*Mindanao Journal of Science and Technology Vol. 22 (2) (2024) 123-142*

# Video-based Fire Detection and Reporting System for Immense Urban Areas

Al-Monte Vince M. Calo* and Jocelyn B. Barbosa
College of Information Technology and Computing
University of Science and Technology of Southern Philippines – Cagayan de Oro
Cagayan de Oro City, 9000 Philippines
**vince.calo@ustp.edu.ph*

Date received: May 30, 2024
Revision accepted: November 5, 2024

---

## Abstract

*Fire disasters cause significant physical and emotional trauma worldwide, resulting in the loss of thousands of lives and billions of dollars in property damage. Delayed detection and reporting of fire incidents lead to slower response times, exacerbating the impact of fires. This study presents a video-based fire detection method integrated with a Geographic Information System (GIS)-based reporting platform aimed at enhancing real-world firefighting efforts. The fire detection method employed a hybrid approach, combining rule-based fire detection with a trained Support Vector Machine (SVM) classifier. The rule-based method utilized motion and color-based segmentation techniques, while the SVM classifier was trained using the chromatic and temporal features of fire objects. Test results showed that the module could detect fire in video streams with an overall classification accuracy of 99.66%, along with precision and recall rates of 99.38% and 99.97%, respectively, supported by an F1-score of 99.68%. This approach demonstrated excellent performance in predicting fires with minimal false alarms. Additionally, the GIS-based platform provides real-time information to responding agencies, further enhancing the effectiveness of firefighting operations.*

***Keywords****:* fire detection, GIS, image processing, SVM, video

---

## 1. Introduction

Fire disasters cause significant physical and emotional trauma to victims. In 2017, fire services worldwide reported approximately 3.1 million fire cases and 16.8 thousand civilian fire deaths (WFSR, 2019). These incidents result in billions of dollars in property damage and countless lives affected by trauma. While many countries have reduced fire-related deaths through improved prevention efforts, low- and middle-income countries still experience high incidences of fire-related deaths and injuries (WFSC, 2012).

As a developing country, the Philippines continues to struggle with high rates of fire incidents, along with related deaths and injuries. Approximately 16,000 fire incidents were recorded annually from 2015 to 2018 (PSA, 2019). Common causes include faulty electrical wiring and neglected open flames, though many causes remain undetermined. Most fires occur between 12:01 A.M. and 3 A.M., primarily in residential areas (70.2%) and commercial areas (16.9%) (Velasco, 2013). Fire incidents often remain undetected until visible from the outside, leading to delayed reporting and challenging fire response efforts. The ideal response time is 5 to 7 minutes, but delays result in a wider fire spread upon responders' arrival. Furthermore, traditional sensor-based fire alarm systems are prone to false alarms and rely on basic fire cues, leading to slower fire detection, particularly in large, open areas. Hence, computer vision-based fire detection offers a promising alternative, addressing the limitations of conventional sensors.

Generally, computer vision-based fire detection methods can be classified into traditional rule-based and learning-based approaches. Rule-based methods (Chen *et al*., 2004; Gunawaardena *et al*., 2016; Chi *et al*., 2017; Vijayalakshmi and Muruganand, 2018) detect fire by analyzing features such as fire color, motion, shapes, and combined techniques. These methods are simple to implement and can achieve fire classification accuracy ranging from 71.43% to 98.89%. However, detecting large or tiny flames remains a significant challenge. In contrast, the learning-based approach employs machine learning to detect fire in videos. Various convolutional neural network (CNN) architectures, such as AlexNet, SqueezeNet, and GoogleNet have been optimized for fire detection in surveillance systems (Muhammad *et al*., 2018, 2019). These architectures achieved high accuracy rates of 94.3 % to 94.5 %. However, they require a large number of network parameters and significant large on-disk storage, limiting their deployment on low-computing devices for real-time system applications.

To address this challenge, several studies have utilized traditional machine learning techniques for fire detection systems in combination with rule-based techniques, resulting in some hybrid approaches that relatively consume fewer computing resources. Specifically, the researchers segmented fire and non-fire areas using the Hue Saturation Value (HSV) color analysis and the Gaussian Mixture Model (GMM) to extract Gabor filter statistical features. These features then serve as training input for a Support Vector Machine (SVM) classifier (Li *et al*., 2023). Similarly, an SVM classifier was trained to predict fire using convex hull, area variability, and centroid stability features (Darajat *et al*., 2023). These fire classification models yielded accuracies of 97.34% and 86.61%, respectively. Alternatively, other researchers minimized CNN network parameters to achieve lightweight CNN architectures. For example, modifications to a shallow neural network for fire detection enabled deployment in mobile and embedded applications, achieving a prediction accuracy of 93.91% (Jadon *et al*., 2019). Another lightweight convolutional neural network employed a feature extractor that utilized convolution layers, depth-wise separable convolution layers, an inception module, and an attention mechanism to extract high-level feature maps. Additionally, its classifier applied a global average pooling layer for feature map dimension reduction and used the Softmax function to calculate the probability of each class, achieving a prediction accuracy of 96.08% (Nguyen *et al*., 2023).

While these developments have enhanced the accuracy of fire detection, there is still considerable room for improvement in fire prediction accuracy, highlighting the need for additional research in this area. More importantly, despite advancements in the field, existing computer vision-based fire detection systems have significantly impact real-world firefighting efforts, as they lack a real-time reporting mechanism that could assist first responders in promptly responding to fire incidents. Geographic Information System (GIS) technology can optimize emergency services by supporting planning, preparedness, mitigation, response, and incident management (ESRI, 2007). However, GIS remains underutilized in fire incident management.

Hence, this study introduces a video-based fire detection and reporting system for large urban areas that accurately detects fires through live video feeds and automatically reports them to a GIS-based fire incident management platform to eliminate communication gaps among first responders during fire disasters. This paper makes several notable contributions that significantly advance the understanding and application of fire detection and incident management technologies. It presents a hybrid fire detection method that combines rule-based detection with a Support Vector Machine (SVM) classifier, achieving accurate fire prediction in videos through a novel fire object persistence density feature. Furthermore, it integrates a GIS-based fire incident management platform designed to address communication gaps among first responders. This platform facilitates coordinated response efforts and supports informed decision-making by providing timely access to relevant information.

## 2. Methodology

The proposed framework of the integrated Fire Detection and Reporting Platform consists of two components: the fire detection and the fire incident reporting module. The concept involves detecting fire within a vicinity using a fire detection camera, which triggers an alarm in a fire-affected establishment and automatically reports the incident to the GIS Reporting Platform. The platform receives the GPS location and other fire-related data, which are plotted as location markers on the GIS map. These markers indicate the locations of different first responders, along with the nearest responding station relative to the fire location. Additional relevant information about the fire incident, including the captured fire image, is tagged on the markers. The different first responders will receive a real-time fire incident notification, as illustrated in Figure 1.

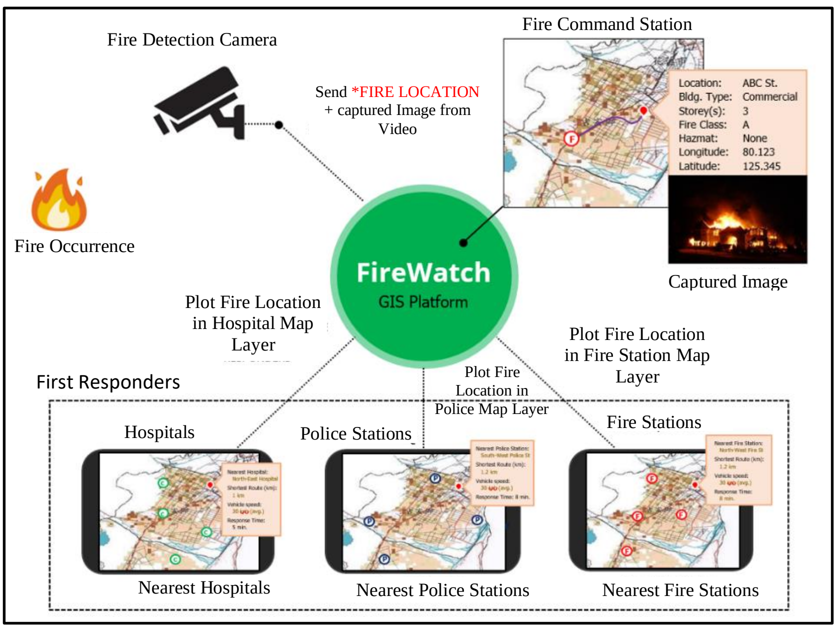



Figure 1. Framework of the Integrated Fire Detection and Reporting

### *2.1 Video-Based Fire Detection Module*

The hybrid fire detection method's flowchart illustrates the steps to identify fire in video streams, as depicted in Figure 2.

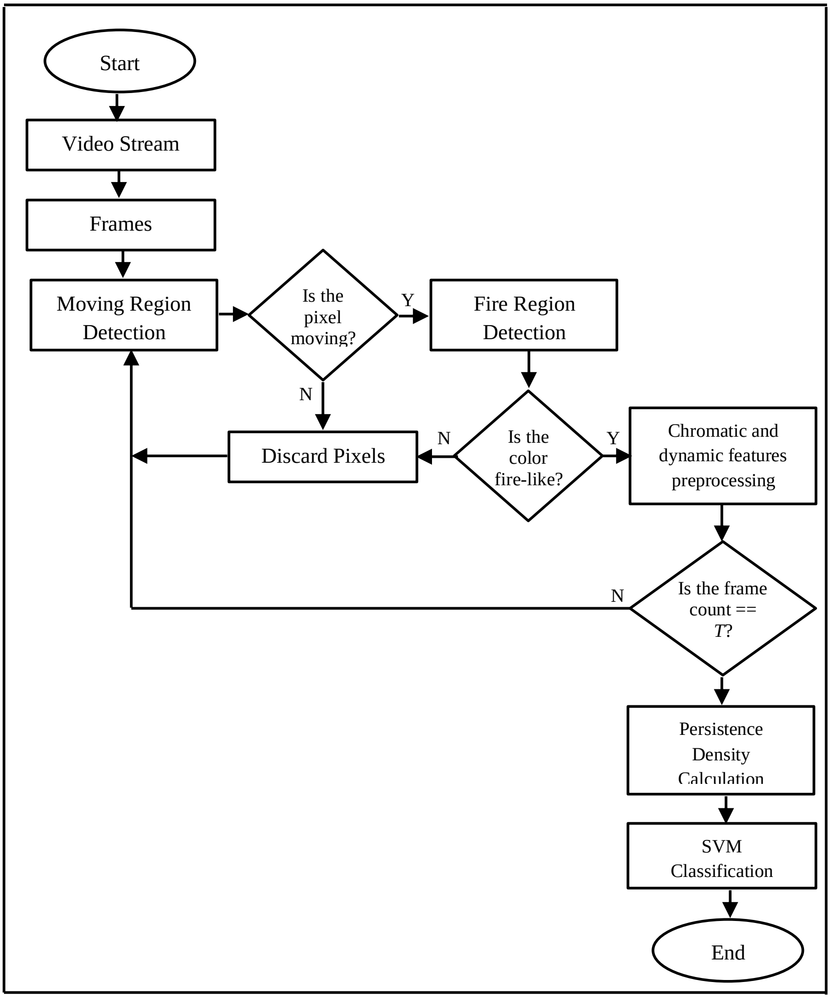



Figure 2. Hybrid Fire Detection Flowchart

The process begins with Video Frames Segmentation, where incoming video streams are divided into manageable segments for analysis. Following this, Moving Region Detection identifies regions of interest that exhibit movement, which helps to isolate dynamic areas for further examination. Once movement is detected, the system proceeds to Fire Region Detection, applying a color masking technique to identify fire-like characteristics in the detected regions and extract chromatic and temporal features.

Next, the system performs Fire Object Persistence Density Calculation, evaluating the temporal persistence of fire regions across multiple frames.

This step enhances detection accuracy by reducing false positives. Finally, the results are fed into an SVM Classification step, where a Support Vector Machine classifier confirms whether the detected region is indeed a fire.

### 2.1.1 Moving Region Detection

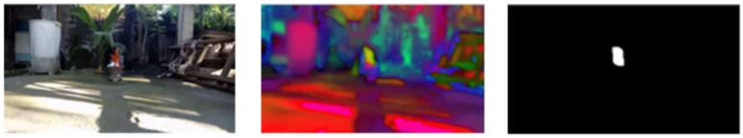
The moving region in each frame of the video stream is detected using absolute frame differencing. The approach begins by initializing the first frame as a reference or background model, which is then used to subtract any changes occurring in the subsequent frames. The difference between two frames is computed through a subtraction operation, where the absolute value of the differences in their corresponding pixel intensities (V-values of the HSV color model) is calculated, as shown in Equation 1.

$$V_{delta} = | V_{background\ model} - V_{current\ frame} | \quad (1)$$

The $V_{delta}$ are the significant changes in pixel intensity values of the current frame from the background model or the first frame. The frame delta will then be a threshold to reveal the regions of the image that have significant changes. If the delta is less than 25, the pixel is discarded and set it to black (i.e., background). If the delta is greater than 25, it is set to white (i.e., foreground). This threshold is intuitively chosen, as values below 25 were found to represent no significant changes based on the experimental observations.

### 2.1.2 Fire Region Detection

In this study, the frame in the RGB color space was converted to the HSV (Hue, Saturation, Value/Brightness) color space to separate pixel values into intensity and chrominance, as shown in Figures 3a and 3b. In the HSV color space, detection is not solely dependent on hue but also considers saturation and luminance values, which help filter out non-fire pixels in the frames. To achieve this, the process involved three steps: (1) collect a fire image dataset and manually extract the RGB values of fire pixel samples; (2) compute the average RGB values from all pixel samples; and (3) convert the average RGB values to the HSV color space to determine the upper and lower limits of HSV values for the color masking process. At the end of the sampling process, which included 1,250 diverse fire images, a range of HSV values was established: 0, 100, 100 (lower limit) to 25, 255, 255 (upper limit). This range was used to extract fire-like pixels from any given frame.



(a) (b) (c)

Figure 3. RGB to HSV frame conversion and manipulation

2.1.3 Fire Object Persistence Density Calculation

In typical fire scenarios, real fire remains within a consistent spatial area across video frames, gradually expanding or contracting, whereas non-fire objects appear and disappear unpredictably. To address this, *Persistence Density*, a novel metric for spatio-temporal fire analysis, was introduced. It clusters objects based on their consistency in appearing within similar regions over time. By assessing the stability and location of these objects, it can effectively distinguish between persistent fire and sporadic non-fire elements, enabling more accurate fire detection.

To quantify the persistence of fire regions in a video, each frame is processed to detect the presence of fire. Given a sequence of video frames, each frame $F_t$, where $t$ denotes the frame index, undergoes fire detection. Fire detection is accomplished by generating a binary mask $M_t(x, y)$ for each frame. In this mask, $M_t(x, y) = 1$ if fire is detected at pixel $(x, y)$ in frame $t$ and $M_t(x, y) =$ 0 otherwise.

Subsequently, a *persistence matrix* $P(x, y)$ is computed for each pixel, which counts the total number of frames in which fire is detected at that pixel across a sequence of $t$ frames. The persistence matrix is given by Equation 2:

$$P(x,y) = \sum_{t=1}^{T} M_t(x,y) \qquad (2)$$

To account for transient fire occurrences, a *time-decay weighted persistence matrix* is introduced, which assigns more importance to recent fire detections. This is achieved using an exponential decay function (Equation 3), where $\lambda$ is the decay rate:

$$P(x,y) = \sum_{t=1}^{T} M_t(x,y) \cdot e^{-\lambda(T-t)} \qquad (3)$$

Finally, the *persistence density* $D(x,y)$ (Equation 4) is calculated by normalizing the persistence matrix by the total number of frames $T$. This results in a value between 0 and 1, representing the proportion of frames in which fire is detected at pixel$(x,y)$:

$$D(x,y) = \frac{P(x,y)}{T} \quad (4)$$

2.1.4 SVM Classification

In this study, the SVM classifier was trained using 19 features, consisting of the mean and standard deviation of color features for both HSV and RGB, as well as spatio-temporal features, including persistence density. The hyperparameters of the SVM classifier were configured with an 'rbf' kernel, gamma set to 'auto', a regularization parameter C was set to 1, and class weight was set to 'balanced'.

Consequently, classification of data samples was performed for $T$ number of frames, which corresponds to the window size used for calculating the persistence density and other temporal features. In the experiment, the value of $T$ was set to 10. Cameras typically capture images at an average of 30 frames per second, including the Raspberry Pi Camera module. Therefore, classification occurs every one-tenth of a second. In the end, the model predicted based on the input variables, which resulted in either 0 (NO FIRE) or 1 (FIRE).

2.1.5 Training and Validation Dataset

Due to the scarcity of fire video datasets, additional videos were recorded and supplemented with publicly available datasets, including those from FireNet (Jadon *et al.*, 2019) and Foggia's (Foggia *et al.*, 2015) collections. Consequently, the training and validation dataset consisted of 193 videos, including 101 fire videos and 92 non-fire videos. The sample videos in the dataset are depicted in Figure 4.

The number of segmented and extracted fire and non-fire regions from each video frame determines the sample count of the dataset, not the total number of videos. Table 1 presents the aggregate of extracted fire and non-fire samples.

Table 1. Training and Validation dataset

| Source | Fire samples | Non-fire samples |
|---|---|---|
| Own dataset | 20,554 | 20,666 |
| FireNet | 4,274 | 1,861 |
| Foggia's | 12,699 | 7,587 |
| Total | 37,527 | 30,114 |

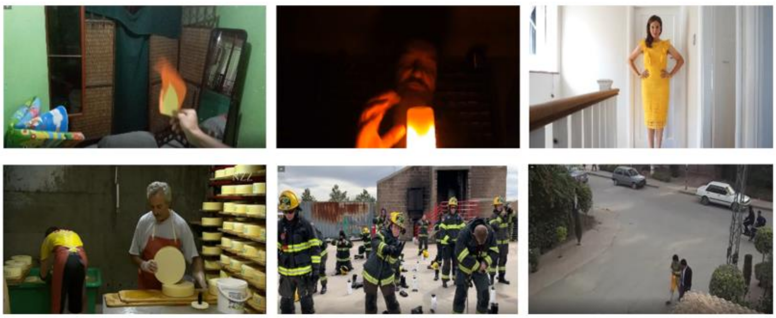


Figure 4. Training and Validation Video Samples

2.1.6 Performance Metrics

The confusion matrix is a fundamental tool used in machine learning to evaluate the performance of a classification model. It provides a visual representation of the model's predictions compared to the actual classes. This matrix helps identify how effectively the model distinguishes between different classes, with its components detailed in Table 2.

Table 2. Confusion Matrix

| | | Actual Label | |
|---|---|---|---|
| | | Positive | Negative |
| Predicted Label | Positive | TP | FP |
| | Negative | FN | TN |

The following are the terms associated with the Confusion Matrix:

1. True Positives (TP): Cases where fire is present (1, True) and the model correctly predicts fire (1, True).

2. True Negatives (TN): Cases where fire is absent (0, False) and the model correctly predicts no fire (0, False).
3. False Positives (FP): Cases where fire is absent (0, False) but the model incorrectly predicts fire (1, True). This is a false prediction indicating a fire when there isn't one.
4. False Negatives (FN): Cases where fire is present (1, True) but the model incorrectly predicts no fire (0, False). This is a false prediction indicating no fire when there is one.

The evaluation of the SVM model's performance was conducted using performance metrics derived from the Confusion Matrix presented in Table 3.

Table 3. Performance Metrics of Fire Classification Models

| Metrics | Formula | Description |
|---|---|---|
| Accuracy | $\frac{TP+TN}{TP+TN+FP+FN}$ | Overall ability of a model to make correct predictions. |
| Precision | $\frac{TP}{TP+FP}$ | Ability of model to correctly predict within the fire class |
| Sensitivity (Recall) | $\frac{TP}{TP+FP}$ | Ability of model to correctly predict fire labels |
| Specificity | $\frac{TP}{TP+FP}$ | Ability of model to correctly predict non-fire labels |
| F1-score | $\frac{2 \times Precision \times Sensitivity}{TP+FP}$ | Harmonic mean of sensitivity and precision |
| False Positive Rate (FPR) | $\frac{FP}{FP+TN}$ | Proportion of non-fire labels that are incorrectly classified as fire labels |
| False Negative Rate (FNR) | $\frac{FN}{FN+TP}$ | Proportion of fire labels that are incorrectly classified as non-fire labels |

*2.2 Fire Incident Reporting Module*

Figure 5 illustrates the key components and flow of information within the Fire Incident Reporting Platform. It showcases the integration of various modules, including data collection from the camera, real-time fire detection using a hybrid fire detection method, and user interfaces for incident reporting. The architecture also highlights the Incident Management Dashboard, notification systems for alerting stakeholders, and cloud storage for secure data management. Overall, this figure provides a visual representation of how the platform enhances fire safety management through efficient detection and reporting.

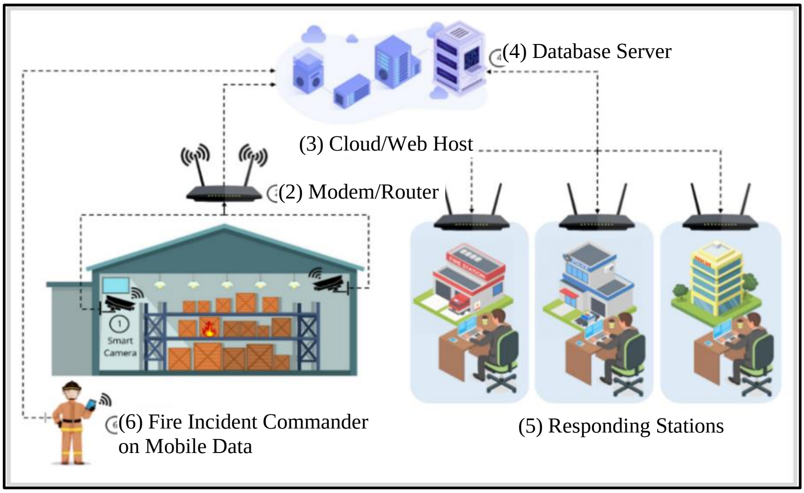



Figure 5. Conceptual Framework of the Fire Incident Reporting Platform

As illustrated in Figure 5, the Fire Incident Reporting Module involves one or more (1) fire detection cameras connected to a wired or wireless local area network (LAN) through a (2) router or modem linked to the cloud. This connection is necessary to enable any of the fire detection cameras to transmit data to a (3) cloud or web host whenever fire is detected within a certain area. The cloud/web host provides both web and database service and includes a (4) database server accessible to multiple (5) first responders. First responders from different agencies can retrieve fire information from the database through a web application served on the internet. Likewise, the (6) fire incident commander of the ongoing response operation will have access to fire information in the web application via a mobile device and will be able to set the alarm level of the fire in real-time.

### 2.2.1 Fire Incident Reporting Web Application

The fire incident reporting web application is built on an edge computing architecture as shown in Figure 6.

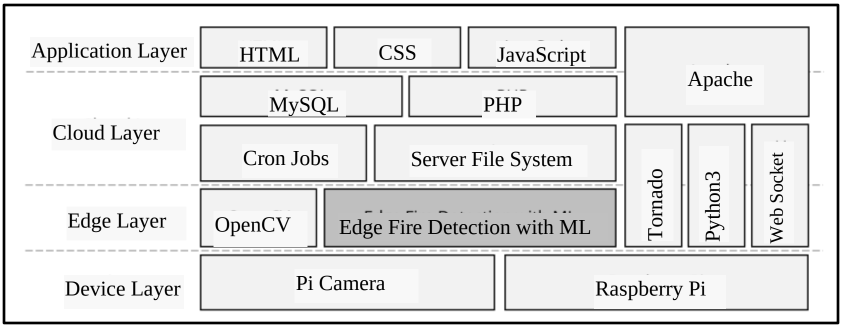



Figure 6. System Layered Architecture

The system architecture for the fire incident reporting web application is organized into four layers: application, cloud, edge, and device. The application layer functions as the front end, built with HTML, CSS, and JavaScript, providing users with an interface to view fire alerts and incident reports. This layer communicates seamlessly with the backend to provide real-time fire detection data. The cloud layer manages backend processes, utilizing MySQL for database management, PHP for server-side logic, and the server file system for storage. It also incorporates cron jobs to automate routine tasks like backups and report generation. This layer efficiently handles data processing and storage, working with the edge layer for real-time updates. The edge layer focuses on real-time fire detection using OpenCV and the machine learning model, analyzing data from devices like cameras to detect fire incidents locally and reduce latency. This layer communicates detection results to the cloud for further action or storage. The device layer includes hardware like Raspberry Pi units and Pi Cameras, which capture video and send it to the edge for immediate processing. Background software such as Apache connects the application and cloud layers, while Tornado with WebSocket ensures real-time communication between the edge and cloud, supporting fast, low-latency fire detection across the system.

### 2.2.2 Functionality Testing

The fire reporting platform was tested using test cases to verify its functionality with certain data inputs. The main focus of the test was to determine whether the fire detection module could effectively communicate with the fire reporting platform through the layers of the edge computing architecture. The actual functionality tests involved identifying expected outputs to serve as a basis for comparison with the system's actual behavior. A total of 10 testers were involved in the study.

## 3. Results and Discussion

All experiments conducted in this study were performed on a local machine with an Intel Core i7, 7th Gen Processor CPU at 2.80 GHz, 12 GB RAM, and an NVIDIA GeForce GTX 1050 2GB. The dataset was split into 80% for training and 20% for validation. Additionally, 27 videos from the FireSense dataset (Grammalidis *et al.*, 2017) were used as the test dataset.

### *3.1 Feature Importance Analysis*

The feature importance analysis for the SVM classifier in fire detection was done using the SHapley Additive exPlanations (SHAP), highlighting the contributions of key features to the model's accuracy as shown in Figure 7.

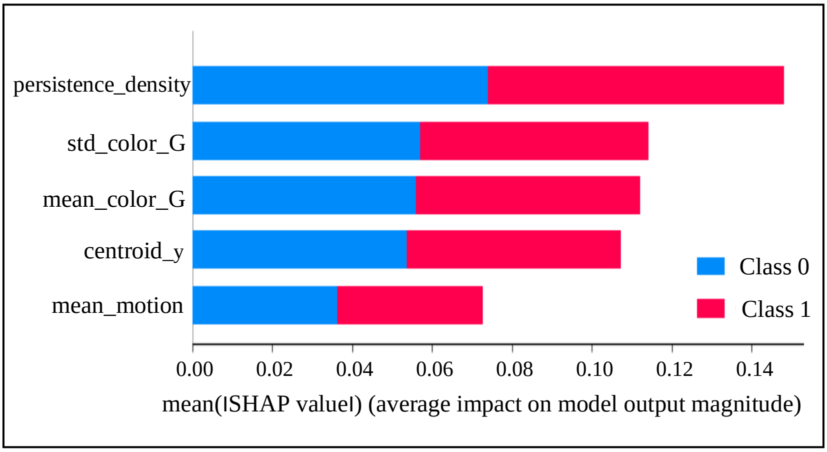



Figure 7. SHAP Analysis of Feature Impact in SVM Classifier

The SHAP analysis highlights the significant influence of key features on the SVM classifier's performance in fire detection. As shown in Table 4, the SVM classifier achieved accuracy rates of **98.90%** for training and **98.83%** for validation, demonstrating its reliability in differentiating between fire and non-fire events. Notably, the persistence_density feature has the highest

SHAP value, making a substantial contribution to the model's decisions, particularly in predicting Class 1 (fire events). Its impact is reflected in the low false positive rates (0.80% for training and 1.06% for validation) and false negative rates (1.39% and 1.26%), ensuring the model effectively minimizes false alarms and accurately identifies real fire incidents.

Table 4. Performance of SVM classifier

| Metrics | Training (%) | Validation (%) | Test (%) |
|---|---|---|---|
| Accuracy | 98.90 | 98.83 | 99.66% |
| False Positive | 0.80 | 1.06 | 0.62% |
| False Negative | 1.39 | 1.26 | 0.03% |
| Precision | 99.24 | 98.99 | 99.38% |
| Recall | 98.61 | 98.74 | 99.97% |
| F1-score | 98.92 | 98.86 | 99.68% |

Furthermore, the high precision (99.24% for training and 98.99% for validation) indicates that fire predictions are highly accurate, while the impressive recall values (98.61% and 98.74%) demonstrate the model's ability to detect most fire events. With a test dataset accuracy of 99.66%, the model proves its reliability when faced with unseen data, confirming its effectiveness across different conditions. The balanced F1 scores across datasets show a good balance between precision and recall, making the SVM classifier a reliable tool for fire detection. Figure 8 depicts the fire prediction samples from the test dataset.

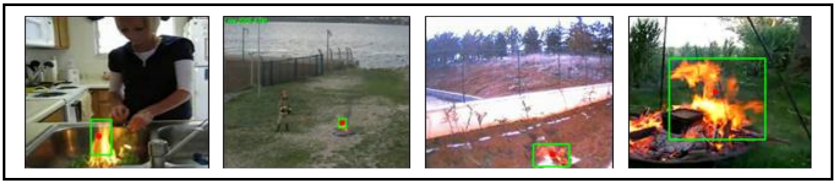


Figure 8. Fire detection samples in test dataset

*3.2 Comparative Analysis*

The confusion matrix in Table 5 shows the actual labels against the predicted labels based on the classification result of the trained SVM model on the test dataset.

Table 5. Confusion matrix of prediction on the FireSense dataset.

| Predicted Label | Actual Label: Fire | Actual Label: Not Fire |
|---|---|---|
| Fire | 7540 | 47 |
| Not Fire | 2 | 6940 |
| Support | 7542 | 6987 |

The confusion matrix highlights the fire detection model's performance, showing that out of 7,542 predicted "Fire" instances, 7,540 were correct, with only 2 false negatives. Conversely, the model accurately identified 6,940 out of 6,987 non-fire instances, resulting in just 47 false positives. This indicates a strong capacity for distinguishing between fire and non-fire events, as evidenced by the minimal misclassification. These values were instrumental in calculating key performance metrics, including accuracy, false positive rate (FPR), false negative rate (FNR), precision, recall, and F1-score on the test dataset, which were compared to existing fire detection models, as depicted in Table 6.

Table 6. Performance comparison against SVM-based hybrid methods

| Metrics | Jadon *et al*. (2019) | Darajat *et al*. (2023) | Proposed Method |
|---|---|---|---|
| Accuracy | 93.91 | 86.61 | 99.66% |
| False Positive | 1.95 | 12.50 | 0.62% |
| False Negative | 4.13 | 14.69 | 0.03% |
| Precision | 94 | 82.43 | 99.33% |
| Recall | 97 | 85.31 | 99.97% |
| F1-score | 95 | 83.85 | 99.65% |

The performance metrics of various SVM-based hybrid methods, as shown in the table above, reveal significant improvements in fire detection capabilities. The proposed method achieves an impressive accuracy of 99.66%, indicating optimizations in model performance compared to previous works. Specifically, Jadon *et al*. (2019) and Darajat *et al.* (2023) reported accuracies of 93.91% and 86.61%, respectively. Additionally, the proposed method's false positive and false negative rates are 0.62% and 0.03%, respectively, suggesting improved precision by minimizing false alarms and missed detections. Noteworthy enhancements in precision and recall are also evident, with the proposed model reaching 99.33% and 99.97%, respectively. These improvements reflect a refined balance between identifying true positives and reducing errors. Although these models were not assessed on the same datasets, the advancements highlighted in these metrics underscore the ongoing progress in SVM-based hybrid fire detection methodologies.

Furthermore, to gain a better understanding of how the proposed hybrid model compares with state-of-the-art CNN-based models tested on the FireSense dataset, a comparison of their respective overall accuracies is presented in Table 7.

Table 7. Accuracy comparison with CNN-based models on FireSense dataset

| Model | Accuracy (%) |
|---|---|
| SqueezeNet | 100 |
| MobileNetV2 | 100 |
| VGG13 | 87.03 |
| NASNetMobile | 96.76 |
| InceptionV3 | 96.76 |
| Proposed Hybrid Model | 99.66 |

The accuracy of various models on the FireSense dataset showcases the effectiveness of different architectures in fire detection tasks. As indicated in Table 7, both SqueezeNet and MobileNetV2 achieve perfect accuracy of 100%, demonstrating their exceptional performance in this specific context. In contrast, other models, such as VGG13 with an accuracy of 87.03%, and NASNetMobile and InceptionV3, both at 96.76%, exhibit varying levels of efficacy. Notably, the proposed hybrid model achieves a robust accuracy of 99.66%, positioning it as a highly effective alternative among the tested models. These results emphasize the competitive nature of contemporary fire detection models while highlighting the potential of hybrid approaches to bridge the performance gaps observed in traditional convolutional neural network (CNN) architectures.

### *3.3 Functionality Test Results*

Ten participants were involved in testing various scenarios to verify whether the fire detection module could effectively communicate through the layers of the edge computing architecture. Each test case compared the system's actual behavior with predefined expected outputs. Notably, all 10 participants confirmed that the system performed the task as expected, successfully detecting fire incidents and relaying alerts through the platform without issues. This consistency across all testers demonstrates the platform's robustness, confirming that the fire detection and reporting modules operated as intended across the device, edge, cloud, and application layers. A sample test case illustrating this process is shown in Figure 9. Additionally, the interface displaying a GIS map for plotting reported fire incidents is shown in Figure 10, offering a visual representation of fire locations in real-time. These results validate the platform's functionality and its seamless integration.

| | | | | | |
|---|---|---|---|---|---|
| **Test Case ID** | BU_002 | **Test Case Description** | Test the Automated Fire Detection and Reporting Functionality | | |
| **Created By** | AV Calo | **Reviewed By** | Adviser | **Version** | 1.0 |
| **Tester's Name** | Tester 1 | **Date Tested** | May 21, 2024 | **Test Case (Pass/Fail/Not** | **Passed** |

| **S #** | **Prerequisites:** | **S #** | **Test Data** |
|---|---|---|---|
| 1 | Running Apache and MySQL Server (MAMP) | 1 | fire1.mp4 (video file) |
| 2 | Running Thor Server in Python | 2 | |

**Test Scenario** Verify on reporting of detected fire frame in video

| **Step #** | **Step Details** | **Expected Results** | **Actual Results** | **Remarks** |
|---|---|---|---|---|
| 1 | Activate Thor Server | Listening on Port 8080 | As Expected | **Passed** |
| 2 | Run fire detection module on fire1.mp4 in Raspberry Pi | Detect fire on fire1.mp4 image frame | As Expected | **Passed** |
| 3 | Automatically Insert fire image to MySQL database with estab_ID | Insert a fire report to the database (image and estab_ID) | As Expected | **Passed** |
| 4 | Retrieve fire report from database | View fire report of Map View of the Web Application | As Expected | **Passed** |

Figure 9. Sample Test Case for functionality testing

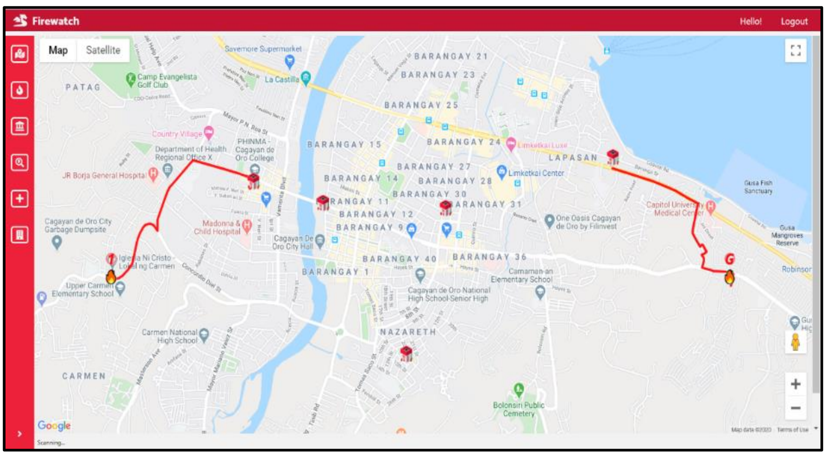



Figure 10. Fire incident Reports on GIS Map

## 4. Conclusion and Recommendation

This study presented a video-based fire detection module integrated with a Geographic Information System (GIS)-based reporting platform to enhance real-world firefighting. The fire detection method employed a hybrid approach, combining a rule-based classification scheme with a Support Vector Machine (SVM) classifier. The introduction of a novel persistence density feature as one of the training inputs, along with other relevant chromatic and dynamic features, had a significant impact on the overall accuracy of the classifier, as supported by the SHapley Additive exPlanations (SHAP) analysis results. Consequently, the hybrid fire detection method achieved a 0.03% false negative rate and a 0.62% false positive rate. The overall classification accuracy was 99.66%, with precision and recall rates of 99.33% and 99.97%, respectively, and an F1-score of 99.65%. These metrics demonstrate the module's excellent performance in predicting fire in video streams with minimal false alarms. Additionally, the GIS-based reporting platform passed all functionality tests, effectively meeting the needs of real-world fire response operations.

Based on these findings, several recommendations were made to improve the study for future research. First, incorporating smoke detection as a criterion in the fire detection module could improve early-stage fire detection. Second, expanding the dataset to include more diverse fire scenarios would enhance the model's robustness. Lastly, it is essential to perform further testing in real-world scenarios to comprehensively assess the system's performance with respect to response time, reliability, user experience, and security. This acknowledgment will not only refine the system's capabilities but also set a clear direction for future research efforts.

## 5. Acknowledgment

The authors would like to acknowledge the contribution of the Bureau of Fire Protection—Region 10 in the completion of this research by extending the knowledge and expertise relevant to the standard fire response procedures.

## 6. References

Brushlinsky, N.N., Ahrens, M., Sokolov, S.V., & Wagner, P. (2019), World Fire Statistics Report. Center of Fire Statistics of International Association of Fire and Rescue Services No. 24

The Geneva Association. (October 2012). World Fire Statistics Bulletin. Retrieved January 25, 2020 from genevaassociation.org: http://genevaassociation.org/PDF/WFSC/GA2012-FIRE28.pdf

Philippine Statistics Authority. (2019). 2019 Philippine Statistical Yearbook. https://psa.gov.ph/system/files/psy/2019-PSY_61322_0.pdf

ESRI (2007). GIS for Fire Station Locations and Response Protocol. An ESRI® White Paper. ESRI 380 New York St., Redlands.

Chen, T.H., Wu, P.H., & Chiou, Y.C. (2004). An early fire-detection method based on image processing. 2004 International Conference on Image Processing, 2004. ICIP '04., 3, 1707-1710 Vol. 3. https://doi.org/10.1109/ICIP.2004.1421401

Chi, R., Lu, Z.M., & Ji, Q.G. (2017). Real-time multi-feature based fire flame detection in video. IET Image Processing, 11(1), 31–37. https://doi.org/10.1049/iet-ipr.2016.0193

Darajat, A.N., Sthevanie, F., & Ramadhani, K.N. (2023). Fire Detection on Video Using Multi-Feature Fusion and Support Vector Machine. 2023 11th International Conference on Information and Communication Technology (ICoICT), 600–605. https://doi.org/10.1109/ICoICT58202.2023.10262454

Foggia, P., Saggese, A., & Vento, M. (2015). Real-Time Fire Detection for Video-Surveillance Applications Using a Combination of Experts Based on Color, Shape, and Motion. IEEE Transactions on Circuits and Systems for Video Technology, 25(9), 1545–1556. https://doi.org/10.1109/TCSVT.2015.2392531

Grammalidis, N., Dimitropoulos, K., & Cetin, E. (2017). Firesense Database Of Videos For Flame And Smoke Detection [Dataset]. Zenodo. https://doi.org/10.5281/ZENODO.836748

Gunawaardena, A.E., Ruwanthika, R.M.M., & Jayasekara, A.G.B.P. (2016). Computer vision based fire alarming system. 2016 Moratuwa Engineering Research Conference (MERCon), 325–330. https://doi.org/10.1109/MERCon.2016.7480162

Jadon, A., Omama, M., Varshney, A., Ansari, M.S., & Sharma, R. (2019). FireNet: A Specialized Lightweight Fire & Smoke Detection Model for Real-Time IoT Applications (arXiv:1905.11922). arXiv. http://arxiv.org/abs/1905.11922

Li, G., Yang, B., Chang, S., Yi, H., Zhang, Y., & Wang, Y. (2023). Gaussian Process for the Machine Learning-based Smart Fire Detection System. 2023 International Conference on Computers, Information Processing and Advanced Education (CIPAE), 446–452. https://doi.org/10.1109/CIPAE60493.2023.00091

Muhammad, K., Ahmad, J., Mehmood, I., Rho, S., & Baik, S.W. (2018). Convolutional Neural Networks Based Fire Detection in Surveillance Videos. IEEE Access, 6, 18174–18183. https://doi.org/10.1109/ACCESS.2018.2812835

Muhammad, K., Khan, S., Elhoseny, M., Hassan Ahmed, S., & Wook Baik, S. (2019). Efficient Fire Detection for Uncertain Surveillance Environment. IEEE Transactions on Industrial Informatics, 15(5), 3113–3122. IEEE Transactions on Industrial Informatics. https://doi.org/10.1109/TII.2019.2897594

Nguyen, D.L., Putro, M.D., & Jo, K.H. (2023). Lightweight Convolutional Neural Network for Fire Classification in Surveillance System. IEEE Access, 11, 101604–101615. IEEE Access. https://doi.org/10.1109/ACCESS.2023.3305455

Velasco, G.N.V. (2013). Epidemiological Assessment of Fires in the Philippines, 2010-2012.

Vijayalakshmi, S.R., & Muruganand, S. (2018). Fire alarm based on spatial temporal analysis of fire in video.