Wildfire Image Detection: Leveraging K-Means Clustering and Supervised Learning Models

CSS581 - Machine Learning

Christian Weimer
Cybersecurity Engineering
University of Washington, Bothell

Clement Jeremie Gomiero
Cybersecurity Engineering
University of Washington, Bothell

Joshua Jiji
Cybersecurity Engineering
University of Washington, Bothell

***Abstract*—Wildfires pose significant environmental, economic and human risks, requiring faster and more reliable detection methods beyond traditional satellite-based and human-reported systems, which often suffer from delays and inaccuracies. Machine learning (ML) and computer vision provide promising alternatives by analyzing satellite, drone, and surveillance imagery for real-time wildfire detection with minimal false positives. While Convolutional Neural Networks (CNNs) are commonly used, this research investigates the effectiveness of traditional ML models, specifically Random Forest and Logistic Regression, with Raw pixels and K-Means clustering as a preprocessing technique. The study examines how K-Means impacts feature representation, particularly its ability to reduce noise and enhance class separability. In addition, both RGB and grayscale images are evaluated to determine the importance of color information in classification performance. Results indicate that Random Forest consistently outperforms Logistic Regression, with K-Means distance features further improving detection accuracy. Compared to CNN-based approaches, the proposed method offers a computationally efficient alternative while maintaining competitive performance. Performance is measured using key metrics such as accuracy, precision, recall and AUC to validate the effectiveness of the model with a given preprocessing, highlighting its potential for real-time wildfire detection in future research.**

***Index Terms*—machine-learning, k-means clustering, random forest, logistic regression, Image processing, classification, RGB images, Gray-scale images**

I. Introduction

Wildfires, also referred to as uncontrolled fires, spread rapidly across forests, vegetation, or grasslands. These fires are primarily fueled by dry conditions, strong winds, and combustible materials such as dry leaves, fallen branches, or tree debris. Although wildfires can be ignited by natural causes, such as lightning strikes, human activities also play a significant role in triggering these disasters. Factors such as unattended campfires, discarded cigarettes, and arson contribute to the increasing number of wildfire occurrences around the world. The impact of climate change has further exacerbated the frequency and intensity of wildfires. Rising global temperatures, prolonged droughts, and unpredictable weather patterns have created conditions conducive to frequent and severe wildfires. According to reports from NASA and the National Interagency Fire Center (NIFC), the annual area burned by wildfires in the United States alone has doubled since 1980, averaging approximately 7.5 million acres per year. In 2020, this figure reached a record-breaking 10.5 million acres, emphasizing the urgent need for effective wildfire prediction and mitigation strategies.

Early detection and prediction of wildfires are crucial to reducing their devastating impact on ecosystems, wildlife, human settlements, and infrastructure. One of the most promising approaches for wildfire prediction involves leveraging historical wildfire data, particularly image datasets, to analyze and identify patterns that precede an outbreak. By examining past wildfire images, critical details such as fire intensity, affected regions, and environmental conditions can be extracted, providing valuable insights for proactive response measures. Traditional methods of wildfire detection rely heavily on satellite monitoring, meteorological data, and human observations. However, these approaches can be time-consuming, prone to errors, and limited by human capabilities. The integration of machine learning (ML) models in wildfire prediction offers a more efficient, scalable, and automated solution. ML algorithms can process vast amounts of image data, recognize complex patterns, and classify wildfire risks with greater accuracy and speed than conventional methods.

The successful implementation of machine learning for wildfire prediction can lead to faster response times, better resource allocation, and improved disaster preparedness. With continuous advancements in artificial intelligence, integrating real-time image processing with environmental data (e.g., temperature, humidity, wind speed) can further enhance prediction accuracy. Future research in this domain should focus on improving model robustness, incorporating multi-modal data sources (e.g., IoT sensors, weather forecasts), and developing cost-effective, real-time wildfire monitoring systems. By harnessing the power of ML-driven wildfire prediction, we can mitigate fire hazards, protect ecosystems, and safeguard human lives.

II. Related Work

Before getting into the core of this research—identifying wildfires using machine learning models—it is essential to examine existing wildfire detection methods and how machine learning enhances them. Current detection techniques, including sensor networks, cameras, and Unmanned Aerial Vehicles (UAVs), already integrate some level of machine learning for fire identification [2]. However, this study explores alternative machine learning models and preprocessing techniques to improve detection accuracy, reduce false positives, and optimize computational efficiency. By evaluating different approaches, this research aims to refine existing methods and contribute to more reliable and scalable wildfire detection systems.

A. Camera and Sensor Nodes

Camera and sensor nodes are critical components of modern wildfire detection systems, combining environmental monitoring with visual detection for improved accuracy and response times. Sensor nodes consist of networks of environmental sensors, including humidity, temperature, and gas sensors, strategically placed in high-risk wildfire areas. These nodes operate autonomously, typically powered by solar energy and rechargeable batteries to ensure long-term functionality. Each node is controlled by a microcontroller, which continuously collects and transmits real-time data to a central monitoring system.

B. Sensor-Based Detection

Wildfire detection using sensor nodes relies on analyzing environmental anomalies, such as sudden temperature spikes, drops in humidity, and increased gas concentrations (e.g., carbon monoxide and volatile organic compounds). These rapid changes indicate potential fire events and can trigger early alerts, allowing for faster response times compared to traditional methods. Sensor-based detection is particularly effective in remote areas where real-time human monitoring is not feasible[6].

C. Camera-Based Detection

While sensor nodes excel at detecting environmental changes, they lack the ability to visually confirm wildfire occurrences. To address this limitation, camera-based detection systems have been widely adopted. These cameras, positioned in high-risk areas, continuously monitor landscapes for visual indicators such as smoke and flames[6]. Equipped with machine learning algorithms, they can automatically differentiate wildfires from human activities, enhancing detection accuracy. By integrating camera-based systems with sensor nodes, detection systems can achieve a more comprehensive and reliable approach, where environmental anomalies trigger further visual analysis before an alert is issued.

D. Unmanned Aerial Vehicles (UAVs)

While sensor and camera networks provide stationary surveillance, they are limited in coverage and flexibility. Unmanned Aerial Vehicles (UAVs), commonly known as drones, have emerged as a powerful alternative for wildfire monitoring by offering real-time aerial data and adaptable coverage. UAVs can operate at various altitudes, reaching heights of up to 30,000 feet, and are equipped with sensors that collect GPS coordinates, high-resolution images, video feeds, and environmental readings[3]. These systems can be remotely controlled by human operators or fully automated using AI-driven flight algorithms, allowing even users with minimal training to operate them efficiently.

E. Machine Learning Advancements in Wildfire Detection

Machine learning (ML) has significantly improved wildfire detection, particularly in analyzing images captured by cameras and UAVs. Since manual monitoring of vast amounts of image data is impractical, automated ML systems are essential for detecting fire patterns efficiently. ML models excel at pattern recognition, making them ideal for identifying wildfire-related anomalies. While this research focuses on image-based detection, ML techniques can also enhance sensor-based detection by improving anomaly recognition in environmental data.

F. Existing Approaches

Several studies have investigated different ML techniques for wildfire detection, primarily using Convolutional Neural Networks (CNNs). One approach employs Temporal Feature Extraction, which captures movement in images—such as flames shifting against a static background—and feeds these features into a CNN model. This technique achieved 93% accuracy in wildfire detection [1]. Another study analyzed raw image pixels without additional preprocessing, aside from standard data cleaning. The CNN model trained on this dataset achieved an F1 score of 81% [7]. A third study implemented data augmentation techniques to enhance wildfire detection by analyzing color, texture, and shape features in images. One approach involved converting RGB images to grayscale and extracting histogram-based features, such as the mean and standard deviation of brightness and the probability distribution of gray values. This study utilized both Convolutional Neural Networks (CNNs) and Support Vector Machines (SVMs), achieving an accuracy of 97.55% [4]. One study specifically focusing on wildfire detection using grayscale images combined dynamic threshold grayscale segmentation with ResNet transfer learning to enhance accuracy while reducing training time. This approach achieved 98.45% accuracy with a training loss of 0.04, using 193 iterations, a batch size of 32, and the SGD optimizer. Comparative experiments demonstrated its superiority over ResNet and VGGNet models in accuracy, training loss, and FPS, improving detection efficiency [5]. However, a limitation of this study is the absence of reported precision and recall metrics, which are crucial for evaluating false positives and false negatives.

III. Proposed Approach

A. VGG

In this study, we leverage VGG16-based feature extraction to enhance wildfire detection through image classification. The proposed approach involves processing images using two distinct methods: raw pixel extraction and deep feature extraction using a pre-trained VGG16 model. By comparing these methods, we aim to evaluate the effectiveness of deep feature representations in distinguishing fire-related patterns from other visual elements.

Initially, fire and no-fire images are loaded and converted into numerical representations. The images are resized to 224×224 pixels to maintain compatibility with VGG16's architecture. Instead of using raw pixel values directly, features are extracted from the last convolutional layers of VGG16, where meaningful spatial and texture information is retained. These extracted feature vectors are then flattened into a one-dimensional format, enabling them to be used as inputs for machine learning classifiers.

The extracted feature vectors are standardized to normalize variations across different images, which enhances model performance. The processed data is then fed into machine learning models such as Random Forest and Logistic Regression to assess their effectiveness in wildfire detection.

By incorporating deep feature extraction rather than relying solely on raw pixel values, this approach aims to improve detection accuracy and minimize false positives. Additionally, standardizing the extracted features helps ensure robust and generalizable model performance, making the system more reliable for real-world wildfire monitoring applications.

B. PCA

To further enhance the feature representation and clustering of fire and non-fire images, Principal Component Analysis (PCA) is employed to reduce the dimensionality of the extracted feature vectors. Given that the raw feature maps from VGG16 contain a high number of dimensions, PCA is applied to retain the most significant components while reducing redundancy. In this study, the number of principal components is set to 5, effectively compressing the feature space while preserving critical information required for classification. This dimensionality reduction not only improves computational efficiency but also helps mitigate the curse of dimensionality, which can negatively impact clustering performance.

Following PCA transformation, the dataset undergoes t-distributed Stochastic Neighbor Embedding (t-SNE) visualization to better understand the underlying cluster structure. t-SNE is a nonlinear dimensionality reduction technique that maps high-dimensional data into a two-dimensional space for visualization. The transformed data is subsequently clustered using the K-Means algorithm, with the number of clusters set to 6, derived from Silhouette score analysis. K-Means assigns each image to a cluster based on feature similarities, and the resulting cluster assignments are visualized using t-SNE scatter plots. The use of seaborn visualization techniques aids in the interpretation of cluster separability and relationships among fire and non-fire images.

C. k-means Cluster Center Distances

In this project, K-Means clustering is also utilized for training, with distances between cluster centers serving as distinguishing features for classification. K-Means operates by grouping data points into k clusters, where each cluster is represented by a center point that is iteratively updated until convergence. These cluster centers summarize the feature space, capturing essential patterns in the data. Instead of using raw features directly for classification, the distances of each data point from the cluster centers are used as input features, a technique that draws inspiration from the Bag of Visual Words (BoVW) approach in image processing.

Once the transformed features are obtained, Random Forest and Logistic Regression classifiers are trained to differentiate between fire and non-fire images. The Random Forest classifier, an ensemble learning method, constructs multiple decision trees and aggregates their predictions for improved robustness. On the other hand, Logistic Regression, a linear model, learns a decision boundary that maximizes class separation based on the provided cluster distance features.

The performance of both classifiers is evaluated using accuracy scores, classification reports, and ROC (Receiver Operating Characteristic) curves. The ROC curve provides a visual representation of the trade-off between the true positive rate and false positive rate, with the area under the curve (AUC) serving as a key performance metric. The results indicate that Logistic Regression outperforms Random Forest in both accuracy and ROC-AUC scores, highlighting the effectiveness of using cluster distances as features. Additionally, a misclassification analysis is performed by identifying test images that were incorrectly classified, allowing for a deeper examination of model weaknesses. The final ROC curves are plotted for all models, providing a comparative view of their discriminative ability.

D. RGB and Grayscale Images

Building upon the previous methodology, implements upon both RGP and grayscale images for feature extraction and classification. While color images provide richer feature representations, grayscale images can enhance model generalization by removing redundant color-based variations. To extract meaningful features from grayscale images, the VGG16 model is employed with an adjusted input format where single-channel grayscale images are replicated across three channels to match the model's expected input dimensions whereas RGB by nature contains three channels. The extracted deep features serve as the foundation for further dimensionality reduction and clustering.

PCA is applied to the both the RGB and grayscale feature set to reduce dimensionality while retaining key information. This ensures that the computational complexity remains manageable and that the primary structural differences between fire and non-fire images are preserved. The reduced feature set is then used for K-Means clustering, where each image is assigned to a specific cluster. Additionally, the distances from each image to the identified cluster centers are computed, forming an alternative feature representation.

For classification, Random Forest and Logistic Regression models are trained using the transformed grayscale feature representations. The classification performance is assessed using accuracy scores, classification reports, and ROC-AUC metrics. By comparing the performance of grayscale-based and color-based models, insights can be gained into the importance of color information in fire detection. The results from both approaches are analyzed through ROC curve visualizations, enabling a comprehensive evaluation of their effectiveness in distinguishing between fire and non-fire instances.

IV. Data

A. Data exploration

The dataset used in this research focuses on classifying wildfire occurrences using labeled images categorized as either fire or no-fire. These images are sourced from publicly available datasets, such as those found on Kaggle [8], and are divided into training and testing sets to facilitate model training and evaluation. The training dataset includes 928 images of wildfire incidents and 904 images without fire, totaling 1,832 samples. The testing dataset comprises 22 fire images and 46 no-fire images, resulting in 68 samples for validation.

To standardize the dataset and improve feature extraction, all images are resized to a consistent resolution of 128×128 pixels. Pixel values are normalized between 0 and 1 to enhance model training efficiency, and color formats are converted from BGR to RGB for accurate representation. Feature extraction techniques, such as raw pixel values, are employed to analyze color distributions, which are essential for distinguishing fire from non-fire images. A CNN network is used to extract features and which is further K-Means clustering is also applied to explore patterns and identify distinct feature groupings within the dataset.

An analysis of the dataset highlights certain challenges, including an imbalance in the number of training and testing samples, which may impact model generalization. Real-world complexities such as lighting variations, smoke interference, and diverse environmental backgrounds may affect model accuracy. Despite these challenges, this dataset serves as a valuable resource for training machine learning models aimed at detecting and classifying wildfire occurrences. This research contributes to the advancement of automated wildfire detection, supporting early intervention and mitigation strategies. Future work may focus on expanding the dataset and incorporating additional environmental variables such as temperature, wind speed, and humidity to enhance prediction accuracy and robustness.

B. Preprocessing

In this phase, a transfer learning approach is applied using the VGG16 model, a convolutional neural network pre-trained on the ImageNet dataset. Instead of using the model for classification, its convolutional layers are utilized to extract high-level image features while bypassing the classification head. To ensure consistency in input dimensions, all images are resized to 224x224 pixels, the required input size for VGG16.

The preprocessing process begins with loading images from the specified directories. Depending on the input mode, images are loaded either in color or grayscale. For grayscale images, the single-channel input is expanded to three channels by replicating the grayscale values across all channels, ensuring compatibility with the VGG16 model, which expects three-channel images. After resizing and converting to arrays, images undergo VGG16-specific preprocessing, adjusting pixel values to match the input distribution of the pre-trained model.

At this stage, images are ready for feature extraction. Instead of directly classifying the images, the VGG16 model will be leveraged to extract meaningful high-level representations, which will serve as input for further machine learning tasks.

C. Feature Construction

1) Feature Extraction: Building on the previous phase, the next step focuses on extracting meaningful features from the preprocessed images using the VGG16 model. By leveraging its ability to recognize patterns and structures, the model converts raw image data into compact, informative representations suitable for machine learning.

Each image is passed through the VGG16 model to extract feature maps from its convolutional layers, capturing essential visual characteristics while reducing the dimensionality of the original images. These extracted features are then flattened into one-dimensional vectors, making them compatible with machine learning models. The structured dataset is created by pairing these feature vectors with their corresponding labels, 1 for "fire" and 0 for "nofire." The training set and test set are then formed by combining feature vectors from both categories.

To further refine the feature space, Principal Component Analysis (PCA) is applied. This technique reduces the dimensionality of the data while preserving as much variance as possible, making the dataset more manageable and potentially enhancing model performance. Since the images are loaded into VGG which requires an image shape of 224 x 224 x 3, this means each image has 15,0528 features. Using PCA, we iteratively reduced the dimensionality down until the silhouette score of the K-Means clusters began to improve. Using this method, we reduced the dimensionality of the feature set to 5.

2) Feature Analysis & Clustering: To gain deeper insights into the separability of "fire" and "nofire" images, t-Distributed Stochastic Neighbor Embedding (t-SNE) is employed. This technique visualizes the high dimensional feature space in a two-dimensional scatter plot, revealing how well the extracted features distinguish between the two classes. Understanding the distribution of these clusters helps assess the effectiveness of feature extraction and guides further improvements in model training and classification.

Additionally, cluster evaluation techniques are used to determine the optimal number of clusters for the K-Means clustering algorithm. The goal is to identify a value of k (the number of clusters) that balances model complexity with the quality of clustering, ensuring that the images are grouped meaningfully.

Two common methods are used for cluster evaluation: the Elbow Method and the Silhouette Score (see Figure 2).

The Elbow Method relies on the inertia, which is the sum of squared distances between each sample and its closest cluster

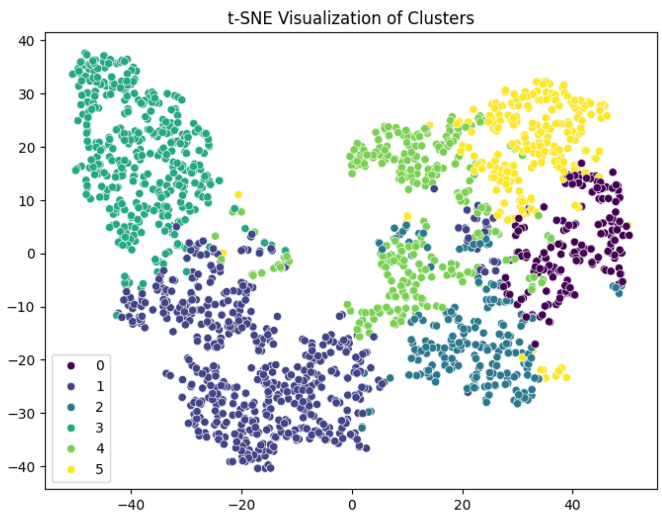

Fig. 1. tSNE Visualization of clusters

center. As the number of clusters increases, inertia tends to decrease because the clusters become smaller and more tightly packed. The idea is to plot the inertia values against different values of k and identify the point where the curve starts to flatten or bend—this is referred to as the ”elbow.” The point at which inertia begins to decrease more slowly indicates the optimal k, as increasing the number of clusters beyond this point offers diminishing returns in terms of variance explained.

The Silhouette Score measures how similar an object is to its own cluster compared to other clusters. It provides a metric of cluster cohesion and separation, with higher scores indicating better-defined clusters. The score ranges from -1 to 1, where values close to 1 suggest that the samples are well-clustered. By plotting the silhouette scores for different values of k, one can identify the peak of the curve, which suggests the optimal number of clusters where the samples are most cohesively grouped while remaining well-separated from other clusters.

The code iterates over a range of k values from 2 to 10, calculating both the inertia and silhouette scores for each k. The results are plotted on two separate graphs: one showing the elbow plot of inertia and the other displaying the silhouette score. These plots help in visualizing the trade-off between the number of clusters and the quality of the clustering, ultimately guiding the choice of the best k for the dataset.

The Elbow Method does not display a distinct bend which which we would look for to help determine the best k value. However, the Silhouette score does show a distinct peak at K = 6 with a score of around 0.315. While this score would ideally be closer to 1.0, this is the best score we were able to generate after PCA dimensionality reduction which dramatically improved the score. As such, k is set to 6 for all subsequence uses of K-Means.

V. Results

The result section is divided into two sections the first one is the RGB images where the process keep all the pixels information which includes their Red-Green-Blue values. The second section is transforming the images color pixels into a gray-scale images.

Tables show the metrics gathered for a Random Forest and Logistic Regression model trained on either the raw pixel data of the images, the K-Means cluster labels and K-means cluster distances from center data. Accuracy measures the overall correctness of the model and is calculated as (TP + TN) / (TP + TN + FP + FN). Precision, calculated as TP / (TP + FP), measures the correctness of positive predictions, ie how many images that the model classifies as fire are actually fire. Recall, calculated as TP / (TP + FN), measures the model's ability to identify all actual positive cases (i.e. how many of the fire images does is classify correctly).

A. RGB Images

1) Raw pixels: The classification results using raw RGB pixel values demonstrate that the Random Forest model achieves perfect classification performance, with 100% accuracy, precision, recall, and AUC for both ”Fire” and ”No Fire” categories, as shown in Table I. This suggests that the model is highly effective at distinguishing fire from non-fire instances when using raw pixel values.

In contrast, Logistic Regression, while still performing well, shows slightly lower accuracy and recall for fire detection. It achieves 98% accuracy overall, with a 100% precision for fire cases but a recall of 95%, indicating that some fire instances are misclassified as ”No Fire.” Despite this, its AUC remains perfect (1.00), suggesting strong discriminative ability overall.

These results highlight the effectiveness of Random Forest in handling high-dimensional image data, leveraging its ensemble approach to capture complex patterns. However, even given that a lot of tests were made, the perfect performance may indicate potential overfitting. An important mention is that the number of estimators for the random forest was set to 100 which is low given the large number of features, meaning that it is unlikely overfitting.

Random Forest	Accuracy	Precision	Recall	AUC
No Fire	1.00	1.00	1.00	1.00
Fire	1.00	1.00	1.00	1.00
Logistic Regression				
No Fire	0.98	0.98	1.00	1.00
Fire	0.98	1.00	0.95	1.00

TABLE I
Raw Pixels Training Metrics

2) K-means Clustering: Applying K-Means clustering as a preprocessing step improves Random Forest's performance, leading to an increase in accuracy and a better trade-off between precision and recall (Table II). However, Logistic Regression suffers a noticeable decline in accuracy, primarily due to the transformation limiting its ability to distinguish between classes. This suggests that while K-Means clustering enhances Random Forest's decision-making, it negatively impacts the linear classification capability of Logistic Regression. The AUC values confirm this trend, with Random Forest

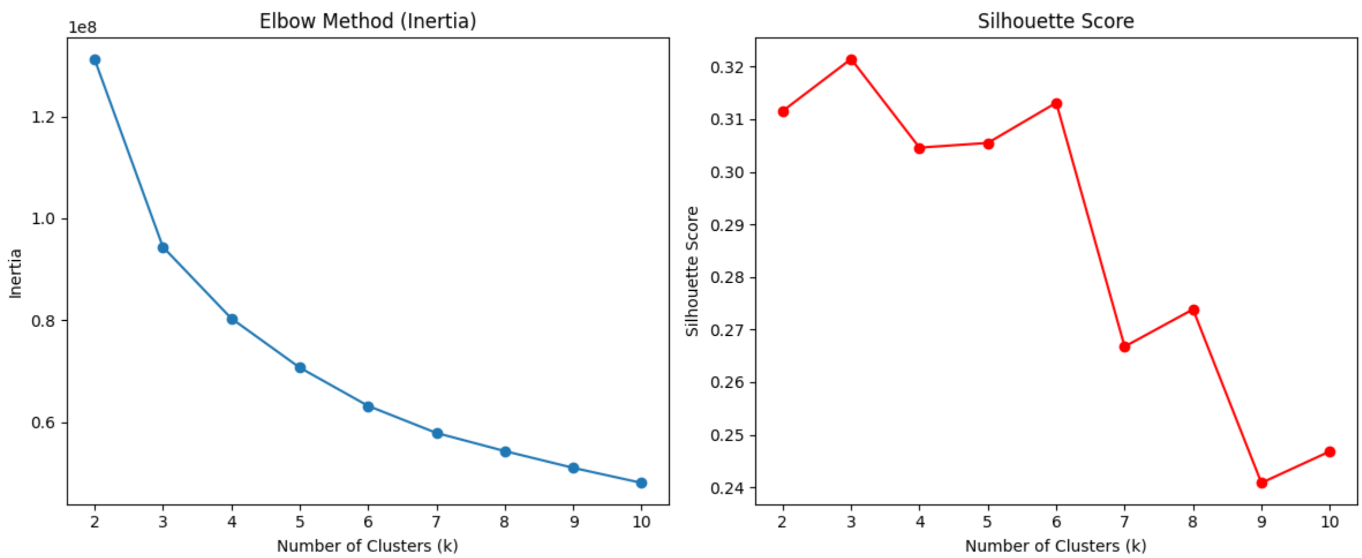

Fig. 2. Elbow and Silhouette Scores

Fig. 3. Visualization of K-Means Clustering Applied to Images

maintaining a strong 0.95 while Logistic Regression drops to 0.85.

3) K-Means Clustering using distances between cluster centers: Using distances between cluster centers as input features significantly improves classification performance for both models, as shown in Table III. Random Forest achieves 97% accuracy, demonstrating strong precision and recall across both the “Fire” and “No Fire” classes. This indicates that Random Forest effectively leverages the transformed feature space to distinguish between the two categories.

Random Forest	Accuracy	Precision	Recall	AUC
No Fire	0.94	0.96	0.96	0.95
Fire	0.94	0.91	0.91	0.95
Logistic Regression				
No Fire	0.73	0.85	0.74	0.85
Fire	0.73	0.57	0.73	0.85

TABLE II
K-MEANS LABEL TRAINING METRICS

Furthermore, Logistic Regression performs really well as it achieves 98% accuracy with perfect recall for fire detection. This suggests that reducing the input dimensionality by summarizing pixel-level information through cluster distances makes it easier for models to identify meaningful patterns, allowing even a simpler linear model to achieve high performance.

Random Forest	Accuracy	Precision	Recall	AUC
No Fire	0.97	0.98	0.98	0.99
Fire	0.97	0.95	0.95	0.99
Logistic Regression				
No Fire	0.98	1.00	0.98	1.00
Fire	0.98	0.96	1.00	1.00

TABLE III
K-MEANS DISTANCES TRAINING METRICS

B. Gray scale Images

1) Raw pixels: Using grayscale images with raw pixel values results in strong classification performance for both Random Forest and Logistic Regression, as shown in Table IV. Random Forest achieves 97% accuracy with high precision and recall for both ”Fire” and ”No Fire” classes, demonstrating its ability to effectively separate the two categories. Similarly, Logistic Regression achieves identical performance, indicating that grayscale preprocessing does not negatively impact classification. These results show that grayscale preprocessing does not significantly degrade model performance and may even enhance classification consistency compared to RGB images.

Random Forest	Accuracy	Precision	Recall	AUC
No Fire	0.97	0.98	0.98	0.99
Fire	0.97	0.95	0.95	0.99
Logistic Regression				
No Fire	0.97	0.98	0.98	0.99
Fire	0.97	0.95	0.95	0.99

TABLE IV
GRAY-SCALE: RAW PIXELS TRAINING METRICS

2) K-means Clustering: Applying K-Means clustering as a preprocessing step negatively impacts classification performance, particularly for Logistic Regression (Table V). While Random Forest maintains an accuracy of 88%, its precision for the "Fire" class decreases, indicating some misclassification of fire instances. However, the most significant decline is observed in Logistic Regression, which drops to 38% accuracy, with notably low recall for the "No Fire" class. The clustering approach likely obscures critical pixel intensity variations, reducing the ability of a simpler model like Logistic Regression to learn meaningful decision boundaries. The AUC values highlight this decline, with Random Forest maintaining a reasonable 0.96 while Logistic Regression drops significantly to 0.75.

Random Forest	Accuracy	Precision	Recall	AUC
No Fire	0.88	0.95	0.87	0.96
Fire	0.88	0.77	0.91	0.96
Logistic Regression				
No Fire	0.38	0.62	0.22	0.75
Fire	0.38	0.31	0.73	0.75

TABLE V
GRAY-SCALE: K-MEANS LABEL TRAINING METRICS

3) K-Means Clustering using distances between cluster centers: When using distances between K-Means cluster centers as features, both Random Forest and Logistic Regression demonstrate strong performance (Table VI). Both models achieve an accuracy of 95%, which shows how this approach effectively simplifies the feature space while maintaining essential information for classification.

The models show high performance, with precision rates of 96% for "No Fire" and 95% for "Fire", along with recall values of 98% for "No Fire" and 91% for "Fire". These results indicate a reliable classification for both categories. Additionally, the AUC values for Random Forest (0.98) and Logistic Regression (0.99) suggest that both models are effective at distinguishing between the "Fire" and "No Fire" labels. Overall, these results highlight that using cluster distances as features can improve the model's ability to generalize and classify grayscale image data effectively.

C. Main Takeaways

The results highlight key differences in performance across various preprocessing techniques and classification models:

Random Forest	Accuracy	Precision	Recall	AUC
No Fire	0.95	0.96	0.98	0.98
Fire	0.95	0.95	0.91	0.98
Logistic Regression				
No Fire	0.95	0.96	0.98	0.99
Fire	0.95	0.95	0.91	0.99

TABLE VI
GRAY-SCALE: K-MEANS DISTANCES TRAINING METRICS

- **Raw RGB Pixels:** Random Forest achieves perfect classification (100% accuracy, precision, recall, and AUC), while Logistic Regression performs slightly worse, with a 98% accuracy and a recall of 95% for fire detection. This suggests that Random Forest is highly effective for raw pixel data but may be prone to overfitting.
- **RGB K-Means Clustering:** While Random Forest maintains strong performance (94% accuracy), Logistic Regression struggles, dropping to 73% accuracy. This indicates that K-Means clustering benefits non-linear models but negatively impacts simpler linear models.
- **RGB K-Means Cluster Distances:** Both models improve significantly with this transformation, with Random Forest reaching 97% accuracy and Logistic Regression reaching 98%, suggesting that summarizing pixel data into cluster distances enhances interpretability and classification.
- **Raw Grayscale Pixels:** Both models perform equally well, achieving 97% accuracy. This demonstrates that grayscale preprocessing does not degrade performance and may enhance model consistency.
- **Grayscale K-Means Clustering:** Classification performance declines, particularly for Logistic Regression, which drops to 38% accuracy. This suggests that clustering obscures essential intensity-based features, making classification harder for linear models.
- **Grayscale K-Means Cluster Distances:** Both models achieve 95% accuracy, indicating that this transformation preserves essential features while simplifying the input space, making classification more effective.

Overall, using K-Means cluster distances as features consistently enhances performance, especially for Logistic Regression, while raw pixel data benefits Random Forest the most.

VI. CONCLUSION AND FUTURE WORK

In this study, we evaluated various approaches to wildfire classification using both traditional machine learning models and preprocessing techniques, emphasizing the strengths and limitations of different feature representations. Raw pixel values served as a strong baseline, with Random Forest consistently outperforming Logistic Regression in terms of accuracy, precision, recall, and AUC. While K-Means clustering as a preprocessing step improved Random Forest's performance, it adversely affected Logistic Regression, likely due to its inability to capture pixel-level details in a linear model. However, utilizing distances between K-Means cluster centers significantly enhanced both models, improving their ability to distinguish between "Fire" and "No Fire" categories.

Furthermore, grayscale images delivered comparable performance to RGB images, suggesting that color information was not critical for effective wildfire detection. Overall, Random Forest proved to be the most reliable model, with K-Means distance features providing the best balance of classification accuracy, precision, and recall. Logistic Regression, while performing well, benefited from feature reduction techniques like K-Means clustering and its distance-based transformations.

These findings highlight that traditional machine learning models, when coupled with feature engineering techniques such as K-Means clustering and dimensionality reduction, can achieve competitive performance relative to deep learning-based approaches like CNNs. Future research could focus on testing different clustering techniques, such as DBSCAN or hierarchical clustering, to assess if they can capture non-linear patterns in the image data and potentially improve performance, particularly for Logistic Regression. In addition, exploring real-time detection is a critical area, as the ultimate goal is to identify the best model for wildfire detection in real-time. Evaluating these models in a real-world, real-time setting would offer valuable insight into their effectiveness and provide a clearer understanding of their potential for operational use in wildfire detection.

References

[1] *A Deep Learning-Based Experiment on Forest Wildfire Detection in Machine Vision Course — IEEE Journals & Magazine — IEEE Xplore*. URL: https://ieeexplore.ieee.org/abstract/document/10083121 (visited on 02/02/2025).

[2] Robert S. Allison et al. "Airborne Optical and Thermal Remote Sensing for Wildfire Detection and Monitoring". In: *Sensors* 16.8 (Aug. 2016). Number: 8 Publisher: Multidisciplinary Digital Publishing Institute, p. 1310. ISSN: 1424-8220. DOI: 10.3390/s16081310. URL: https://www.mdpi.com/1424-8220/16/8/1310 (visited on 02/03/2025).

[3] Abdelmalek Bouguettaya et al. "A review on early wildfire detection from unmanned aerial vehicles using deep learning-based computer vision algorithms". In: *Signal Processing* 190 (Jan. 1, 2022), p. 108309. ISSN: 0165-1684. DOI: 10.1016/j.sigpro.2021.108309. URL: https://www.sciencedirect.com/science/article/pii/S0165168421003467 (visited on 02/02/2025).

[4] *Deep Learning-Based Wildfire Image Detection and Classification Systems for Controlling Biomass - Prakash - 2023 - International Journal of Intelligent Systems - Wiley Online Library*. URL: https://onlinelibrary.wiley.com/doi/full/10.1155/2023/7939516 (visited on 02/02/2025).

[5] Hai Li and Peng Sun. "Image-Based Fire Detection Using Dynamic Threshold Grayscale Segmentation and Residual Network Transfer Learning". In: *Mathematics* 11.18 (Jan. 2023). Number: 18 Publisher: Multidisciplinary Digital Publishing Institute, p. 3940. ISSN: 2227-7390. DOI: 10.3390/math11183940. URL: https://www.mdpi.com/2227-7390/11/18/3940 (visited on 03/05/2025).

[6] Ankita Mohapatra and Timothy Trinh. "Early Wildfire Detection Technologies in Practice—A Review". In: *Sustainability* 14.19 (Jan. 2022). Number: 19 Publisher: Multidisciplinary Digital Publishing Institute, p. 12270. ISSN: 2071-1050. DOI: 10.3390/su141912270. URL: https://www.mdpi.com/2071-1050/14/19/12270 (visited on 02/03/2025).

[7] Dmitry Rashkovetsky et al. "Wildfire Detection From Multisensor Satellite Imagery Using Deep Semantic Segmentation". In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 14 (2021). Conference Name: IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, pp. 7001–7016. ISSN: 2151-1535. DOI: 10.1109/JSTARS.2021.3093625. URL: https://ieeexplore.ieee.org/abstract/document/9468971 (visited on 02/03/2025).

[8] *Wildfire Detection Image Data*. URL: https://www.kaggle.com/datasets/brsdincer/wildfire-detection-image-data (visited on 02/12/2025).