement could involve the use of drone or aerial surveillance, where UAVs equipped with thermal and visual cameras could be deployed to detect flames and smoke over large, difficult-to-reach areas, such as forests or industrial zones. Another area for future enhancement is the use of multi-modal sensors, including temperature sensors, smoke detectors, and gas sensors, to complement the image-based detection, providing a more robust and reliable detection system. The integration of Internet of Things (IoT) technology could allow for automated alerting to emergency responders and smart integration with fire suppression systems for immediate action.

Finally, the system could be optimized for use in diverse weather conditions, incorporating adaptive algorithms that account for environmental factors such as fog, heavy rain, or varying lighting conditions. This would ensure that the fire detection system performs reliably in all situations, further enhancing safety and prevention measures in a wide range of real-world scenarios.

CHAPTER 8

CONCLUSIONAND REFERENCES

10.1 CONCLUSION

In conclusion, this project demonstrates the potential of the YOLOv10 algorithm in advancing fire detection systems, providing a highly efficient and accurate method for detecting flames and smoke in real-time. By addressing common challenges such as cluttered backgrounds, varying fire intensities, and low visibility, the proposed system significantly improves detection performance compared to previous methods. Through the integration of advanced techniques like enhanced feature extraction, attention mechanisms, and improved bounding box regression, the system is capable of precise flame and smoke localization, even in complex and challenging environments.
The real-time capabilities of the system, coupled with its ability to adapt to different settings, make it a valuable tool for enhancing fire safety and prevention. Whether in industrial plants, residential areas, or public spaces, the system provides an effective means of early fire detection, potentially saving lives and reducing property damage. The use of transfer learning and data augmentation further enhances the system's robustness and efficiency, making it scalable and suitable for a wide range of applications.
Overall, this project contributes to the development of more reliable and adaptable fire detection technologies, paving the way for future advancements that could incorporate multi-hazard recognition, edge computing, and IoT integration to further improve safety and response times in the face of fire-related incidents.

10.2 REFERENCES

1] A. K. Jaiswal, P. Tiwari, S. Kumar, D. Gupta, A. Khanna, and J. J. P. C. Rodrigues, ‘‘Identifying pneumonia in chest X-rays: A deep learning approach,’’ Measurement, vol. 145, pp. 511–518, Oct. 2019.
[2] D. M. Gavrila and V. Philomin, ‘‘Real-time object detection for ‘smart’ vehicles,’’ in Proc. 7th IEEE Int. Conf. Comput. Vis., Sep. 1999, pp. 87–93.
[3] L. Wei, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, ‘‘SSD: Single shot MultiBox detector,’’ in Computer Vision— ECCV. Cham, Switzerland: Springer, 2016.
[4] S. Chen, J. Hong, T. Zhang, J. Li, and Y. Guan, ‘‘Object detection using deep learning: Single shot detector with a refined feature-fusion structure,’’ in Proc. IEEE Int. Conf. Real-Time Comput. Robot. (RCAR), Aug. 2019, pp. 219–224.
[5] L. Cui, R. Ma, P. Lv, X. Jiang, Z. Gao, B. Zhou, and M. Xu, ‘‘MDSSD: Multi-scale deconvolutional single shot detector for small objects,’’ Sci. China Inf. Sci., vol. 63, no. 2, pp. 98–100, 2020. [6] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, ‘‘DSSD: Deconvolutional single shot detector,’’ 2017, arXiv:1701.06659.
[7] P. Shamsolmoali, M. Zareapoor, E. Granger, J. Chanussot, and J. Yang, ‘‘Enhanced single-shot detector for small object detection in remote sensing images,’’ in Proc. IEEE Int. Geosci. Remote Sens. Symp., 2022, pp. 1716–1719.
[8] D. Jiang, B. Sun, S. Su, Z. Zuo, P. Wu, and X. Tan, ‘‘FASSD: A feature fusion and spatial attention-based single shot detector for small object detection,’’ Electronics, vol. 9, no. 9, p. 1536, Sep. 2020.
[9] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, ‘‘YOLOv4: Optimal speed and accuracy of object detection,’’ 2020, arXiv:2004.10934.
[10] J. Redmon and A. Farhadi, ‘‘YOLO9000: Better, faster, stronger,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jul. 2017, pp. 6517–6525.
[11] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, ‘‘YOLOX: Exceeding YOLO series in 2021,’’ 2021, arXiv:2107.08430.
[12] X. Huang, X. Wang, W. Lv, X. Bai, X. Long, K. Deng, Q. Dang, S. Han, Q. Liu, X. Hu, D. Yu, Y. Ma, and O. Yoshie, ‘‘PP-YOLOv2: A practicalobject detector,’’ 2021, arXiv:2104.10419.

[13] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, ''You only look once:Unified, real-time object detection,'' in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 779–788.
[14] J. Redmon and A. Farhadi, ''YOLOv3: An incremental improvement,'' 2018, arXiv:1804.02767.
[15] Y.-L. Chung, H.-Y. Chung, and C.-W. Chou, ''Efficient flame recognition method based on a deep convolutional neural network and image processing,'' in Proc. IEEE 8th Global Conf. Consum. Electron. (GCCE), Oct. 2019, pp. 573–574.
[16] C. Chaoxia, W. Shang, and F. Zhang, ''Information-guided flame detection based on faster R-CNN,'' IEEE Access, vol. 8, pp. 58923–58932, 2020.
[17] H. Yar, Z. A. Khan, F. U. M. Ullah, W. Ullah, and S. W. Baik, ''A modified YOLOv5 architecture for efficient fire detection in smart cities,'' Exp. Syst. Appl., vol. 231, Nov. 2023, Art. no. 120465.
[18] M. N. Uddin, M. Sakibul Islam Sakib, S. Nawer, and R. T. Mohona, ''Improved fire detection by YOLOv8 and YOLOv5 to enhance fire safety,'' in Proc. 26th Int. Conf. Comput. Inf. Technol. (ICCIT), Dec. 2023, pp. 1–6.
[19] Z. Dai, ''Image flame detection method based on improved YOLOv3,'' IOP Conf. Ser., Earth Environ. Sci., vol. 693, Mar. 2021, Art. no. p. 012012.
[20] Y.-Y. Qin, J.-T. Cao, and X.-F. Ji, ''Fire detection method based on depthwise separable convolution and YOLOv3,'' Int. J. Autom. Comput., vol. 18, no. 2, pp. 300–310, Apr. 2021.
[21] X. Wu, Y. Cao, X. Lu, and H. Leung, ''Patchwise dictionary learning for video forest fire smoke detection in wavelet domain,'' Neural Comput. Appl., vol. 33, no. 13, pp. 7965–7977, Jul. 2021.
[22] Y. Cao, Q. Tang, X. Wu, and X. Lu, ''EFFNet: Enhanced feature foreground network for video smoke source prediction and detection,'' IEEE Trans. Circuits Syst. Video Technol., vol. 32, no. 4, pp. 1820–1833, Apr. 2022.
[23] F. Yang, Q. Xue, Y. Cao, X. Li, W. Zhang, and G. Li, ''Multi-temporal dependency handling in video smoke recognition: A holistic approach spanning spatial, short-term, and long-term perspectives,'' Exp. Syst. Appl., vol. 245, Jul. 2024, Art. no. 123081.
[24] Y. Liu, Z. Shao, and N. Hoffmann, ''Global attention mechanism: Retain information to enhance channel-spatial interactions,'' 2021, arXiv:2112.05561.
[25] Q. Hou, D. Zhou, and J. Feng, ''Coordinate attention for efficient mobile network design,'' in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2021, pp. 13713–13722.
[26] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, ''ECA-net: Efficient channel attention for deep convolutional neural networks,'' in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 11531–11539.
[27] S. Liu, D. Huang, and Y. Wang, ''Learning spatial fusion for single-shot object detection,'' 2019, arXiv:1911.09516.
[28] X. Wang, T. Xiao, Y. Jiang, S. Shao, J. Sun, and C. Shen, ''Repulsion loss: Detecting pedestrians in a crowd,'' in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jul. 2017, pp. 7774–7783.
[29] C.-Y. Wang, H.-Y. Mark Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, ''CSPNet: A new backbone that can enhance learning capability of CNN,'' in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), Jun. 2020, pp. 1571–1580.
[30] T. Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, ''Feature pyramid networks for object detection,'' in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jul. 2017, pp. 2117–2125.
[31] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, ''Path aggregation network for instance segmentation,'' in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 8759–8768.
[32] S. Woo, J. Park, J. Y. Lee, and I. S. Kweon, ''CBAM: Convolutional block attention module,'' in Proc. Eur. Conf. Comput. Vis. (ECCV). Cham, Switzerland: Springer, 2018, pp. 3–19.
[33] X. Zhang, X. Zhou, M. Lin, and J. Sun, ''ShuffleNet: An extremely efficient convolutional neural network for mobile devices,'' in Proc. IEEE/CVFConf. Comput. Vis. Pattern Recognit., Jun. 2018,
pp. 6848–6856.