Computer Vision-Based Wildfire Detection in Video: Deep Learning Using Motion Estimation

BY

FATEMEH TAGHVAEI
B.Sc. in Computer Engineering, University of Isfahan, Iran, 2021

THESIS

Submitted as partial fulfillment of the requirements
for the degree of Master of Science in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Chicago, 2024

Chicago, Illinois

Defense Committee:

Ahmet Enis Cetin, Chair and Advisor
Jim Kosmach
Erdem Koyuncu

To myself,

the only person worthy of my company

ACKNOWLEDGMENTS

I would like to thank my advisor Professor Ahmet Enis Cetin for his support.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

CHAPTER PAGE

TABLE OF CONTENTS (Continued)

CHAPTER PAGE

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

UAV	Unmanned Aerial Vehicles
GPS	Global Positioning System
NASA	National Aeronautics and Space Administration
NOAA	National Oceanic and Atmospheric Administration
YOLO	You Only Look Once
CD	Change Detection
ME	Motion Estimation

SUMMARY

To address the significant issue of false alarms in computer vision-based wildfire detection systems, this study investigates the use of motion estimation to improve accuracy. Using a single neural network for wildfire detection presents challenges, particularly in misclassifying fog or cloud as smoke due to their visual similarities. The incorporation of the Gunnar-Farneback optical flow for motion estimation after initial wildfire detection produced promising results on non-wildfire videos, reducing the average false alarm rate from 5.69% to 0.40% while maintaining an acceptable true detection rate for wildfire videos. The integration of motion estimation, specifically using the Gunnar-Farneback optical flow approach, demonstrates a robust strategy in minimizing false alarms in non-wildfire videos while maintaining an acceptable true detection rate for wildfire videos.

CHAPTER 1

INTRODUCTION

Wildfire detection provides many advantages across different areas, including environmental preservation, human safety, and economic savings. Early detection is crucial for preventing fires from spreading uncontrollably, which helps protect natural ecosystems and biodiversity. Wildfires can destroy vegetation and wildlife, disrupt ecosystems, and reduce the aesthetic value of natural landscapes. Early detection can minimize these adverse effects and help preserve ecosystem resilience and sustainability.

In terms of human safety, early detection is vital for saving lives and preventing injuries by enabling timely warnings and evacuations in populated areas. Quick identification allows communities enough time to evacuate safely and avoid harm. Economically, early detection helps reduce property damage and the financial burden of firefighting efforts.

Technological advancements, such as computer vision and deep learning-based detection, provide proactive and efficient wildfire detection [2].

1.1 Problem Statement

Computer vision-based wildfire detection systems play a crucial role in mitigating the impact of wildfires through early detection [2]. A significant challenge arises in accurately differentiating between genuine fire, smoke, and environmental elements such as fog and clouds, leading to occasional false alarms and affecting the reliability of these systems.

The main issue comes from the visual similarity between smoke, fog, and clouds, making it difficult for the system to accurately identify actual fires. Previous studies show situations where alarms were triggered in the presence of fog and clouds, affecting the accuracy of wildfire detection systems [3, 4].

Relying solely on visual data introduces complexities that lead to false positives and unnecessary panic and response efforts. Ensuring the system's reliability is essential, as it minimizes false alarms and helps maintain trust in the detection system.

False alarms in wildfire detection systems create problems beyond bothering communities. They put pressure on emergency response resources and divert attention away from real threats, slowing down responses to actual emergencies. Frequent false alarms can lead to complacency among community members, making them less likely to respond urgently when real danger arises.

Unnecessary emergency responses also bring financial costs for public agencies, individuals, and firefighters. Misuse of resources can affect the budgets of firefighting agencies and individuals, impacting their ability to acquire essential equipment and maintain infrastructure. The economic burden of false alarms highlights the importance of accurate wildfire detection systems.

Advancing the accuracy and reliability of computer vision-based wildfire detection systems by refining algorithms is essential for effectively safeguarding communities and natural environments. Addressing the challenge of distinguishing fire and smoke from fog and clouds is a critical consideration in improving these systems.

1.2 Research Objectives

1.2.1 Address Visual Resemblance Challenges

To advance the system's ability to discern between smoke, fog, and cloud, our primary focus is on upgrading its discriminatory capabilities. This involves investigating and implementing strategies to overcome the challenges posed by the visual resemblance among these environmental elements. We aim to minimize false alarms and improve the precision of wildfire detection. To achieve this, we plan to explore innovative approaches that contribute to better discrimination. Additionally, we intend to implement these approaches in the system, enhancing its overall efficiency in accurately identifying different environmental elements while avoiding unnecessary alerts.

1.2.2 Enhance System Reliability

A crucial aspect of our research is finding the right balance between accurately detecting real fires and minimizing false alarms. Our system needs to maintain its effectiveness in identifying genuine fires (true detection rate) while also reducing instances of false alarms. Achieving this equilibrium ensures that our improved system consistently provides accurate and dependable alerts. This approach aims to enhance the overall reliability of the system, making it able to distinguish between real fire incidents and other factors that might otherwise trigger unnecessary alarms.

1.3 Limitations

1.3.1 Challenges in Training Data Collection

Gathering suitable training data for our study was challenging due to the lack of existing datasets in this field. We invested significant time and effort in constructing a comprehensive training dataset to enhance the robustness of our study. This involved collecting various fire and non-fire instances through different scenarios and conditions.

A key challenge was acquiring pairs of images for effective training. We relied on videos, particularly those from stationary cameras, to generate these image pairs. However, the limited availability of videos depicting specific scenarios, such as wildfires and fog, added difficulty.

1.3.2 Testing Data Limitation

Focusing exclusively on stationary cameras posed challenges in obtaining a diverse testing dataset, which may affect the broader applicability of the findings, especially for dynamic camera setups. While essential for certain study parameters, this focus also constrained the scope of the testing data. This constraint raises questions about the system's adaptability to dynamic camera setups and the need for further investigation. Additional research into scenarios involving cameras with dynamic movements is crucial for expanding our understanding of the system's performance in real-world circumstances.

1.3.3 Frame Rate Discrepancies

Managing varying frame rates in our collected videos presented a technical challenge that demanded careful consideration. To maintain consistency and enable fair comparison, we found it necessary to normalize the frame rates across all videos. This process of normalizing frame rates was crucial for ensuring coherence and meaningful analysis of our data. However, it added an extra layer of complexity to the data preparation process. Fundamentally, the importance of this step lies in the fact that having a uniform frame rate is essential not only for creating a coherent dataset but also for conducting accurate testing of our models. The uniformity in frame rates ensures that the videos are comparable, allowing for reliable dataset construction and robust testing, contributing to the overall reliability and validity of our study.

1.4 Organization of The Thesis

In Chapter 2, we offer essential background information to aid readers in understanding the research, along with a review of related work in the field. Chapter 3 presents a novel wildfire detection approach based on the Gunnar-Farneback optical flow as a part of a deep neural network framework. In Chapter 4, we introduce a wildfire detection system employing change detection. Furthermore, we our findings in Chapter 5 reveal that integrating motion estimation, particularly the Gunnar-Farneback optical flow, into the neural network-based system significantly reduces false alarms. However, we observe that change detection is not a good choice as it adversely affects the true detection rate of the system.

CHAPTER 2

BACKGROUND

In this chapter, we review wildfire detection technologies, including sensor nodes, UAVs, camera networks, and satellite surveillance. We offer a brief overview of each technology and discuss their respective advantages and disadvantages. We also explore existing active-fire management algorithms, such as machine learning techniques that involve supervised and unsupervised learning methods, as well as deep learning approaches like classification, segmentation, and object detection.

2.1 Early Wildfire Detection Technologies

The evolution of early wildfire detection in the United States from the establishment of fire lookout towers in the early 1900s to the adaptation of modern technologies reflects a continuous commitment to improving wildfire management. The initial use of fire lookout towers, manned by "fire spotters" equipped with heliographs, marked a crucial step in detecting and responding to wildfires promptly. Over time, as technology advanced, the reliance on traditional fire towers diminished.

The shift towards more advanced technologies, such as temperature sensor nodes, Unmanned Aerial Vehicles (UAV)s, camera networks, and satellites demonstrates an effort to enhance and modernize the early wildfire detection system. These technologies build upon the foundation laid by the fire lookout towers, aiming to provide more accurate and timely information about potential wildfires. By identifying wildfires in their early stages and monitoring the spread of fires, these modern tools contribute to a more proactive approach to firefighting [5].

2.1.1 Sensor Nodes

Sensor nodes play a crucial role in wildfire detection, consisting of low-power sensors such as humidity, temperature, and gases [6]. They are placed to monitor the environment for potential fires. When configured as a wireless sensor network, these nodes communicate wirelessly and rely on solar energy and rechargeable batteries for sustained operation. In case of a fire, algorithms are employed to analyze the data from sensors, distinguishing the rapid changes in environmental variables indicative of a wildfire. Researchers have developed various algorithms, such as the low-complexity algorithms proposed by Varela et al, which utilize temperature and humidity data for wildfire detection [7]. Other approaches include machine learning algorithms, fuzzy logic controllers, and Bayesian classifiers, each utilizing different sensor inputs for accurate fire identification [5, 8–10].

Sensor nodes stand out as an economical and user-friendly technology for wildfire detection. However, they cannot cover huge areas and they are useful for protecting important infrastructure. Their straightforward design facilitates easy customization and attracts a diverse user base. Notably, the energy-efficient nature of sensor nodes simplifies power supply challenges, making them cost-effective and efficient choices for extended surveillance. While sensor nodes offer advantages for wildfire detection, deploying them encounters challenges, particularly in remote areas. Initial deployment requires human effort or sophisticated methods. Once installed, sensor nodes can work independently for extended periods. However, their susceptibility to heat during fires presents maintenance challenges, and careful planning for retrieval and replacement [5].

2.1.2 Unmanned Aerial Vehicles

Unmanned Aerial Vehicles (UAVs), commonly known as operated and travel by-flight systems, play a crucial role in wildfire detection by providing real-time, accurate data from inaccessible vantage points. The mobility of UAVs allows for rapid visual monitoring of fire progression. Originally comparable to consumer drones, UAVs now operate within an extended altitude range of up to 30,000 feet [11, 12]. The collected data include GPS locations, images, and videos [13]. These systems can be remotely controlled by humans or automated, creating users with limited experience. Automated UAVs, once limited to basic surveillance, now handle complex tasks, facilitating quick and organized responses to wildfires [14].

UAVs offer flexibility in surveillance, dynamically adjusting to areas of interest, they provide high resolution in high-risk zones, actively tracking fire spread from a safe distance. The efficiency surpasses static networks, making them cost-effective compared to satellites and adapting to identifying smaller wildfires. UAVs often require human involvement, presenting a notable challenge. The critical issues include limited flight duration, especially for hand-launched LALEs, making them unsuitable for continuous surveillance in wildfire regions [5].

2.1.3 Camera Networks

Camera networks, comprising interconnected cameras equipped with advanced features, are employed for fire monitoring in wide areas. In early camera networks, videos were broadcast to a control center and manually scanned for signs of fire. In modern setups, cameras are integrated with AI and communication servers to optimize surveillance. They can be deployed in urban or remote locations, offering early detection and prediction of fires. AI integration enables autonomous identification of fire indications, reducing reliance on human-operated watch towers. Internet access allows communication and real-time broadcasting of videos to notify users of significant events [15–19].

Camera networks offer advantages such as autonomy and increased distance from potential fire sites, reducing the risk of damage. Their placement is simplified compared to sensor networks and UAVs due to longer range and straightforward operation. This contrasts with sensor nodes, which have lower coverage and various short-range sensors. The extensive coverage of stationary cameras partly addresses the challenge of continuous power requirements, limiting deployment in extremely remote areas. The initial setup of camera towers demands significant resources, potentially making it economically unfeasible in certain locations. The positioning of cameras also restricts the inclusion of sensors, limiting potential enhancements, such as equipping cameras with temperature of gas sensors, which are ineffective due to the cameras' design for distant operation from the target area [5].

2.1.4 Satellite Surveillance

NASA and NOAA pioneered wildfire observation using polar orbiting and geostationary satellites. While polar satellites provide global monitoring, they have a low temporal sampling rate. Geostationary satellites offer higher temporal data for specific areas but lack global environmental monitoring. Instruments on these satellites capture diverse data types, including wildfire detection and monitoring pollutant transport. Processing data for small wildfires is challenging due to lower spatial resolution. Distinguishing smoke from clouds is also a difficulty [5].

Satellite provides a wide-ranging perspective on fire behavior by surveying large land areas in a single sweep. Their long operational lifespan and multi-purpose capabilities extend beyond fire detection, including tracking oceanic currents and monitoring climate changes. Meteorological satellites from NASA and NOAA contribute to understanding environmental processes, offering extensive and publicly available data that have driven discoveries in various disciplines. A key challenge in wildfire observation with satellites is optimizing coverage area and data collection frequency. High-flying altitudes limit the identification of smaller fires. Dense clouds and smoke cover impede ground visibility, affecting the tracking of active fires [20].

2.2 Active-Fire Management Algorithms

Detecting wildfires through computer vision involves leveraging machine learning and deep learning algorithms to analyze data and recognize crucial features such as smoke and flames. By training these algorithms on extensive image and video datasets, they acquire the capability to discern these distinctive features. The application of such trained machine learning models to AI systems, such as UAVs, facilitates automated data processing. This, in turn, enables the implementation of early wildfire detection mechanisms. Notably, recent advancements in deep learning algorithms have showcased impressive performance in efficiently handling wildfire detection tasks. Their ability to process vast datasets in real time enhances the overall effectiveness and responsiveness of wildfire detection systems [21, 22].

2.2.1 Machine Learning Techniques

As a subset of artificial intelligence, machine learning plays a crucial role in wildfire management. In the context of wildfire detection, machine learning algorithms offer a dynamic and adaptable approach. By using the information from remote sensing sources like satellite and drone imagery, these algorithms enable real-time monitoring and early detection of wildfires, contributing to efficient response mechanisms. There are two main types of machine learning algorithms: supervised and unsupervised learning. Supervised learning involves using labeled data to teach algorithms how to recognize patterns and make predictions. Specifically for wildfires, this means utilizing data from satellites and drones to detect potential fires in their early stages. On the other hand, unsupervised learning works with unlabeled data. In the context of wildfire management, this type of learning proves valuable for analyzing datasets and revealing hidden relationships or irregularities [23].

2.2.1.1 Supervised Learning

In supervised learning algorithms, there are two main types: classification and regression. Classification assigns input data to categories, while regression predicts numerical values. Both techniques play a crucial role in early wildfire detection. Classification identifies areas prone to fire. On the other hand, regression predicts fire spread and intensity based on factors like temperature and humidity, aiding in quick action to minimize damage.

Supervised learning has several advantages in the context of wildfire-related tasks. These algorithms demonstrate good performance by attaining reasonable accuracy across various wildfire applications. One notable advantage lies in their capacity to fully automate tasks like fire detection and monitoring. Hence, reducing the need for extensive human involvement. Additionally, supervised learning methods offer customization options, enabling adaptation to specific wildfire challenges through precise hyper-parameter tuning. Moreover, these methods ensure consistency in results across diverse wildfire scenarios and possess the capability to scale and adapt to changing environmental circumstances [23].

Supervised learning, despite its advantages, comes with certain drawbacks. Data acquisition is a challenge in supervised methods as their effectiveness heavily relies on the quality and quantity of training data. Biases in the training data can impact the model's performance. Additionally, the resource demand associated with some supervised algorithms is notable, requiring computational power and memory during training. Moreover, a notable risk of over-fitting arises, particularly when models become excessively complex, leading to challenges in effectively generalizing new data. Lastly, the process of data collection for constructing a high-quality dataset involves challenges such as labeling, and ensuring data quality, and diversity which can impact the overall effectiveness of supervised learning methods [23].

Classification methods include, Naive Bayes, Decision Tree, Random Forest, Adaptive Boosting, Bayesian Classifier, Logistic Regression, K-Nearest Neighbors, Support Vector Machine and Artificial Neural Network [23].

Naive Bayes is a probabilistic classifier that can determine the occurrence or absence of wildfires using sensor data [24,25]. Decision Tree is a non-parametric algorithm. It is a valuable algorithm in decision-making and guiding actions through the analysis of environmental factors associated with wildfires [26]. Random Forest is an ensemble learning technique. It can be used to predict various aspects of wildfire, including occurrence, severity, or behavior. It improves reliability and accuracy by training multiple decision trees in a bagged manner. It addresses the limitation of individual decision trees, resulting in more robust fire prediction [27]. Adaptive Boosting is a technique to enhance the performance of classifiers. By effectively integrating classifiers, it significantly improves the accuracy and reliability of wildfire detection techniques [28]. Bayesian Classifier employs Bayesian probability to analyze sensor data, assessing the likelihood of the presence of fire [29]. Logistic Regression is an effective method to analyze the relationship between factors obtained from sensors. It aids in understanding the impact of different variables on the likelihood of fire [30]. K-Nearest Neighbors algorithm operates as a non-parametric method classifying data according to their similarities. In the context of wildfire applications, KNN works well for real-time decision-making [31]. Support Vector Machine is designed with the primary objective of identifying the optimal hyperplane in an N-dimensional space. This hyperplane serves as a critical element in ensuring the robustness of the model, particularly when facing dynamic conditions during wildfires. By seeking the best-fitting hyperplane, SVM contributes to maintaining the efficacy of the model under changing environmental circumstances [32].

2.2.1.2 Unsupervised Learning

In unsupervised learning algorithms, there are two main types: clustering and dimension reduction. Clustering involves separating data points into clusters without prior knowledge about the data. The goal is to ensure that data within the same cluster is highly similar, while data in different clusters is dissimilar. Clustering plays a crucial role in identifying wildfires and facilitating early detection. Dimension reduction is a technique to identify independent features and eliminate irrelevant or redundant ones from the dataset. Dimension reduction addresses issues such as increased computational complexity. Unsupervised learning algorithms have several advantages, primarily derived from their ability to operate without relying on labeled data for training. One notable benefit is the substantial reduction in the required size of sensor data, facilitating more efficient data handling. Additionally, these algorithms exhibit scalability, making them versatile for various wildfire applications while maintaining acceptable performance and avoiding over-fitting issues due to their robustness. Their applicability extends to the effective management of noisy and complex datasets, as the absence of a training process enables quicker data analysis. Furthermore, the unbiased nature of unsupervised algorithms, free from dependence on labeled data, makes them less susceptible to human bias, allowing for the discovery of unexpected patterns within the data. Their reduced computational intensity is also beneficial in scenarios where computational resources are limited [23].

2.2.2 Deep Learning Techniques

Deep learning has rapidly gained attention as a revolutionary field in recent times. In wildfire management, it has proven effective in detection, classification, and segmentation. These techniques can automatically learn patterns from various data sources like satellite and aerial images, aiding in the timely and accurate identification of wildfires. They handle inputs such as high-resolution images and real-time video streams, contributing to improved wildfire management through active-fire detection and real-time monitoring [23].

2.2.2.1 Wildfire Classification Approaches

Classification task stands as one of the earliest subjects in the context of wildfire detection. Image classification involves assigning categories and labels to pixel or vector groups within an image according to their characteristics. Ensuring that every image is categorized according to its appropriate category is the primary goal of image classification [33].

In the following section, we briefly discuss influential convolutional neural network architectures commonly used in classification tasks. DenseNet emphasizes dense connectivity, EfficientNet employs compound scaling for efficiency, MobileNet focuses on computational cost reduction, and Inception uses parallel convolutions for multi-scale feature extraction.

- **DenseNet** incorporates the skip connection idea from ResNet, creating a network where each layer connects to all preceding layers. This design leads to diverse feature maps, aiding in feature reuse, propagation, and addressing vanishing gradients [34].
- **EfficientNet** incorporates architecture scaling, a common strategy in neural network design aimed at improving efficiency. The compound coefficient uniformly scales the dimensions of the network using a constant ratio, resulting in enhanced efficiency during training [35]
- **MobileNet** is a lightweight convolutional neural network specifically designed for efficient deployment on mobile devices. The primary focus of MobileNet is to provide a computationally efficient yet effective solution for mobile vision applications, considering factors such as low power consumption, fast execution, and minimal memory usage [36].

- **Inception** architecture, also known as GoogLeNet, introduced the concept of inception modules. Thees modules simultaneously use filters of different sizes, enabling the network to understand information at various levels of details and larger patterns in the input data efficiently. The use of multiple filter sizes aims to find a balance between computational requirements and network depth. Unlike traditional networks with a fixed filter size, Inception allows the network to adapt and select the most suitable filter size for different parts of the input, enhancing its ability to understand diverse features. This design not only improves the network's capability to describe complex information but also optimizes the use of computer memory [37].

2.2.2.2 Wildfire Segmentation Approaches

Image segmentation consists of two components: the feature extractor and the segmentation decoder. The feature extractor is responsible for determining crucial patterns within input images and employs a sequence of convolutional and pooling layers to enhance its capability to extract significant features. On the other hand, the decoder decodes the extracted features to generate segmented outputs. The decoder incorporates essential components such as the upsampling layer, convolutional layers, and softmax layer. The upsampling layer, increases the spatial resolution of feature maps, resulting in recovering details that may have been lost during downsampling. This is critical for wildfire segmentation as it facilitates the precise identification and characterization of the boundaries of fire and smoke regions. The softmax layer, serving as the final layer in the segmentation decoder, converts the model's output into a probability distribution. For wildfire segmentation tasks, the softmax layer assigns probabilities to each pixel for different classes, such as fire and no-fire [23].

In the following section, we briefly introduce three key convolutional neural network architectures for semantic segmentation tasks. U-Net utilizes skip connections for spatial information preservation, SegNet employs an encoder-decoder architecture with max-pooling indices, and DeepLab utilizes atrous convolutions and spatial pyramid pooling for multi-scale context capture.

- **U-Net** is characterized by its distinctive U-shaped architecture. It consists of a contracting path for capturing broader context and an expanding path for precise feature localization, maintaining a careful balance between complexity and depth. U-Net uses skip connections to keep track of details across the network. This design choice has made U-Net widely used in different image segmentation tasks. Its flexibility and scalability have significantly influenced the development of computer vision systems [38].
- **SegNet** is a convolutional neural network architecture developed for semantic segmentation. SegNet focuses on retaining spatial information through an encoder-decoder architecture. The encoder captures hierarchical features, while the decoder restores spatial resolution. SegNet's simplicity lies in its use of max-pooling indices during the encoding phase, storing them for efficient up-sampling during decoding. This approach facilitates effective segmentation while maintaining computational efficiency. SegNet has been successfully applied in various domains, including autonomous vehicles and medical image analysis [39].

- **DeepLab** utilizes a deep convolutional neural network with atrous convolution to accurately classify and delineate objects at the pixel level, ensuring high precision in semantic segmentation. The network's ability to capture multi-scale contextual information while maintaining spatial resolution makes it effective for various applications, including image recognition and medical imaging [40].

2.2.2.3 Wildfire Object Detection Approaches

Identifying and locating specific objects in areas affected by wildfire, like flames and smoke is a crucial aspect of wildfire management. When firefighters and emergency responders can accurately track and identify these objects, they can make well-informed decisions on how to handle wildfires. This approach uses deep learning models to recognize objects of interest in images or videos, like fire and smoke. Similar to wildfire classification and segmentation, object detection methods rely on deep neural networks, often based on CNNs, to automate the recognition and delineation of wildfire-related objects [23].

- **YOLO** is characterized as a popular object detection algorithm in computer vision. This model, known for its real-time processing and unified architecture, stands out due to its unique approach. Unlike traditional models relying on region proposals, YOLO treats object detection as a regression problem, predicting bounding boxes and class probabilities in one step [41]. This design allows YOLO to process images in real time, making it notably faster than previous methods. The original YOLO model aimed to provide a unified solution for real-time object detection. Over time, various YOLO versions have been developed, each enhancing aspects like accuracy, speed, and model size. YOLOV2 has been acknowledged as state-of-the-art in tasks like PASCAL VOC and COCO. YOLOV3 and YOLOV4 represent further refinements, promising improvements in accuracy and speed [42]. YOLOV5, sharing a foundational architecture with YOLOV4, introduces numerous improvements in speed and precision [43]. YOLOV6 incorporates modifications, including an anchor-free detection approach and a feature pyramid network, adding flexibility and robustness to object size variations [44]. YOLOV7, adopting a single-shot detector architecture, integrates advanced features such as an EfficientNet-based backbone network, Spatial Pyramid Pooling (SPP), and Path Aggregation Network (PAN) [45].

CHAPTER 3

WILDFIRE DETECTION IN VIDEO WITH THE HELP OF OPTICAL FLOW

3.1 Introduction

Motion estimation plays an essential role in various applications within computer vision. By analyzing the temporal changes between consecutive frames in video sequences, motion estimation enables tasks such as object tracking, video stabilization, action recognition, and even depth estimation in 3D reconstruction.

In the context of wildfire detection, we propose an assumption based on the observation that wildfires and their smoke move more rapidly and randomly compared to other visually similar environmental elements such as fog and clouds. This difference in movement speed and randomness forms the basis of our approach to enhance the accuracy of wildfire detection.

To test our hypothesis, we developed an algorithm that, in the first stage, identifies potential regions of fire based on a neural network. Then, only for those locations, it estimates the motion between consecutive frames. The corresponding estimated motion for each region is then fed to a secondary neural network for classification purposes. This approach allows us to detect wildfires and make decisions based on the estimated motion associated with wildfires and smoke, distinguishing them from the slower movements of other environmental factors.

3.2 Related Work

In the domain of wildfire detection, researchers have applied motion estimation particularly optical flow techniques for diverse objectives. In the following section, we provide an overview of how these techniques have been utilized for wildfire detection.

Wang et al. proposed an algorithm for fire smoke detection. The algorithm starts by using bilateral filter on each frame to remove interference caused by poor video quality. It then employs a Gaussian mixture model with background subtraction to extract foreground regions. Using a color model, the algorithm identifies suspected smoke areas. Once identified, texture features are extracted from the suspected areas using Local Binary Pattern (LBP) and Local Binary Pattern Variance (LBPV) methods. Additionally, the algorithm uses optical flow based on an image pyramid to determine the movement direction of suspicious area contours. Finally, a support vector machine is utilized to classify the texture and motion features [46].

Avalhais et al. proposed a fire detection method called SPATFIRE (SPAtio-Temporal segmentation of FIRe Events) to address challenges posed by videos captured with non-stationary cameras and uncontrolled lighting conditions. It uses a color model named Fire-Like Pixel Detector, which operates in the HSV color space, to detect fire-like regions in video frames. The algorithm starts with a color spatial segmentation to pre-select candidate fire regions and then performs a temporal segmentation to extract sparse flow from candidate regions and dense flow from the background in consecutive frames. Global motion compensation is achieved using block-based motion estimation to correct for camera movement. The final step involves a support vector machine classifier which analyzes motion patterns to determine whether a specific video segment contains fire [47].

Yuan et al. proposed a forest fire detection method that utilizes both color and motion features for processing images captured from UAV. The system architecture consists of two layers to improve the robustness of fire detection. The first layer, color-based fire detection, segments fire candidate regions using a light computation algorithm that employs chromatic features in the Lab color space to detect fire-colored pixels while filtering out non-fire-colored pixels. The second layer, motion-based fire detection, further analyzes the segmented fire-colored regions using two optical flow methods. A classical optical flow estimates camera motion, while the optimal mass transport optical flow method assesses motion within the regions. Finally, fire pixel candidates are confirmed based on an empirical discrimination rule [48].

They also proposed an algorithm that utilizes brightness and motion clues, as well as image processing techniques such as histogram-based segmentation and optical flow approaches for detecting fire pixels. The process starts with histogram-based segmentation to identify hot objects as fire-candidate regions, filtering out non-fire background areas. Next, the classical optical flow method calculates motion vectors within the candidate regions to identify moving objects and eliminate stationary non-fire objects. Once fire pixels are confirmed, fire zones are tracked using a blob counter scheme [49].

Hu et al. proposed an approach for real-time fire monitoring which combines a deep convolutional long recurrent network (DCLRN) with an optical flow method. The method utilizes the static and dynamic characteristics of fire by converting RGB fire images into optical flow images in real time. It employs convolutional neural networks for spatial learning and recurrent convolutional architectures for sequence learning allowing for effective fire monitoring [50].

Dang-Ngoc et al. proposed a forest fire detection approach using UAV-captured. Their method addresses the challenge of moving backgrounds and objects in dynamic scenes. The detection process consists of two stages. Firstly, fire-colored pixels are determined using multi-color features, employing RGB, YCbCr, and HSI color spaces. Secondly, an optical flow algorithm analyzes the motion characteristics of forest fires, extracting motion features. The results are then combined, and fire pixels are identified using specific rules [51].

Gupta et al. proposed a method to enhance the segmentation of smoke plumes in wildfire smoke videos. Their method includes four key components: the dark channel de-haze pre-processor, the Dynamic Optimal Frame (DOP) Module, the dense optical flow module, and the spatio-temporal segmentation network. In the dark channel de-haze pre-processor, image haze removal is employed to eliminate haze from video frames, making smoke more discernible. The DOF Module involves salient frame selection and automatic mask annotation. Frame selection is facilitated by BubbleNet. The dense optical flow module computes temporal features using Farneback's motion estimation technique. The spatiotemporal segmentation network, a fully convolutional network, integrates inputs from annotated frames, automatic mask annotation, and dense optical flow, generating smoke masks for the entire video [52].

Wahyono et al. proposed a fire detection system that begins by creating a color probability model using the Gaussian Mixture Model and Expectation Maximization (GMM-EM) methods to segment fire regions. After obtaining the candidates, they utilize a support vector machine and random forest to verify them based on color histograms. Further verification involves assessing the motion of fire regions, and examining their centroids and areas. If the method detects irregular motion in a fire region, it is classified as a fire; otherwise, it is not. [53].

Peng et al. proposed a flame detection method that combines image processing and deep learning methods. Their approach includes three main steps: motion detection, visual saliency detection (VSD), and flame image classification using transfer learning. The process begins by capturing images from cameras. CNT background subtraction algorithm is employed to extract moving objects, then creates a boundary around the moving objects for further analysis. Subsequently, three salient region images are extracted from the moving images using visual saliency detection, particularly ObjectnessBING. The system incorporates flame image classification based on transfer learning [54].

Shahid et al. proposed a hybrid CNN-ViT model for fire detection. Their model utilizes spatial attention to capture salient fire characteristics within inter-frame, alongside temporal attention to identify fire flicker and motion across frames. They leverage a pre-trained feature extractor and employ domain adaptation techniques. The integration of spatial attention and the temporal attention module has notably enhanced the capability of fire detection performance compared to single image-based models, particularly in challenging and highly interfered environments [55].

3.3 Methodology

We develop a two-stage wildfire detection algorithm, as shown in Figure 2, building on our prior work illustrated in Figure 1. This algorithm efficiently identifies and verifies potential wildfire regions. In the following section, we provide an overview of the overall structure:

- We use a block-based method in our approach, rather than inputting the entire frame to the neural network. In this method, each frame is divided into smaller blocks, and each block is analyzed individually. This approach simplifies the creation of training datasets compared to techniques such as R-CNN and YOLO [41,56]. Instead of requiring annotated bounding boxes for each object, our block-based analysis marks each block as either fire or no-fire, making it easier to detect small fire regions and smoke. More information on this is given in Section 3.3.1.
- In the first stage, a neural network based on EfficientNetV2 is used to identify possible wildfire areas with computational efficiency. Section 3.3.2 provides more details on EfficientNetV2.
- In the second stage, a verification step uses the Gunnar Farneback optical flow method to estimate motion in video frames. This step confirms whether the regions identified in the first stage are indeed fire by helping differentiate between actual fire and other smoke-like phenomena such as clouds and fog. The system processes pairs of frames at a time, leveraging temporal information to enhance the capability of the system to differentiate between smoke and visually similar phenomena, thereby improving fire detection

by reducing false alarms. More details on the optical flow method are provided in Section 3.3.3.

- For motion field classification, we use a ResNet18 neural network, chosen for its balance between computational efficiency and strong performance in various computer vision tasks. Further details on the ResNet architecture and training process can be found in Sections 3.3.4 and 3.5.

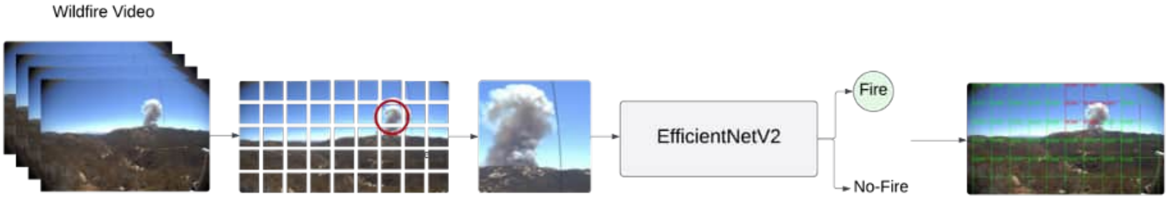


Figure 1: Preliminary Work: Single-Stage Wildfire Detection: Addressing False Positives in Non-Wildfire Videos.

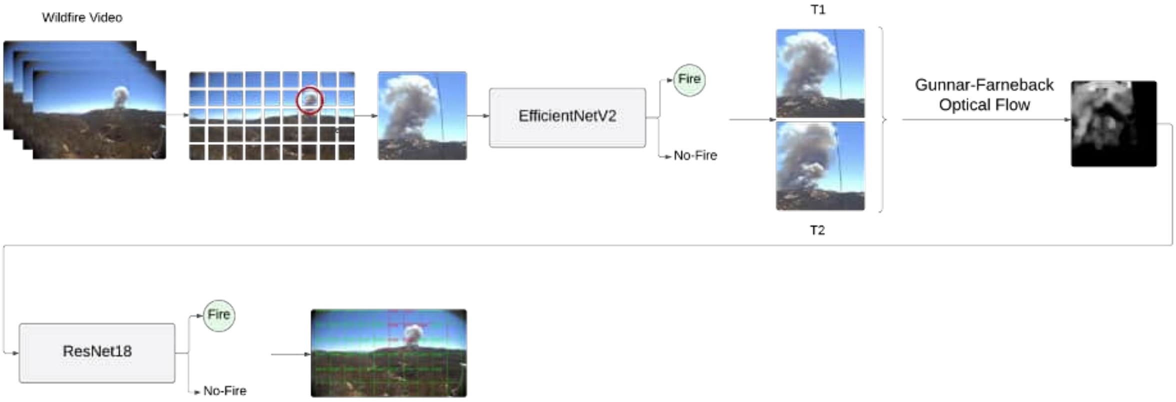


Figure 2: Proposed Method: Two-Stage Wildfire Detection with Motion Estimation Integration: Reducing False Positives.

3.3.1 Block-Based Analysis of Frames

In our approach to wildfire detection, the method involves analyzing frames by dividing them into smaller blocks, each with a size of 224×224 pixels. This block-based approach allows for a more effective assessment of the presence of smoke, particularly in cases where it may be located near the edges of the frame. The process begins by dividing the entire frame into a grid of 45 blocks, each of the specified size. The key aspect of this division is the assignment of a probability value to each individual block. This value represents the likelihood of a fire occurring within the subset of the image covered by the block itself, as well as its adjacent blocks to the right, bottom, and bottom-right. To determine the position of each block within the image, the width and height of the image region (M_i and N_i, respectively) are considered, along with the width and height of each block (M_b and N_b). The row number (R) and column number (C) of each block are then calculated using the floor function. The formula for these calculations is as follows:

$$R = \left\lfloor \frac{N_i}{N_b} \right\rfloor, \quad C = \left\lfloor \frac{M_i}{M_b} \right\rfloor$$

where $\lfloor \cdot \rfloor$ denotes the floor function [4].

3.3.2 EfficientNet Architecture

EfficientNet is a type of convolutional neural network designed specifically for image classification tasks. The main aim of EfficientNet is to improve performance while using fewer resources. Examples of these resources are computational power and memory usage. It achieves this by using a unique method called compound scaling, which adjusts the network's depth, width, and resolution in a balanced way. Depth scaling increases the number of layers in the network to capture more features without over-fitting. Width scaling increases the number of channels in each layer, enhancing the network's ability to represent data without adding too much computational burden. Resolution scaling involves resizing input images to capture finer details without significantly increasing computational cost [57]. In addition, EfficientNetV2 improves training speed and efficiency by using techniques like training-aware neural architecture search and introducing new operations like Fused-MBConv [58].

3.3.3 Gunnar-Farneback Algorithm

Optical flow refers to the apparent motion observed in a sequence of images captured by a camera over time. In simple terms, it represents how the positions of pixels change in consecutive frames, indicating the perceived movement of objects within the scene. This concept helps in understanding the dynamic aspects of a visual sequence, particularly when there is motion involved such as a moving camera or objects within the scene [59].

The Gunnar-Farneback algorithm is a dense optical flow algorithm builds upon the Lucas-Kanade method and is known for its effectiveness in handling both small and large displacements. It involves Gaussian pyramid for multi-scale analysis and a local polynomial expansion to capture non-linear motion patterns. This method is known for its ability to estimate dense flow fields across different scales, making it suitable for a wide range of computer vision applications [60].

3.3.4 ResNet Architecture

ResNet, is a type of convolutional neural network designed for image classification tasks. The distinguishing feature of it introduces the concept of residual learning by utilizing residual blocks. A residual block consists of a skip connection, that skips one or more layers. This helps in mitigating the vanishing gradient problem, making it easier to train deep networks. ResNet-18, the introductory model, comprises 18 layers and is characterized by basic building blocks. The deeper ResNet-34, with 34 layers, builds upon the ResNet-18 architecture, offering increased capability in capturing features. ResNet-50 introduces bottleneck blocks. These blocks consist of three consecutive convolutional layers, effectively reducing computational complexity while maintaining the network's performance. It creates a balance between efficiency and depth, making it well-suited for various computer vision tasks. ResNet-101 and ResNet-152 stand out with 101 and 152 layers, respectively [61].

3.4 Dataset for Training

Data is crucially important for the success of classification tasks in machine learning. The quality, quantity, and diversity of datasets significantly impact the development of effective classification models. An example is Krizhevsky et al.'s ImageNet Classification task, which highlighted the power of deep convolutional neural networks trained in massive labeled image datasets. This study has demonstrated that having a large and diverse dataset significantly improves the strength and adaptability of classification models [62].

The ImageNet Large Scale Visual Recognition Challenge, discussed by Russakovsky et al. emphasized the importance of large datasets in increasing computer vision capabilities. This competition focuses on classifying a wide range of images, emphasizing the importance of different datasets for improving visual recognition. Researchers involved in the challenge realized that the quantity of data is critical to the model's ability and performance to accurately classify [63].

Eventually, the acquisition and use of high-quality, diversified dataset is critical for obtaining high classification performance. These findings demonstrate that efficient classification requires careful consideration of the scale and diversity of datasets utilized for model training.

The primary dataset to train the initial neural network contained around 14k images of non-wildfire and 9k wildfire images [3]. To train the secondary neural network to classify fire and non-fire motion fields, we needed a specific dataset. Since an appropriate dataset wasn't available, we created our dataset. This involved using videos from HPWREN and the internet that depicted wildfire, fog, and any kind of environmental elements that have visual similarity to smoke [64]. We extracted frames from these videos, each frame having a size of 1920×1080 pixels, and divided them into 224× 224 pixel blocks. Pairs of fire and non-fire images became the basis for estimating the motion field. The dataset creation procedure is presented in Figure 3 and Figure 4. The dataset is organized into two categories, motion fields associated with fire and non-fire. In each class, we had around 3K images. A few samples of training datasets are shown in Figure 5 and Figure 6.

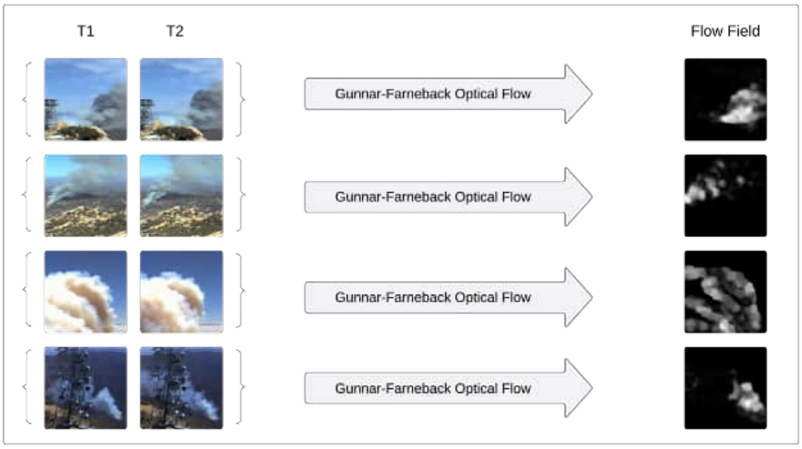

T1
T2
Flow Field
Gunnar-Farneback Optical Flow
Gunnar-Farneback Optical Flow
Gunnar-Farneback Optical Flow
Gunnar-Farneback Optical Flow

Figure 3: Fire Motion Filed Dataset Creation Process

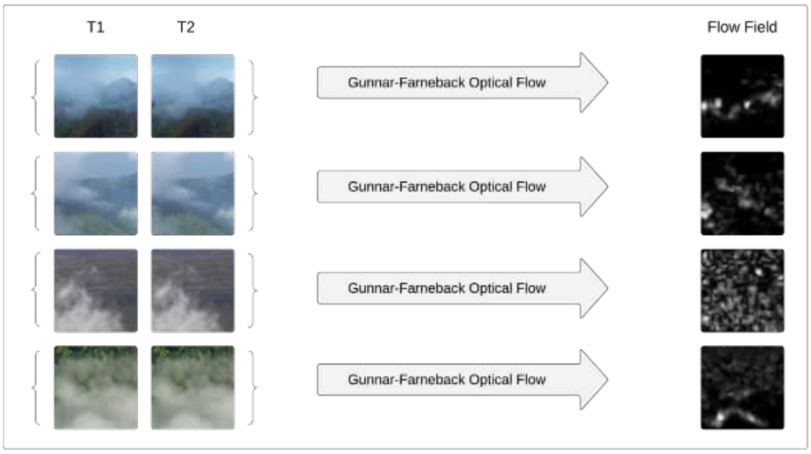

T1
T2
Flow Field
Gunnar-Farneback Optical Flow
Gunnar-Farneback Optical Flow
Gunnar-Farneback Optical Flow
Gunnar-Farneback Optical Flow

Figure 4: Non-Fire Motion Filed Dataset Creation Process

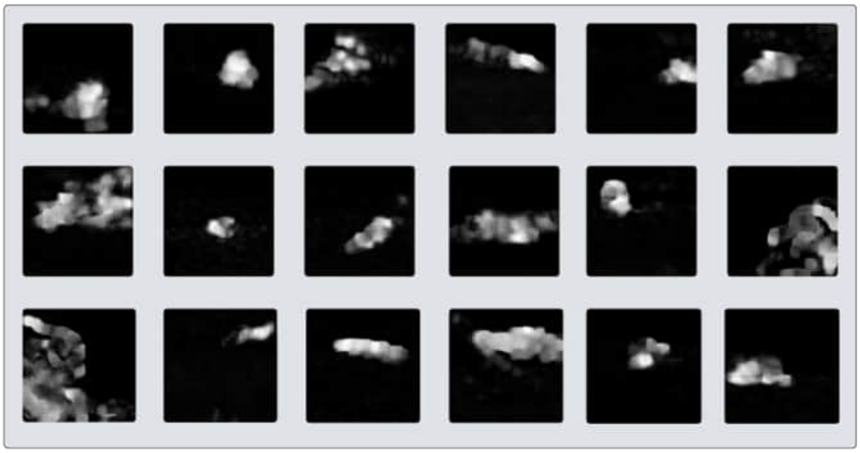


Figure 5: Fire Motion Filed Dataset Samples for Training

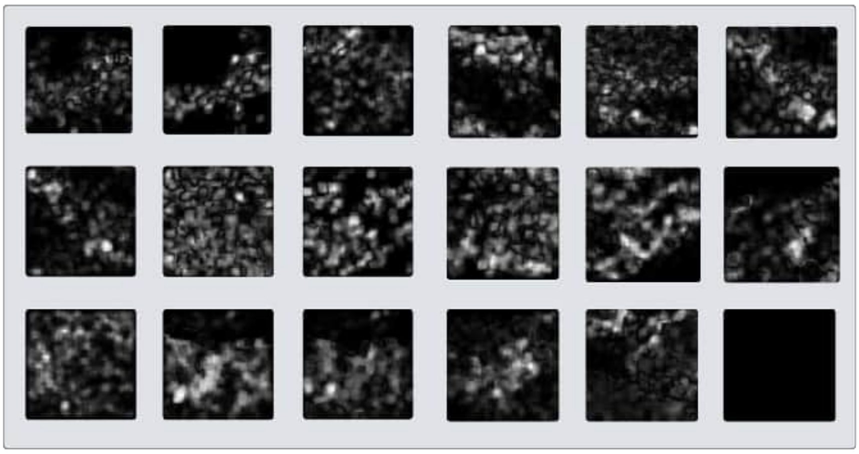


Figure 6: Non-Fire Motion Filed Dataset Samples for Training

3.5 Implementation Details

Given the limited size of our dataset, we employed transfer learning. Transfer learning offers distinct advantages, particularly when dealing with small datasets. Another significant advantage of transfer learning, especially relevant to our scenario, is its ability to mitigate the risk of overfitting [65]. The pre-trained ResNet18 model arrives with a pre-existing understanding of generalized features, enabling it to adapt rapidly to our specific classification task. This not only enhances the model's capacity to discern subtle differences in motion field patterns but also promotes a better generalization on our limited dataset. During initialization, the pre-trained model was loaded, and adjustments were made to the final fully connected layer to align with our binary classification task. Stochastic gradient descent was used as the optimization algorithm with an initial learning rate set to 0.001. Throughout training, the model's parameters remained unfrozen, enabling continuous adaptation through gradient updates computed based on the loss function. In the subsequent validation phase, we assessed the model's effectiveness on a separate dataset to understand its ability to generalize. To regulate the learning rate during training, a dynamic learning rate scheduling mechanism was introduced. The learning rate underwent reduction by a factor of 0.1 every 7 epochs, ensuring a carefully managed convergence. We set the number of epochs to 20 and evaluated the model after each epoch, calculating accuracy metrics, including loss and overall accuracy, for both the training and validation phases.

3.6 Experimental Results

In this study, we evaluated our method across a total of 160 videos, evenly split between those depicting wildfire scenes and those depicting non-wildfire situations. The selection of non-wildfire videos was based on weather conditions resembling those of wildfires, such as fog, cloud, or dust. To assess the efficacy of incorporating motion estimation in wildfire detection, we conducted tests using two approaches: one utilizing a neural network solely for detection and another combining the neural network with motion estimation. To maintain consistency across all videos, we standardized the frame rate to 4 frames per second and specifically targeted videos captured by stationary cameras. Additionally, we examined the impact of analyzing consecutive frames versus frame skipping to assess motion. We will assess the network's performance on wildfire videos in Section 3.6.1. False alarms in non-wildfire videos will be analyzed in Section 3.6.2. Additionally, detection latency will be evaluated in Section 3.6.3

3.6.1 Network Performance

To evaluate the true detection rate, we initially quantify the number of blocks identified as fire by the initial neural network trained on fire and non-fire datasets utilizing EfficientNetV2. These identified blocks serve as the baseline for our analysis. Subsequently, we measure the blocks identified as fire after integrating motion estimation into our method and passing them through a second neural network trained on motion fields utilizing ResNet18. We conduct true detection rate analysis on both consecutive frames and frame skipping.

In Table IX, we present a detailed comparison between the performance of the initial neural network and its integration with motion estimation when analyzing consecutive frames. Similarly, Table XI examines the performance when employing frame skipping. Each row in the tables represents findings from a single video.

In the "Without Motion Estimation" column, we calculated the fire regions using only the initial neural network trained on fire and no-fire images. "Detection No." indicates the total number of identified fire region blocks in the entire wildfire video. In the "With Motion Estimation" column, "Detection No." indicates the fire-detected blocks identified after integration of motion estimation. Thus, this number indicates the number of fire-detected blocks after the second neural network classifies the motion fields. We use the number of blocks identified by the first neural network as our baseline to calculate the detection rate for the integration of motion estimation. Hence, the "Detection Rate" is calculated by dividing each value in the "Detection No." column of "With Motion Estimation" by the "Detection No." of "Without Motion Estimation"

For instance, in Table IX. for video number 1, the initial neural network detects 74 fire blocks throughout the video, while the integration of motion estimation and the subsequent analysis by the second neural network identifies 74 fire blocks. Thus, we divide 74 by 74 to obtain the detection rate of 100.0%.

For most videos in the dataset, while analyzing consecutive frames, we maintained a high detection rate. For example in video number 1, 2, 4, 9, 28, 31, 39, 43, 46, 51, 54, 56, 59, 64, 65 and 74 we obtained a true detection rate of 100%. However, in some cases, such as videos number 5, 26, 48, 75 and 76 we obtained a detection rate of less than 50% but we do not miss fire in any videos.

Comparing frame analysis methods, frame skipping demonstrates a higher true detection rate compared to analyzing consecutive frames.

For instance, in video number 33 in Table XI, the true detection rate increases from 93.65% to 94.75% when employing frame skipping. Similarly, in video number 80, which initially had a true detection rate of 84.95%, frame skipping yields a true detection rate of 92.66%.

Table I summarizes the results of our comprehensive assessment of the wildfire testing dataset. It turns out that frame skipping leads to a higher average true detection rate, highlighting its effectiveness in improving detection accuracy.

	Without Motion Estimation	With Motion Estimation
	True Detection Rate (%)	True Detection Rate (%)
Consecutive frames	-	81.33
Skipping one frame	-	**83.50**

TABLE I: Average True Detection Rate Comparison in a Total of 80 Wildfire Videos

3.6.2 False Alarm Analysis

To evaluate the false alarm rate, we first calculate the total blocks in a single video by multiplying the frame number by 45 as we divide each frame into 45 overlapping blocks. Since in non-wildfire videos, the system should not trigger any alarm, we consider any alarm as a false alarm.

In Table XIII, we present a detailed comparison between the performance of the initial neural network and its integration with motion estimation when analyzing consecutive frames. Similarly, Table XV examines the performance when employing frame skipping. Each row in the tables represents findings from a single video.

In the "Without Motion Estimation" column, we measured how many times the initial neural network triggers alarms, in other words, how many times it classifies any visually similar environmental element as smoke or fire. "False-Alarm No." indicates the total number of misclassified fire region blocks in the entire non-wildfire video. The False-Alarm Rate is calculated by dividing the False-Alarm No. by the total number of blocks, as we described earlier.

Our analysis, as presented in Table XIII, demonstrates a notable reduction in false alarm rates for non-wildfire videos when motion estimation is integrated into the initial neural network. Specifically, we observed a consistent improvement in reducing false alarms across most scenarios.

For instance, in the case of video number 1, the initial neural network identified 187 false alarms, whereas the incorporation of motion estimation reduced this count to 0 false alarm across the entire video, resulting in a reduction from a false alarm rate of 5.26% to 0%. However, as evidenced in video number 7 and 8, the false alarm rate remained unchanged.

Furthermore, our examination of frame skipping, as detailed in Table XV, revealed no significant disparity in results between consecutive frame and frame skipping analysis. This suggests that the integration of motion estimation into the system, irrespective of the frame analysis method employed, effectively mitigates false alarms in non-wildfire scenarios.

Table II presents the outcomes derived from our comprehensive assessment of the non-wildfire testing dataset. Each video was individually analyzed using our method, allowing us to quantify the occurrence of false alarms in each instance and subsequently compute the average across the dataset. Our analysis shows a notable improvement, with the false alarm rate decreasing significantly from 5.70% to 0.33% when examining consecutive frames, and from 5.69% to 0.40% when employing frame skipping.

	Without Motion Estimation	With Motion Estimation
	False-Alarm Rate (%)	False-Alarm Rate (%)
Consecutive frames	5.70	0.33
Skipping one frame	5.69	**0.40**

TABLE II: Average False Alarm Rate Comparison in a Total of 80 Non-Wildfire Videos

3.6.3 Detection Latency

Detection latency, which measures the time it takes for a wildfire detection system to raise an alarm after the onset of a fire, is commonly assessed in frames.

In our evaluation in Table III, we establish the baseline using the initial system without motion estimation, considering the detection of fire in a frame as our reference point. In most cases, the incorporation of motion estimation does not compromise the system's performance.

For instance, in the first video, both consecutive frame analysis and frame skipping analysis detect fire at frame 47 showing comparable results. However, there are cases where the proposed frame-skipping analysis outperforms consecutive frame analysis.

For example, in video 5, fire is detected at frame 44 in consecutive frame analysis, but at frame 43 in frame skipping analysis, representing a notably improved detection time.

Overall, frame-skipping analysis demonstrates superior performance compared to consecutive frame analysis.

TABLE III: Detection Latency Comparison in Consecutive Frame and Frame Skipping Analysis

	Without Motion Estimation	With Motion Estimation	
Video No.		Consecutive Frames	Frame skipping
1	47	47	47
2	57	57	57
3	44	44	44
4	45	45	45
5	42	44	43
6	50	50	50
7	42	47	43
8	44	44	44
9	48	48	48
10	45	45	45
11	1	1	1
12	45	48	46
13	57	57	57
14	23	23	25
15	45	45	45
16	36	36	36
17	42	42	42
18	41	41	41
19	41	41	41
20	45	45	45
21	33	33	33
22	45	45	45
23	32	32	32

Continued on next page

Table III – continued from previous page

	Without Motion Estimation	With Motion Estimation	
Video No.		Consecutive Frames	Frame skipping
24	41	41	41
25	44	44	44
26	44	46	46
27	44	44	44
28	57	57	57
29	48	48	48
30	24	24	27
31	49	49	49
32	49	50	49
33	42	42	42
34	26	26	26
35	34	34	34
36	56	56	56
37	43	43	43
38	48	48	48
39	67	67	67
40	49	55	55
41	54	62	62
42	45	45	45
43	53	53	53
44	47	47	47
45	46	50	50
46	77	77	77
47	43	43	43
48	49	57	57
49	42	42	42
50	45	46	46
51	44	44	44
52	43	43	43
53	43	49	49
54	59	59	59
55	41	41	41
56	54	54	54
57	44	44	44
58	45	45	45
59	44	44	44
60	47	49	49
61	43	43	43
62	43	43	43
63	46	46	46
64	54	54	54
65	44	44	44
66	46	46	46
67	45	45	45
68	43	43	43
69	53	55	55
70	47	50	50
71	43	46	46
72	43	43	43
73	47	47	47
74	53	53	53

Continued on next page

Table III – continued from previous page

	Without Motion Estimation	With Motion Estimation	
Video No.		Consecutive Frames	Frame skipping
75	54	54	54
76	45	45	45
77	42	47	47
78	44	44	44
79	42	42	42
80	45	45	45

3.7 Comparison with Other Methods

In this section, we compare our method with other related works in the field of fire detection. It is challenging to make direct comparisons using the HPWREN dataset due to different testing data used in other methods. Additionally, many of the existing papers do not provide all three key metrics: detection rate, false alarm rate, and accuracy. However, we compare our false alarm rate with previous works. Wang et al. [46] achieved a false alarm rate of 5.40% while testing only one non-fire video. Dang-Ngoc et al. [51] reached an average false alarm rate of 3.91% across 20 videos, while Peng et al. [54] reported an average rate of 1.12% across 16 videos. In contrast, our method demonstrates a significant improvement, achieving an average false alarm rate of just 0.40% across 80 videos. This substantial reduction in false alarm rate highlights the efficacy of our approach.

Method	Detection Rate (%)	False-Alarm Rate (%)	Accuracy (%)
Wang et al. [46]	86.57	5.40	-
Avalhais et al. [47]	-	-	-
Yuan et al. [48]	-	-	-
Yuan et al. [49]	-	-	-
Hu et al. [50]	-	-	93.30
Dang-Ngoc et al. [51]	96.09	3.91	-
Gupta et al. [52]	-	-	95.30
Wahyono et al. [53]	89.97	-	-
Peng et al. [54]	-	1.12	99.28
Shahid et al. [55]	-	-	98.62
Our Method Without ME	-	5.69	93.47
Our Method With ME	83.50	0.40	98.00

TABLE IV: Comparison of Detection Rate, False-Alarm Rate, and Accuracy. ME stands for motion estimation.

Authors	Fire Video No.	Non-Fire Video No.
Wang et al. [46]	4	1
Avalhais et al. [47]	-	-
Yuan et al. [48]	2	-
Yuan et al. [49]	-	-
Hu et al. [50]	10	20
Dang-Ngoc et al. [51]	30	-
Gupta et al. [52]	5	-
Wahyono et al. [53]	12	9
Peng et al. [54]	11	16
Shahid et al. [55]	-	-
Our Dataset	80	80

TABLE V: Comparison of Fire and Non-Fire Video Numbers in Testing Dataset

3.8 Conclusion

In this chapter, we utilize motion estimation specifically in wildfire detection systems. Relying solely on the initial neural network to detect fire may lead to false alarms especially when environmental elements such as fog and clouds are present. After the initial neural network identifies fire or smoke areas, motion estimation is applied, focusing only on locations recognized as fire by the first model. Gunnar-Farneback optical flow, known for dense motion estimation, was employed to generate motion fields from consecutive frames. These motion fields were then fed into a second neural network trained on both fire and fog motion fields. Transfer learning was utilized due to limited datasets, adapting the ResNet18 architecture for training motion fields. Testing involved a dataset of both non-wildfire and wildfire videos. The primary aim was to minimize false alarms in non-wildfire videos while maintaining a high true detection rate for wildfires. Remarkable improvements were observed, with the false alarm rate dropping significantly from 5.69% to an impressive 0.40%, without compromising the true detection rate. Furthermore, the system exhibited resilience when tested under different conditions, such as skipping frames, showing adaptability and reliability. The integration of motion estimation effectively mitigated false alarms in non-wildfire videos while maintaining a commendable true detection rate for wildfire events.

CHAPTER 4

WILDFIRE DETECTION IN VIDEO VIA CHANGE DETECTION

4.1 Introduction

Image change detection has applications across various disciplines such as remote sensing [66]. Researchers in this field focus on identifying regions of change in images captured at different times, which is known as the change map or change mask. These changes can be caused by various factors, including the appearance or disappearance of objects, object motion relative to the background, or changes in the shape, brightness, or color of stationary objects [67].

However, it's crucial to differentiate significant changes from those induced by camera motion, sensor noise, illumination variation, or non-uniform attenuation, which are considered unimportant changes. The challenge lies in defining what is considered significantly different or unimportant, as these criteria can vary across applications.

In the context of wildfire detection, we propose an assumption based on the observation that wildfires and their smoke move more rapidly than other similar environmental elements such as fog and clouds. To test our hypothesis, we use an image change detection approach that measures changes between consecutive frames to detect wildfires and make decisions based on these changes.

4.2 Methodology

The method outlined in this chapter follows a similar structure to our previous approach in Section 3.3, with a few small differences.

The difference in our approach lies in the verification process after detecting potential wildfire locations. In the previous chapter, we utilized motion estimation, specifically the Gunnar-Farnebak optical flow, for the verification step. However, in this chapter, we employ change detection techniques to verify whether the identified locations are genuine fires or false alarms. To perform this verification, we use two different change detection methods. The first method involves simple image differencing, while the second method uses a neural network for change detection. We use the change detection architecture described in the work of [1].

- We continue to employ a block-based method in our approach, as detailed in Section 3.3.1.
- In the initial stage, a neural network based on EfficientNetV2 is utilized to efficiently identify potential wildfire areas. More information about EfficientNetV2 can be found in Section 3.3.2
- In the subsequent verification stage, change detection is employed. Further details on the change detection techniques can be found in Section 4.2.1 and Section 4.2.2.
- For change map classification, we utilize a ResNet18 neural network due to its balance of computational efficiency and strong performance in various computer vision tasks. Additional information on the ResNet architecture and training process is available in Sections 3.3.4 and .

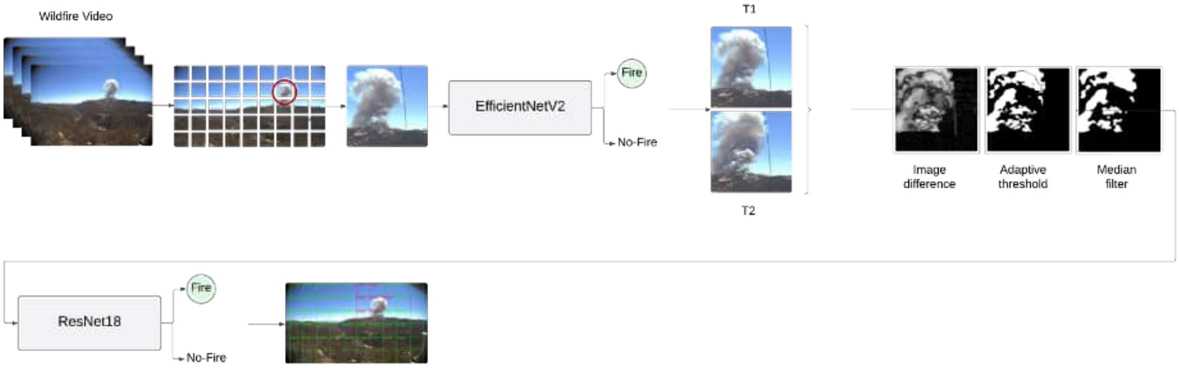


Figure 7: Proposed Method: Two-Stage Wildfire Detection with Classical Change Detection Integration

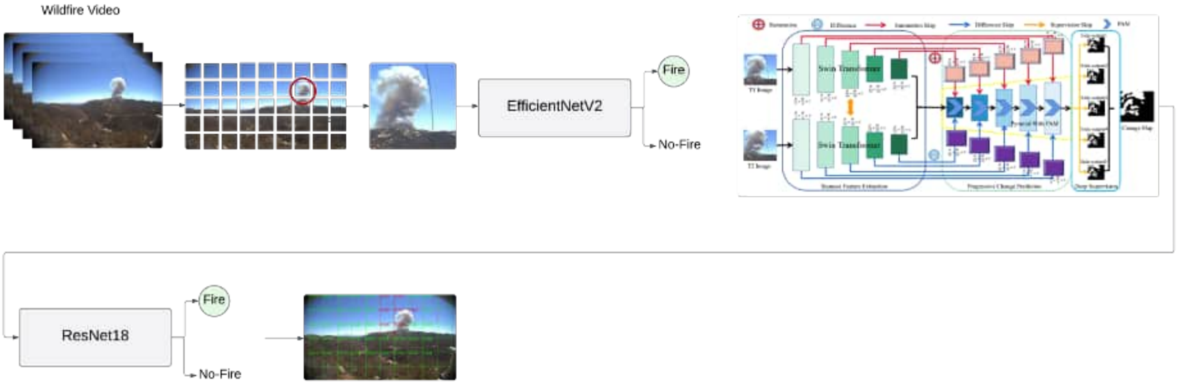


Figure 8: Proposed Method: Two-Stage Wildfire Detection with a Neural Network based Change Detection Integration. The neural network part is adapted from [1].

4.2.1 Change Detection Based on Image Differencing

Image difference analysis plays a crucial role in various computer vision applications, aiding in the identification of significant alternations between two images. This analysis involves computing the absolute difference between the grayscale representations of two images. This process highlights areas where pixel values differ, providing insight into potential modifications or alternations.

Following the computation of the absolute difference, an adaptive thresholding technique is applied to emphasize significant changes and filter out inconsequential variations. Adaptive thresholding is a technique employed to segment an image into regions of interest by establishing a dynamic threshold, rather than using a fixed value. It is essential in distinguishing between pixels representing significant changes and this within an acceptable range. This is achieved by evaluating the local intensity of pixels and adjusting the threshold accordingly.

In the implemented approach, a threshold value of 30 was chosen, designating pixel intensities above this threshold as significant changes. The mechanism of adaptive thresholding involves traversing the image and, for each pixel, computing a local threshold based on its neighborhood. This adaptability allows for improved sensitivity to variations in different regions of the image, making it a powerful tool in discerning subtle change.

Subsequent to adaptive thresholding, a median filtering step is introduced to refine the results. The median filter is a non-linear smoothing technique that replaces each pixel's value with the median value of its local neighborhood. The median filter plays a crucial role in noise reduction and the elimination of outliers from the binary thresholded image.

The median filter, with a specified kernel size of 5, influences the extent of smoothing applied to the thresholded image. This filtering step enhances the overall accuracy of the analysis by making sure that unimportant details don't distract from real changes.

4.2.2 Change Detection Based on a Fully Transformer Network

In recent years, deep learning, particularly utilizing Convolutional Neural Networks, has gained prominence in remote sensing image processing due to its feature representation capabilities. However, existing methods have shortcoming, including the underutilization of semantic information in high-resolution images, leading to difficulty in distinguishing pseudo-changes. Additionally, boundary information in complex images is often lacking, resulting in irregular and visually unappealing change region delineations. Furthermore, the temporal information in dual-phase images is not fully exploited, contributing to the poor performance of current CD methods. To address these challenges, Fully Transformer Network (FTN) was introduced. The FTN uses a three-branch structure that utilizes Transformers' strengths in modeling long-range dependencies to improve global feature extraction. To emphasize areas of change, it generates summation and difference features by directly comparing temporal features of images from two different phases, providing comprehensive CD regions. A pyramid structure, combined with a Progressive Attention Module (PAM), is introduced to aggregate multi-level visual features from transformers, improving boundary perception and feature representation through additional inter dependencies with channel attentions [1].

The framework comprises three components: Siamese Feature Extraction (SFE), Deep Feature Enhancement (DFE), and Progressive Change Prediction (PCP). When given dual-phase remote sensing images as inputs, SFE initially extracts multi-level visual features using two shared Swin Transformers. Following this, DFE uses these visual features to produce summation and difference features, highlighting change regions with temporal information. Lastly, the Progressive Attention Module (PAM) is applied for the final Change Detection (CD) prediction [1].

4.2.2.1 Siamese Feature Extraction

The Siamese Feature Extraction (SFE) component of the framework employs a Siamese structure with two encoder branches that share learnable weights. This structure is designed for extracting multi-level features from dual-phase remote sensing images captured at temporal phases 1 (T1) and 2 (T2). The selected backbone, the Swin Transformer, reduces computational complexity by substituting the Multi-Head Self-Attention with Window-based and Shifted Window-based Multi-Head Self-Attention. The Swin Transformer introduces additional elements, such as MLP, LayerNorm (LN), and residual connections, to enhance representation ability. Through this structure, multi-level visual features are extracted, with high-level features capturing global semantic information and low-level features retaining local details, collectively aiding in the identification of change regions [1].

4.2.2.2 Progressive Change Prediction

In response to the varied shapes and scales of change regions,the framework introduces Progressive Change Prediction (PCP). Taking inspiration from the feature pyramid concept, PCP is designed to accommodate CD predictions across different scenarios. To enhance representation ability, a pyramid structure with a Progressive Attention Module (PAM) is employed, incorporating additional inter-dependencies through channel attentions. The structure of PAM involves taking summation features and difference features as inputs. A channel-level attention is then applied to amplify features relevant to change regions. Additionally, a residual connection is introduced to enhance the learning ability of the model. This combination of features within PCP allows for a comprehensive and progressive approach to change prediction, considering the diverse characteristics of change regions in terms of shapes and scales [1].

4.3 Dataset for Training

Our change detection model takes a pair of 224×224 images and generates the corresponding change map. To train our change detection model, we needed a pair of images before the change and after the change and their corresponding change map. Due to the unavailability of such a dataset, we created our dataset. To create the dataset, several HPWREN videos and videos on the internet containing fire and fog were used. We extracted the frames in each video. As the input to the neural network is a pair of 224×224 images, we divided each frame into 224×224 blocks. Then we only picked those blocks that contained fire and fog or clouds. To generate the change map for each pair of images we used simple image differencing followed by adaptive thresholding and median filtering between each two consecutive frames. Same as before, we divided each image difference frame into 224×224 blocks.

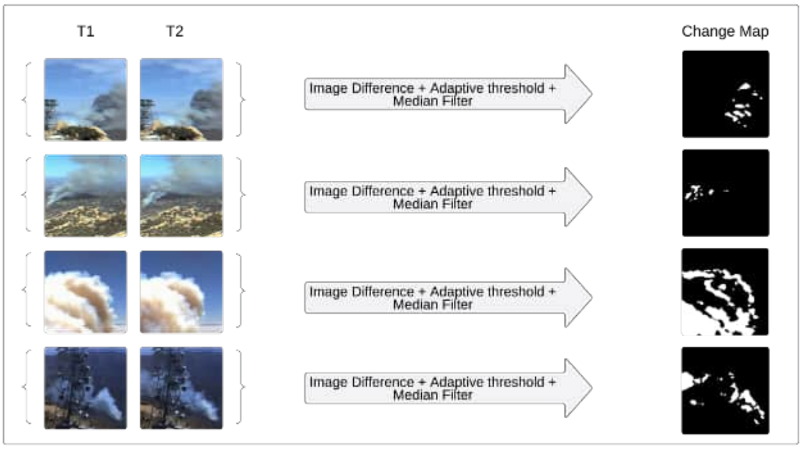

T1
T2
Change Map
Image Difference + Adaptive threshold + Median Filter
Image Difference + Adaptive threshold + Median Filter
Image Difference + Adaptive threshold + Median Filter
Image Difference + Adaptive threshold + Median Filter

Figure 9: Fire Change Mask Dataset Creation Process

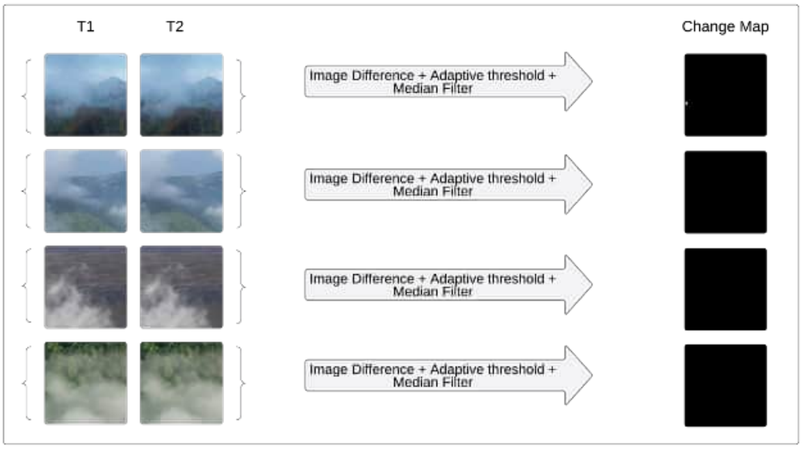

T1
T2
Change Map
Image Difference + Adaptive threshold + Median Filter
Image Difference + Adaptive threshold + Median Filter
Image Difference + Adaptive threshold + Median Filter
Image Difference + Adaptive threshold + Median Filter

Figure 10: Non-Fire Change Mask Dataset Creation Process

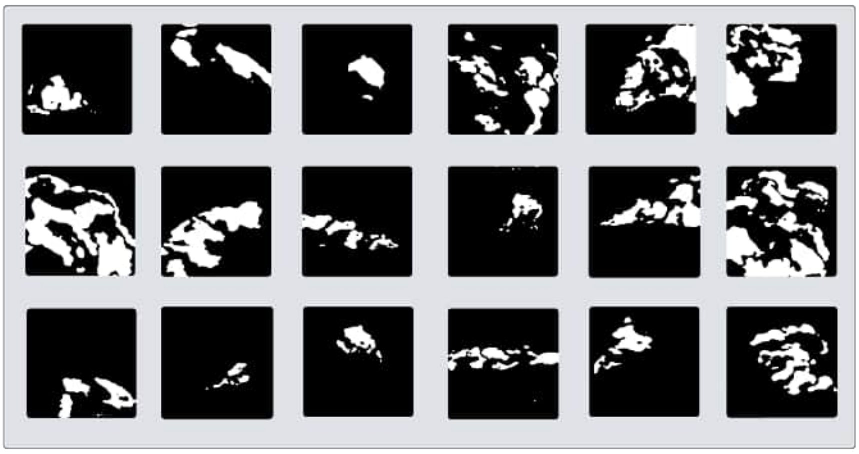


Figure 11: Fire Change Mask Dataset Samples for Training

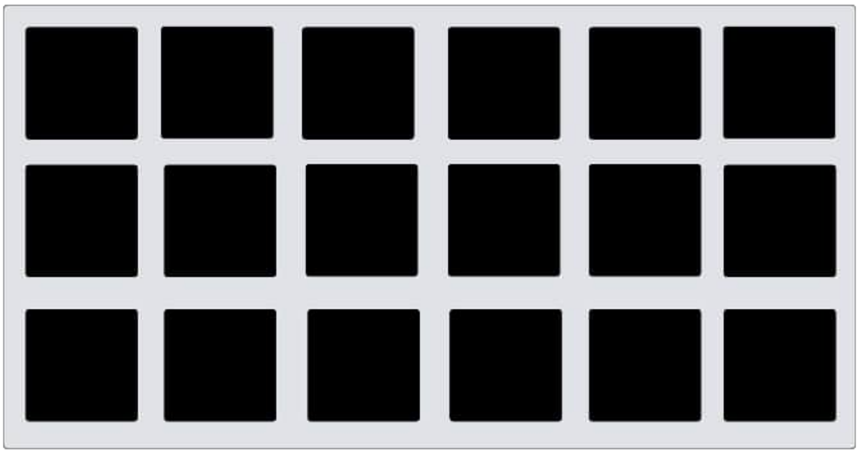


Figure 12: Non-Fire Change Mask Dataset Samples for Training

4.4 Implementation Details

We employed the code provided by the authors in [1] to train the neural network for change detection. The structure of the training dataset is organized as follows: Folder A contains images captured before the change, each with a resolution of 224×224 pixels. Folder B contains images captured after the change, and the Folder label contains the corresponding change masks.

To train ResNet18 for classifying change maps generated image differencing or neural networks, we followed the procedure outlined in Section 3.5. The setup remained the same, with the only variation being the dataset.

4.5 Experimental Results

In this study, we evaluated our method across a total of 160 videos, evenly split between those depicting wildfire scenes and those depicting non-wildfire situations. The selection of non-wildfire videos was based on weather conditions resembling those of wildfires, such as fog, cloud, or dust. To assess the idea of incorporating change detection in wildfire detection, we conducted tests using two approaches: one utilizing a classical change detection and a neural network-based change detection approach. To maintain consistency across all videos, we standardized the frame rate to 4 frames per second and specifically targeted videos captured by stationary cameras. Additionally, we examined the impact of analyzing consecutive frames versus frame skipping to assess change. We will assess the network's performance on wildfire videos in Section 4.5.1. False alarms in non-wildfire videos will be analyzed in Section 4.5.2.

4.5.1 Network Performance

To assess the true detection rate, we begin by quantifying the number of blocks identified as fire by the initial neural network trained on a dataset containing both fire and non-fire instances, utilizing EfficientNetV2. These identified blocks serve as the baseline for our analysis. Subsequently, we measure the blocks identified as fire after integrating change detection into our method and passing them through a second neural network trained on change maps using ResNet18. We then conduct a true detection rate analysis on both consecutive frames and frame skipping.

In Table X, we present a detailed comparison between the performance of the initial neural network and its integration with classical change detection and neural network-based change detection, when analyzing consecutive frames. Similarly, Table XII examines the performance when employing frame skipping. Each row in the tables represents findings from a single video.

In the "Without CD" column, we calculated the fire regions using only the initial neural network trained on fire and no-fire images. "Detection No." indicates the total number of identified fire region blocks in the entire wildfire video. In the "With Classical CD" column, "Detection No." indicates the fire-detected blocks identified after the integration of change detection. Thus, this number indicates the number of fire-detected blocks after the second neural network classifies the change maps. Similarly for column "With DL-based CD". We use the number of blocks identified by the first neural network as our baseline to calculate the detection rate for the integration of change detection. Hence, the "Detection Rate" is calculated by dividing each value in the "Detection No." column of "With Classical CD" and "With DL-based CD" by the "Detection No." of "Without CD"

Based on our findings it is evident that the incorporation of change detection, whether classical or deep learning-based, into the system fails to detect fire in the majority of testing videos, except for a few instances where the wildfire exhibits rapid movement, resulting in significant changes. For example, in Table X for video number 1, the initial neural network detects 74 fire blocks throughout the video, while integrating classical change detection and subsequent analysis by the second neural network identifies 12 fire blocks, resulting in a 16.21% detection rate. However, deep learning-based change detection only detects 4 fire blocks, resulting in a detection rate of 5.40%.

The approach of integrating change detection, whether classical or deep learning-based, into the system proves unsuccessful. In consecutive frame analysis, both classical and deep learning-based change detection fail to detect fire in video number 7 and numerous other instances, such as videos 29, 34, 43, 46, 47, 48, 50, 51, 68, 69, 75 and 76 for classical change detection, and videos 7, 29, 30, 34, 35, 41, 43, 45, 46, 47, 48, 50, 51, 53, 63, 65, 68, 69, 75 and 76 for deep learning change detection. Although there is a slight improvement observed in frame skipping analysis, both change detection methods still fail to reliably detect fire.

4.5.2 False Alarm Analysis

To evaluate the false alarm rate, we first calculate the total blocks in a single video by multiplying the frame number by 45 as we divide each frame into 45 overlapping blocks. Since in non-wildfire videos, the system should not trigger any alarm, we consider any alarm as a false alarm.

In Table XIV, we present a detailed comparison between the performance of the initial neural network and its integration with change detection when analyzing consecutive frames. Similarly, Table XVI examines the performance when employing frame skipping. Each row in the tables represents findings from a single video.

In the "Classical CD" the "False-Alarm No." indicates the total number of misclassified fire region blocks in the entire non-wildfire video. The False-Alarm Rate is calculated by dividing the False-Alarm No. by the total number of blocks, as we described earlier. Similarly for the "DL-based CD" column.

In Table XIV and Table XVI We notice a noteworthy improvement in reducing the false alarm rate in non-wildfire videos. In the majority of cases, we attained a 0% false alarm rate for non-wildfire videos. However, due to the failure of the approach to reliably detect fire, we cannot consider this method as reliable overall

4.6 Conclusion

In this chapter, we utilize change detection specifically in wildfire detection systems. Relying solely on the initial neural network to detect fire may lead to false alarms especially when environmental elements such as fog and clouds are present. After the initial neural network identifies fire or smoke areas, change detection is applied, focusing only on locations recognized as fire by the first model. A classical change detection using image differencing and a deep learning-based change detection were employed to generate change maps from consecutive frames. These change maps were then fed into a second neural network trained on both fire and non-fire change maps. Transfer learning was utilized due to limited datasets, adapting the ResNet18 architecture for training change maps. Testing involved a dataset of both non-wildfire and wildfire videos. The primary aim was to minimize false alarms in non-wildfire videos while maintaining a high true detection rate for wildfires. The results indicating a very low detection rate on wildfire videos suggest that this approach is not reliable.

CHAPTER 5

SUMMARY OF OUR FINDINGS

In this section, we present a summary of our findings to help the reader better compare the results. The summary is based on Table IX - Table XVI.

In Table VI, we provide a summary of the findings from Table IX - Table XII.

It turns out that among all three techniques, integrating motion estimation into the system performs best. In a comparative analysis between consecutive frame analysis and frame skipping, it was found that consecutive frame analysis achieved an average true detection rate of 81.33%, while frame skipping outperformed this with an average true detection rate of 83.50%. This assessment was conducted across a dataset of 80 wildfire videos.

Conversely, none of the change detection methods performed well, significantly compromising the true detection rate, thereby reducing the effectiveness on wildfire videos. Classical change detection achieved 27.20% using consecutive frame analysis, while frame skipping achieved 35.64%. On the other hand, DL-based change detection yielded worse results, with 11.22% on consecutive frame analysis and 16.72% on frame skipping. These findings lead to the conclusion that none of the change detection methods performed satisfactorily.

Method	**Frame Analysis**	
	Consecutive Frame	**Frame Skipping**
First Model + Motion Estimation	81.33%	**83.50%**
First Model + Classical Change Detection	27.20%	35.64%
First Model + DL-based Change Detection	11.22%	16.72%

TABLE VI: Comparison of Three Methods: Average True Detection Rate Across 80 Wildfire Videos

In Table VII, we provide a summary of the findings from Table XIII - Table XVI. We have calculated the average false alarm rate for each method.

Interestingly, we observed that the choice of frame analysis method does not significantly impact the false alarm rate on non-wildfire videos. Specifically, in consecutive frame analysis, we achieved an average false alarm rate of 0.33%, while in frame skipping, we achieved 0.40%.

Our findings indicate that DL-based change detection utilizing consecutive frame analysis yields the lowest false alarm rate at 0.10% compared to other methods. However, it's noteworthy that this approach demonstrates a very low true detection rate.

Method	**Frame Analysis**	
	Consecutive Frame	**Frame Skipping**
First Model	5.70%	5.69%
First Model + Motion Estimation	0.33%	**0.40%**
First Model + Classical Change Detection	0.47%	0.72%
First Model + DL-based Change Detection	0.10%	0.11%

TABLE VII: Comparison of Three Methods: Average False Alarm Rate Across 80 Non-Wildfire Videos

CHAPTER 6

CONCLUSION

This study focuses on a key problem in computer vision-based wildfire detection: false alarms.

Our earlier study relied on a single neural network for detecting wildfires, but it exhibited certain limitations, especially in foggy situations where it occasionally misclassifies fog or clouds as smoke. To overcome this challenge, we explored two distinct methods: optical flow-based motion estimation and change detection in video. The integration of Gunnar-Farnebeck based motion estimation into our existing approach emerged as a promising solution.

In this work, the process begins after the initial neural network identifies possible fire or smoke areas. Rather than applying motion estimation broadly, we focus on estimating motion only in locations recognized as fire by the first model.

For motion estimation, we used the Gunnar-Farnebeck optical flow, a technique known for its dense estimation of motion across all pixel values within frames. The resulting motion field serves as the input for the second neural network trained on both fire and fog motion fields. The challenge of limited datasets led us to employ transfer learning for training our model. We utilized the ResNet18 architecture to train the motion fields, leveraging its pre-trained features and adapting them to our specific task.

Upon implementation of the system, we conducted rigorous testing, utilizing a dataset consisting of 80 wildfire videos and 80 non-wildfire videos. Most wildfire videos contained 79 frames in total and all non-wildfire videos contained 79 frames.

The principle aim of evaluating the system's performance with non-wildfire videos was to assess its efficacy in mitigating false alarms. Generally, when the input corresponds to a wildfire or a non-wildfire video, the system is expected to avoid triggering any alarms, thereby minimizing occurrences of false positives. Simultaneously, we validated the system's ability to distinguish between fire and non-fire. This dual-testing strategy aimed to ensure that the system not only minimized false alarms but also maintained a high true detection rate for actual wildfire events.

The results showcased a remarkable improvement in the overall performance of the system. Integrating motion estimation with the initial neural network led to a substantial reduction in the average false alarm rate, decreasing from 5.70% to an impressive 0.33%. This enhancement was achieved without compromising the true detection rate, which remained acceptable at 81.33% when analyzing consecutive frames. It is important to note that all wildfire events are detected, ensuring no instances of missed fires.

Further exploration involved testing the system under different conditions, such as skipping one frame. Even with this alteration, the system exhibited resilience, while the false alarm rate slightly increased to 0.40%, the true detection rate improved to 83.50%. This resilience suggests that the system's effectiveness is not overly sensitive to the specific frames analyzed. Even when skipping one frame, it maintained an acceptable true detection rate for wildfire videos, showcasing its adaptability and reliability.

In the alternative approach, we delved into change detection methods to determine the most effective strategy for reducing the false alarm rate while maintaining a reliable true detection rate. Initially, we employed change detection utilizing image difference, followed by adaptive thresholding, and subsequently applied median filtering to retain only the significant changes between frames. Similar to the motion estimation approach, we trained a ResNet18 on change map images derived from our dataset containing instances of fire and no-fire. The resulting change image served as the input to the ResNet18, which was specifically trained on change maps.

The outcomes from the classical change detection methodology revealed a notable reduction in the false alarm rate on non-wildfire videos, decreasing from 5.70% to 0.47% when analyzing consecutive frames. However, when comparing consecutive frame analysis with skipping one frame, the false alarm rate slightly increased from 0.47% to 0.72%. Furthermore, integrating the system with classical change detection significantly diminished the true detection rate, reaching 27.20% when analyzing consecutive frames and 35.64% when skipping one frame. These findings suggest that when change detection is integrated with the initial neural network, analyzing non-consecutive frames yields better results.

However, it is noteworthy that the true detection rate on wildfire videos is worse which could be attributed to several factors.

Firstly, in videos containing wildfires, the size of the fire is often small, and the changes between each frame may not be sufficiently significant. Consequently, employing image change detection might not capture substantial alternations, leading the neural network to classify it as no change and impacting the true detection rate.

Secondly, even if there is a discernible change, misclassification by the second model may further contribute to a lower true detection rate.

Moreover, our exploration extended to image change detection based on neural networks, where we employed a Fully Transformer Network featuring three integral components: Siamese Feature Extraction (SFE), Deep Feature Enhancement (DFE), and Progressive Change Prediction (PCP). This network takes two images as input and predicts the change map. Subsequently, the change map serves as input to the ResNet18 to discern whether the detected change belongs to fire or non-fire.

The outcomes on non-wildfire videos were noteworthy, demonstrating that integrating the initial model with a deep learning-based change detection method substantially reduced the false alarm rate from 5.70% to 0.10% when analyzing consecutive frames. Even when analyzing frames with one frame skipped, the false alarm rate remained low at 0.11%.

However, this approach posed a significant impact on the true detection rate for wildfire videos diminishing to 11.22% when analyzing consecutive frames and 16.72% when analyzing frame skipping.

Overall, the integration of motion estimation into the system emerges as a robust strategy, effectively mitigating false alarms in non-wildfire videos while maintaining a commendable true detection rate for wildfire events. The incorporation of motion estimation, specifically through the Gunnar-Farneback optical flow method, refines the system's ability to discern between actual fire and non-fire. However, despite the success of motion estimation, our exploration into various change detection approaches, including classical methods and neural network-based methods reveals challenges in achieving a balance between reducing the false alarm rate and maintaining the true detection rate. These change detection methods, while showing promise in reducing false alarms for non-wildfire videos, introduce substantial compromises in the true detection rate for wildfire events.

APPENDICES

TABLE VIII: Videos in Testing Dataset from HPWREN Database

Video No.	Video Name	Video No.	Video Name
1	20200829-inside-Mexico-om-e-mobo-c	41	20170625-FIRE-mg-s-iqeye
2	20200829-inside-Mexico-pi-s-mobo-c	42	20170708-Whittier-syp-n-mobo-c
3	20200905-ValleyFire-lp-n-mobo-c	43	20170711-FIRE-bl-e-mobo-c
4	20200905-ValleyFire-pi-w-mobo-c	44	20170711-FIRE-sm-n-mobo-c
5	20200905-ValleyFire-sm-e-mobo-c	45	20171026-FIRE-smer-tcs8-mobo-c
6	20200906-BobcatFire-wilson-e-mobo-c	46	20171207-FIRE-bh-n-mobo-c
7	20200930-inMexico-lp-s-mobo-c	47	20180603-FIRE-rm-w-mobo-c
8	20201105-Roundfire-om-e-mobo-c	48	20180603-FIRE-sm-n-mobo-c
9	20201105-Roundfire-pi-s-mobo-c	49	20180603-FIRE-sm-w-mobo-c
10	20201127-Hawkfire-pi-w-mobo-c	50	20180603-FIRE-smer-tcs8-mobo-c
11	20201208-FIRE-om-s-mobo-c	51	20180603-FIRE-smer-tcs9-mobo-c
12	20210107-Miguelfire-om-w-mobo-c	52	20180605-FIRE-rm-w-mobo-c
13	20210110-Borderfire-lp-s-mobo-c	53	20180605-FIRE-smer-tcs9-mobo-c
14	20210209-FIRE-hp-e-mobo-c	54	20180606-FIRE-lo-s-mobo-c
15	20210711-FIRE-wc-e-mobo-c	55	20180606-FIRE-ml-s-mobo-c
16	20210810-Lyonsfire-housefire-lp-n-mobo-c	56	20180606-FIRE-pi-e-mobo-c
17	20220214-PrescribedFire-pi-n-mobo-c	57	20180611-fallbrook-rm-w-mobo-c
18	20220302-Jimfire-1101-stgo-e-mobo-c	58	20180612-FIRE-rm-w-mobo-c
19	20220302-Jimfire-1101-stgo-s-mobo-c	59	20180612-FIRE-smer-tcs9-mobo-c
20	20220622-HighlandFire-wc-n-mobo-c	60	20180614-Bridle-hp-n-mobo-c
21	20220713-Lonestarfire-om-w-mobo-c	61	20180614-Hope-wc-e-mobo-c
22	20220727-Casnerfire-bm-s-mobo-c	62	20180704-Benton-hp-n-mobo-c
23	20220831-Border32fire-pi-s-mobo-c	63	20180706-FIRE-sm-e-mobo-c
24	20220905-FairviewFire-bh-n-mobo-c	64	20180706-FIRE-sm-n-mobo-c
25	20220905-FairviewFire-smer-tcs3-mobo-c	65	20180706-FIRE-wc-e-mobo-c
26	20220905-FairviewFire-stgo-e-mobo-c	66	20180706-West-lp-n-mobo-c
27	20220905-FairviewFire-tp-w-mobo-c	67	20180717-otay-om-s-mobo-c
28	20230406-FireInMexico-om-s-mobo-c	68	20180718-FIRE-syp-w-mobo-c
29	20230712-FIRE-pi-s-mobo-c	69	20180720-Cinnamon-wc-e-mobo-c
30	20230718-FIRE-bh-w-mobo-c	70	20180723-FIRE-tp-e-mobo-c
31	20230727-BonnyFire-hp-e-mobo-c	71	20180725-Cranston-hp-n-mobo-c
32	20230727-BonnyFire-hp-n-mobo-c	72	20180725-FIRE-smer-tcs10-mobo-c
33	20160604-FIRE-smer-tcs3-mobo-c	73	20180726-FIRE-so-n-mobo-c
34	20160619-FIRE-lp-e-iqeye	74	20180726-FIRE-so-w-mobo-c
35	20160619-FIRE-om-e-mobo-c	75	20180727-FIRE-bh-s-mobo-c
36	20170519-FIRE-rm-w-mobo-c	76	20180727-FIRE-bl-e-mobo-c
37	20170520-FIRE-lp-s-iqeye	77	20180727-FIRE-wc-n-mobo-c
38	20170520-FIRE-pi-w-mobo-c	78	20180728-FIRE-rm-w-mobo-c
39	20170609-FIRE-sm-n-mobo-c	79	20230806-PassFire-ws-n-mobo-c
40	20170625-BBM-bm-n-mobo	80	20230809-BunnieFire-bm-s-mobo-c

Appendix A

RESULTS ON WILDFIRE VIDEOS: CONSECUTIVE FRAME ANALYSIS

A.1 Detection Rate Comparison: With and Without Motion Estimation

TABLE IX: Comparative Analysis of True Detection Rates in a Total of 80 Wildfire Videos Using Consecutive Frames

	Without Motion Estimation	With Motion Estimation	
Video No.	Detection No.	Detection No.	Detection Rate (%)
1	74	74	100.0
2	23	23	100.0
3	149	121	81.20
4	93	93	100.0
5	79	37	46.83
6	145	118	81.37
7	89	47	52.80
8	72	71	98.61
9	37	37	100.0
10	283	274	96.81
11	755	639	84.63
12	68	58	85.29
13	78	77	98.71
14	330	292	88.48
15	85	80	94.11
16	41	21	51.21
17	185	127	68.64
18	102	73	71.56
19	98	74	75.51
20	177	142	80.22
21	227	133	58.59
22	183	163	89.07
23	125	114	91.20
24	45	37	82.22
25	111	92	82.88
26	13	5	38.46
27	30	29	96.66
28	60	60	100.0
29	35	31	88.57
30	160	121	75.62
31	32	32	100.0
32	49	41	83.67
33	394	369	93.65
34	13	9	69.23

Continued on next page

Appendix A (Continued)

Table IX – continued from previous page

	Without Motion Estimation	With Motion Estimation	
Video No.	Detection No.	Detection No.	Detection Rate (%)
35	38	21	55.26
36	58	45	77.58
37	140	98	70.00
38	56	55	98.21
39	46	46	100.0
40	102	59	57.84
41	19	11	57.19
42	325	231	71.07
43	30	30	100.0
44	97	93	90.21
45	68	52	76.47
46	1	1	100.0
47	36	33	91.66
48	21	3	14.28
49	95	91	95.78
50	60	54	90.00
51	28	28	100.0
52	242	232	95.86
53	95	67	70.52
54	60	60	100.0
55	72	71	98.61
56	23	23	100.0
57	185	167	90.27
58	194	186	95.87
59	90	90	100.0
60	172	142	82.55
61	153	140	91.50
62	143	130	90.90
63	79	57	72.15
64	9	9	100.0
65	26	26	100.0
66	147	82	55.78
67	81	58	71.60
68	19	10	52.63
69	23	16	69.56
70	71	38	53.52
71	56	49	87.50
72	223	194	86.99
73	51	50	98.03
74	18	18	100.0
75	20	6	30.00
76	40	12	30.00
77	127	97	76.37
78	208	193	92.78
79	247	191	77.32
80	113	96	84.95
Average	-	-	**81.33**

A.2 Detection Rate Comparison: Classical and DL-based Change Detection

TABLE X: Comparative Analysis of True Detection Rates in a Total of 80 Wildfire Videos Using Consecutive Frames

Video No.	Without CD	With Classical CD		With DL-based CD	
	Detection No.	Detection No.	Detection Rate (%)	Detection No.	Detection Rate (%)
1	74	12	16.21	4	5.40
2	23	9	39.13	5	21.73
3	149	72	48.32	4	2.68
4	93	27	29.03	8	8.60
5	79	34	43.03	19	24.05
6	145	72	49.65	35	24.13
7	89	0	0.00	0	0.00
8	72	11	15.27	5	7.08
9	37	16	43.24	14	37.83
10	283	221	78.09	117	41.34
11	755	230	30.46	175	23.17
12	68	5	7.35	2	2.94
13	78	25	32.05	7	8.97
14	330	166	50.30	72	21.81
15	85	35	41.17	8	9.41
16	41	18	43.90	7	17.07
17	185	29	15.67	5	2.70
18	102	53	51.96	2	1.96
19	98	69	70.40	3	3.06
20	177	56	31.63	33	18.64
21	227	16	7.04	1	0.44
22	183	42	22.95	23	12.56
23	125	53	42.40	24	19.20
24	45	6	13.33	1	2.22
25	111	34	30.63	6	5.40
26	13	3	23.07	3	23.07
27	30	3	10.00	2	6.66
28	60	13	21.66	8	13.33
29	35	0	0.00	0	0.00
30	160	1	0.62	0	0.00
31	32	8	25.00	3	9.37
32	49	7	14.28	4	8.16
33	394	321	81.47	106	26.90
34	13	0	0.00	0	0.00
35	38	7	18.42	0	0.00
36	58	3	5.17	1	1.72
37	140	29	20.71	9	6.42
38	56	7	12.5	1	1.78
39	46	12	26.08	14	30.43
40	102	27	26.47	10	9.80
41	19	3	15.78	0	0.00
42	325	206	63.38	92	28.30

Continued on next page

Appendix A (Continued)

Video No.	Without CD	Classical CD		DL-based CD	
	Detection No.	Detection No.	Detection Rate (%)	Detection No.	Detection Rate (%)
43	30	0	0.00	0	0.00
44	97	24	21.81	13	11.81
45	68	5	7.35	0	0.00
46	1	0	0.00	0	0.00
47	36	0	0.00	0	0.00
48	22	0	0.00	0	0.00
49	97	15	15.15	6	6.06
50	60	0	0.00	0	0.00
51	28	0	0.00	0	0.00
52	242	120	49.58	68	28.09
53	95	6	6.31	0	0.00
54	60	23	38.33	5	8.33
55	72	15	20.83	4	5.55
56	23	16	69.56	6	26.08
57	185	111	60.00	71	38.37
58	194	147	75.77	66	34.02
59	90	39	43.33	19	21.11
60	172	82	47.67	55	31.97
61	153	43	28.10	19	12.41
62	143	9	6.29	4	2.79
63	79	11	13.92	0	0.00
64	9	6	66.66	4	44.44
65	26	4	15.38	0	0.00
66	147	64	43.53	26	17.68
67	81	21	25.92	3	3.70
68	19	0	0.00	0	0.00
69	23	0	0.00	0	0.00
70	71	51	71.83	13	18.30
71	56	5	8.92	1	1.78
72	223	123	55.15	53	23.76
73	51	12	23.52	3	5.88
74	18	5	27.77	2	11.11
75	20	0	0.00	0	0.00
76	40	0	0.00	0	0.00
77	127	15	11.81	6	4.72
78	208	104	50.00	48	23.07
79	247	122	49.39	68	27.53
80	113	5	4.42	1	0.88
Average	-	-	**27.20**	-	**11.22**

Appendix B

RESULTS ON WILDFIRE VIDEOS: FRAME SKIPPING ANALYSIS

B.1 Detection Rate Comparison: With and Without Motion Estimation

TABLE XI: Comparative Analysis of True Detection Rates in a Total of 80 Wildfire Videos Using Frame Skipping

	Without Motion Estimation	With Motion Estimation	
Video No.	Detection No.	Detection No.	Detection Rate (%)
1	73	73	100.0
2	22	22	100.0
3	140	111	79.28
4	86	86	100.0
5	74	33	44.59
6	139	121	87.05
7	85	50	58.82
8	71	70	98.59
9	35	35	100.0
10	271	263	97.04
11	745	652	87.51
12	66	61	92.42
13	74	70	94.59
14	326	267	81.90
15	82	77	93.90
16	41	23	56.09
17	180	131	72.77
18	96	65	67.70
19	94	74	78.72
20	171	134	78.36
21	223	129	57.84
22	176	160	90.90
23	121	105	86.77
24	45	45	100.0
25	109	92	84.40
26	13	6	46.15
27	30	30	100.0
28	57	57	100.0
29	33	28	84.84
30	155	131	84.51
31	34	34	100.0
32	47	41	87.23
33	381	361	94.75
34	13	7	53.84

Continued on next page

Appendix B (Continued)

Table XI – continued from previous page

	Without Motion Estimation	With Motion Estimation	
Video No.	Detection No.	Detection No.	Detection Rate (%)
35	37	33	89.18
36	54	41	75.92
37	115	93	80.86
38	53	52	98.11
39	41	41	100.0
40	96	65	67.70
41	17	9	52.94
42	306	228	74.50
43	26	26	100.0
44	93	93	100.0
45	64	52	81.25
46	1	1	100.0
47	35	32	91.42
48	21	3	14.28
49	92	87	94.56
50	56	55	98.21
51	27	27	100.0
52	235	227	96.59
53	91	66	72.52
54	56	56	100.0
55	72	72	100.0
56	21	21	100.0
57	179	168	93.85
58	188	186	98.93
59	89	89	100.0
60	166	133	80.12
61	149	135	90.60
62	139	130	93.52
63	79	68	86.07
64	8	6	75.00
65	25	25	100.0
66	142	48	33.80
67	79	60	75.94
68	18	13	72.22
69	23	20	86.95
70	68	38	55.88
71	54	52	96.29
72	221	191	86.42
73	49	49	100.0
74	16	16	100.0
75	20	8	40.00
76	39	13	33.33
77	123	107	86.99
78	202	194	96.03
79	239	192	80.33
80	109	101	92.66
Average	-	-	**83.50**

Appendix B (Continued)

B.2 Detection Rate Comparison: Classical and DL-based Change Detection

TABLE XII: Comparative Analysis of True Detection Rates in a Total of 80 Wildfire Videos Using Skipping Frames

Video No.	Without CD	With Classical CD		With DL-based CD	
	Detection No.	Detection No.	Detection Rate (%)	Detection No.	Detection Rate (%)
1	73	29	39.73	14	19.17
2	22	15	68.18	9	40.90
3	140	75	53.57	3	2.14
4	86	44	51.16	10	11.62
5	74	35	47.29	13	17.56
6	139	81	58.27	44	31.65
7	85	0	0.00	0	0.00
8	71	17	23.94	8	11.26
9	35	19	54.28	15	42.85
10	271	216	79.70	132	48.70
11	745	335	44.96	111	14.89
12	66	22	33.33	9	13.63
13	74	38	51.35	7	9.45
14	326	186	57.05	121	37.11
15	82	36	43.90	24	29.26
16	41	17	41.46	6	14.63
17	180	34	18.88	4	2.22
18	96	42	43.75	5	5.20
19	94	61	64.89	2	2.12
20	171	78	45.61	49	28.65
21	223	32	14.34	9	4.03
22	176	61	34.65	42	23.86
23	121	53	43.80	33	27.27
24	45	16	35.55	3	6.66
25	109	41	37.61	10	9.17
26	13	8	61.53	3	23.07
27	30	5	10.00	3	16.66
28	57	26	45.61	14	24.56
29	33	4	12.12	0	0.00
30	155	1	0.64	0	0.00
31	34	5	14.70	6	17.64
32	47	6	12.76	7	14.89
33	381	343	90.02	102	26.77
34	13	0	0.00	0	0.00
35	37	6	16.21	0	0.00
36	54	13	24.07	2	3.70
37	115	39	33.91	18	15.65
38	53	15	28.30	6	11.32
39	41	16	39.02	15	36.58
40	96	25	26.04	5	5.20
41	17	3	17.64	1	5.88
42	306	203	66.33	91	29.73

Continued on next page

Appendix B (Continued)

Video No.	Without CD	With Classical CD		With DL-based CD	
	Detection No.	Detection No.	Detection Rate (%)	Detection No.	Detection Rate (%)
43	26	2	7.69	1	3.84
44	93	53	56.98	32	34.40
45	64	6	9.37	1	1.56
46	1	0	0.00	0	0.00
47	35	0	0.00	0	0.00
48	21	0	0.00	0	0.00
49	92	19	20.65	8	8.69
50	56	0	0.00	0	0.00
51	27	0	0.00	0	0.00
52	235	123	52.34	67	28.51
53	91	10	10.98	3	3.29
54	56	19	33.92	4	7.14
55	72	29	40.27	12	16.66
56	21	14	66.66	17	80.95
57	179	130	72.62	97	54.18
58	188	146	77.65	40	21.27
59	89	58	65.16	21	23.59
60	166	99	59.63	71	42.77
61	149	53	35.57	48	32.21
62	139	19	13.66	2	1.43
63	79	10	12.65	10	12.65
64	8	7	87.50	2	25.00
65	25	5	20.00	4	16.00
66	142	66	46.47	17	11.97
67	79	32	40.50	25	31.64
68	18	4	22.22	0	0.00
69	23	0	0.00	1	4.34
70	68	56	82.35	20	29.41
71	54	16	29.62	8	14.81
72	221	124	56.10	53	23.98
73	49	37	75.51	15	30.61
74	16	5	12.50	2	31.25
75	20	0	0.00	0	0.00
76	39	0	0.00	0	0.00
77	123	18	14.63	8	6.50
78	202	110	54.45	33	16.33
79	239	131	54.81	60	25.10
80	109	36	33.02	13	11.92
Average	-	-	**35.64**	-	**16.72**

Appendix C

RESULTS ON NON-WILDFIRE VIDEOS: CONSECUTIVE FRAME ANALYSIS

C.1 False Alarm Rate Comparison: With and Without Motion Estimation

TABLE XIII: Comparative Analysis of False Alarm Rates in a Total of 80 Non-Wildfire Videos Using Consecutive Frames

	Without Motion Estimation		With Motion Estimation	
Video No.	False-Alarm No.	False-Alarm Rate (%)	False-Alarm No.	False-Alarm Rate (%)
1	187	5.26	0	0.00
2	160	4.50	1	0.02
3	29	0.81	5	0.14
4	87	2.44	46	1.29
5	50	1.40	3	0.08
6	3	0.08	0	0.00
7	2	0.05	2	0.05
8	8	0.22	8	0.22
9	33	0.92	31	0.87
10	23	0.64	21	0.59
11	14	0.39	4	0.11
12	401	11.27	0	0.00
13	85	2.39	0	0.00
14	193	5.42	0	0.00
15	65	1.82	24	1.82
16	7	0.19	5	0.14
17	160	4.50	40	1.12
18	161	4.52	28	0.78
19	8	0.22	0	0.00
20	166	4.66	1	0.02
21	64	1.80	34	0.95
22	66	1.85	0	0.00
23	293	8.24	14	0.39
24	4	0.11	0	0.00
25	16	0.45	4	0.11
26	923	25.96	0	0.00
27	870	24.47	0	0.00
28	1357	38.17	0	0.00
29	4	0.11	0	0.00
30	4	0.11	4	0.11
31	175	4.92	25	0.70

Continued on next page

Appendix C (Continued)

Table Table XIII – continued from previous page

	Without Motion Estimation		With Motion Estimation	
Video No.	False-Alarm No.	False-Alarm Rate (%)	False-Alarm No.	False-Alarm Rate (%)
32	23	0.64	10	0.28
33	48	1.35	1	0.02
34	9	0.25	0	0.00
35	39	1.09	17	0.47
36	32	0.90	1	0.02
37	818	23.00	0	0.00
38	59	1.65	0	0.00
39	372	10.46	0	0.00
40	16	0.45	0	0.00
41	125	3.51	72	2.02
42	28	0.78	4	0.11
43	618	17.38	17	0.47
44	29	0.81	11	0.30
45	820	23.06	0	0.00
46	9	0.25	3	0.08
47	920	25.87	1	0.02
48	893	25.11	0	0.00
49	2	0.05	0	0.00
50	6	0.16	0	0.00
51	3	0.08	0	0.00
52	93	2.61	6	0.16
53	13	0.36	0	0.00
54	793	22.36	24	0.67
55	169	4.75	0	0.00
56	48	1.35	0	0.00
57	199	5.59	4	0.11
58	4	0.11	0	0.00
59	86	2.41	14	0.39
60	154	4.33	3	0.08
61	87	2.44	0	0.00
62	83	2.33	0	0.00
63	33	0.92	4	0.11
64	10	0.28	5	0.14
65	54	1.51	0	0.00
66	666	18.73	2	0.05
67	1	0.02	0	0.00
68	435	12.23	0	0.00
69	571	16.06	21	0.59
70	17	0.47	0	0.00
71	5	0.14	0	0.00
72	65	1.82	6	0.16
73	779	22.47	209	5.87
74	8	0.22	0	0.00
75	2	0.05	0	0.00
76	178	5.00	20	0.56
77	124	3.48	31	0.87
78	251	7.06	100	2.81
79	539	15.16	8	0.22
80	262	7.36	19	0.53
Average	-	**5.70**	-	**0.33**

Appendix C (Continued)

C.2 False Alarm Rate Comparison: Classical and DL-based Change Detection

TABLE XIV: Comparative Analysis of False Alarm Rates in a Total of 80 Non-Wildfire Videos Using Consecutive Frames

	Classical CD		DL-based CD	
Video No.	False-Alarm No.	False-Alarm Rate (%)	False-Alarm No.	False-Alarm Rate (%)
1	0	0.00	0	0.00
2	0	0.00	0	0.00
3	0	0.00	0	0.00
4	44	1.23	6	0.16
5	7	0.19	0	0.00
6	0	0.00	0	0.00
7	0	0.00	0	0.00
8	4	0.11	0	0.00
9	21	0.59	10	0.28
10	2	0.05	2	0.05
11	1	0.02	1	0.02
12	0	0.00	0	0.00
13	1	0.02	0	0.00
14	0	0.00	0	0.00
15	10	0.28	1	0.02
16	0	0.00	0	0.00
17	46	1.29	3	0.08
18	64	1.80	10	0.28
19	0	0.00	0	0.00
20	0	0.00	0	0.00
21	3	0.08	0	0.00
22	0	0.00	0	0.00
23	60	1.68	5	0.14
24	0	0.00	0	0.00
25	0	0.00	0	0.00
26	0	0.00	0	0.00
27	0	0.00	0	0.00
28	0	0.00	0	0.00
29	0	0.00	0	0.00
30	0	0.00	0	0.00
31	52	1.46	13	0.36
32	0	0.00	0	0.00
33	0	0.00	0	0.00
34	0	0.00	0	0.00
35	13	0.36	2	0.05
36	1	0.02	0	0.00
37	0	0.00	0	0.00
38	0	0.00	0	0.00
39	0	0.00	0	0.00
40	0	0.00	0	0.00
41	88	2.47	42	1.18
42	1	0.02	0	0.00

Continued on next page

Appendix C (Continued)

Table Table XIV – continued from previous page

	Classical CD		DL-based CD	
Video No.	False-Alarm No.	False-Alarm Rate (%)	False-Alarm No.	False-Alarm Rate (%)
43	54	1.51	17	0.47
44	0	0.00	0	0.00
45	0	0.00	0	0.00
46	0	0.00	0	0.00
47	1	0.02	1	0.02
48	0	0.00	0	0.00
49	0	0.00	0	0.00
50	0	0.00	0	0.00
51	0	0.00	0	0.00
52	0	0.00	0	0.00
53	0	0.00	0	0.00
54	0	0.00	0	0.00
55	0	0.00	0	0.00
56	0	0.00	0	0.00
57	81	2.27	2	0.05
58	0	0.00	0	0.00
59	0	0.00	0	0.00
60	2	0.05	0	0.00
61	0	0.00	0	0.00
62	0	0.00	0	0.00
63	16	0.45	0	0.00
64	0	0.00	0	0.00
65	0	0.00	0	0.00
66	0	0.00	0	0.00
67	0	0.00	0	0.00
68	0	0.00	0	0.00
69	0	0.00	0	0.00
70	0	0.00	0	0.00
71	0	0.00	0	0.00
72	0	0.00	0	0.00
73	228	6.41	69	1.94
74	0	0.00	0	0.00
75	0	0.00	0	0.00
76	6	0.16	0	0.00
77	106	2.98	31	0.88
78	191	5.37	22	0.61
79	154	4.33	26	0.73
80	85	2.39	37	1.04
Average	-	**0.47**	-	**0.10**

Appendix D

RESULTS ON NON-WILDFIRE VIDEOS: FRAME SKIPPING ANALYSIS

D.1 False Alarm Rate Comparison: With and Without Motion Estimation

TABLE XV: Comparative Analysis of False Alarm Rates in a Total of 80 Non-Wildfire Videos Using Frame Skipping

	Without Motion Estimation		With Motion Estimation	
Video No.	False-Alarm No.	False-Alarm Rate (%)	False-Alarm No.	False-Alarm Rate (%)
1	184	5.24	0	0.00
2	156	4.44	1	0.02
3	29	0.82	7	0.19
4	82	2.33	44	1.25
5	48	1.36	3	0.08
6	3	0.08	1	0.02
7	2	0.05	2	0.05
8	7	0.19	6	0.17
9	33	0.94	32	0.91
10	22	0.62	20	0.56
11	14	0.39	7	0.19
12	398	11.33	0	0.00
13	82	2.33	2	0.05
14	191	5.44	0	0.00
15	64	1.82	28	0.79
16	7	0.19	5	0.14
17	157	4.47	29	0.82
18	153	4.35	17	0.48
19	8	0.22	0	0.00
20	165	4.70	2	0.05
21	64	1.82	45	1.28
22	66	1.88	4	0.11
23	291	8.29	27	0.76
24	4	0.11	0	0.00
25	14	0.39	7	0.19
26	910	25.92	0	0.00
27	853	24.30	0	0.00
28	1336	38.06	0	0.00
29	4	0.11	0	0.00
30	4	0.11	4	0.11
31	171	4.87	32	0.91

Continued on next page

Appendix D (Continued)

Table Table XV – continued from previous page

	Without Motion Estimation		With Motion Estimation	
Video No.	False-Alarm No.	False-Alarm Rate (%)	False-Alarm No.	False-Alarm Rate (%)
32	23	0.65	15	0.42
33	48	1.36	5	0.14
34	9	0.25	0	0.00
35	39	1.11	25	0.71
36	32	0.91	1	0.02
37	811	23.10	0	0.00
38	59	1.68	3	0.08
39	368	10.48	3	0.08
40	15	0.42	0	0.00
41	116	3.30	76	2.16
42	28	0.79	12	0.34
43	611	17.40	44	1.25
44	29	0.82	11	0.31
45	810	23.07	7	0.19
46	9	0.25	4	0.11
47	909	25.89	9	0.25
48	884	25.18	0	0.00
49	2	0.05	0	0.00
50	6	0.17	1	0.02
51	3	0.08	0	0.00
52	89	2.53	17	0.48
53	13	0.37	0	0.00
54	784	22.33	104	2.96
55	165	4.70	2	0.05
56	47	1.33	0	0.00
57	197	5.61	2	0.05
58	4	0.11	0	0.00
59	85	2.42	19	0.54
60	149	4.24	1	0.02
61	87	2.47	0	0.00
62	80	2.27	0	0.00
63	33	0.94	1	0.02
64	10	0.28	5	0.14
65	54	1.53	0	0.00
66	659	18.77	3	0.08
67	1	0.02	0	0.00
68	424	12.07	0	0.00
69	563	16.03	42	1.19
70	17	0.48	0	0.00
71	5	0.14	0	0.00
72	63	1.79	6	0.17
73	793	22.59	280	7.97
74	8	0.22	0	0.00
75	2	0.05	0	0.00
76	173	4.92	21	0.59
77	124	3.53	27	0.76
78	250	7.12	61	1.73
79	539	15.35	1	0.02
80	259	7.37	9	0.25
Average	-	**5.69**	-	**0.40**

D.2 False Alarm Rate Comparison: Classical and DL-based Change Detection

TABLE XVI: Comparative Analysis of False Alarm Rates in a Total of 80 Non-Wildfire Videos Using Frame Skipping

	Classical CD		DL-based CD	
Video No.	False-Alarm No.	False-Alarm Rate (%)	False-Alarm No.	False-Alarm Rate (%)
1	0	0.00	0	0.00
2	2	0.05	0	0.00
3	4	0.11	0	0.00
4	31	0.88	6	0.17
5	8	0.22	1	0.02
6	0	0.00	0	0.00
7	0	0.00	0	0.00
8	4	0.11	3	0.08
9	21	0.59	14	0.39
10	5	0.14	2	0.05
11	2	0.05	1	0.02
12	0	0.00	0	0.00
13	8	0.22	0	0.00
14	0	0.00	0	0.00
15	17	0.48	5	0.14
16	1	0.02	0	0.00
17	44	1.25	4	0.11
18	66	1.88	31	0.88
19	8	0.22	0	0.00
20	16	0.45	0	0.00
21	3	0.08	0	0.00
22	0	0.00	0	0.00
23	117	3.33	19	0.54
24	0	0.00	0	0.00
25	0	0.00	0	0.00
26	0	0.00	0	0.00
27	0	0.00	0	0.00
28	0	0.00	0	0.00
29	0	0.00	0	0.00
30	0	0.00	0	0.00
31	75	2.13	11	0.31
32	0	0.00	0	0.00
33	0	0.00	0	0.00
34	0	0.00	0	0.00
35	19	0.54	1	0.02
36	16	0.45	2	0.05
37	0	0.00	0	0.00
38	0	0.00	0	0.00
39	1	0.02	0	0.00
40	0	0.00	0	0.00
41	110	3.13	21	0.59
42	1	0.02	1	0.02

Continued on next page

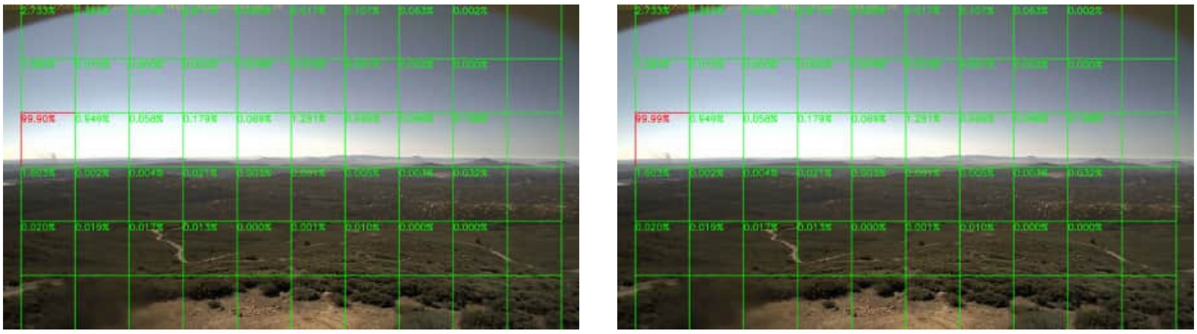
Appendix D (Continued)

Table Table XVI – continued from previous page

	Classical CD		DL-based CD	
Video No.	False-Alarm No.	False-Alarm Rate (%)	False-Alarm No.	False-Alarm Rate (%)
43	256	7.29	54	1.53
44	0	0.00	0	0.00
45	0	0.00	0	0.00
46	2	0.05	0	0.00
47	1	0.02	1	0.02
48	0	0.00	0	0.00
49	0	0.00	0	0.00
50	0	0.00	0	0.00
51	0	0.00	0	0.00
52	3	0.25	0	0.00
53	0	0.00	0	0.00
54	17	0.48	0	0.00
55	0	0.00	0	0.00
56	0	0.00	0	0.00
57	84	2.39	8	0.22
58	0	0.00	0	0.00
59	10	0.28	0	0.00
60	2	0.05	0	0.00
61	1	0.02	0	0.00
62	0	0.00	0	0.00
63	29	0.82	4	0.11
64	9	0.25	0	0.00
65	0	0.00	0	0.00
66	0	0.00	0	0.00
67	0	0.00	0	0.00
68	0	0.00	0	0.00
69	3	0.08	0	0.00
70	0	0.00	0	0.00
71	0	0.00	0	0.00
72	8	0.22	1	0.05
73	337	9.60	49	1.39
74	0	0.00	0	0.00
75	0	0.00	0	0.00
76	27	0.76	6	0.17
77	98	2.79	8	0.22
78	170	4.84	16	0.45
79	260	7.40	14	0.39
80	141	4.01	54	1.53
Average	-	**0.72**	-	**0.11**

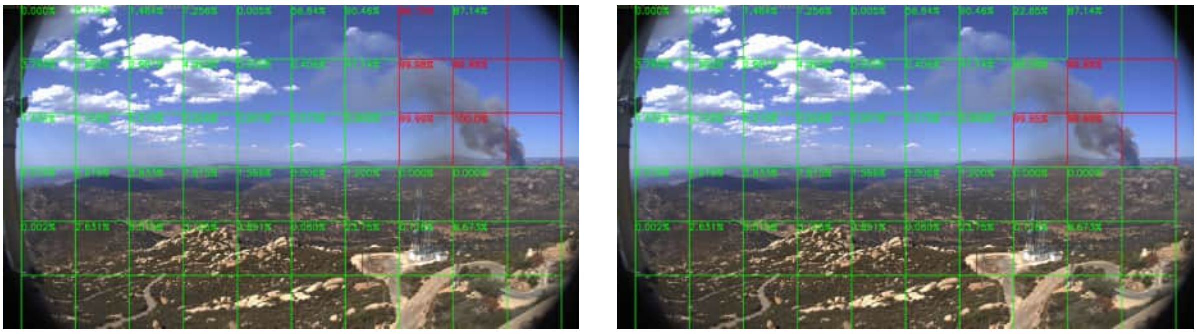
Appendix E

RESULTS ON WILDFIRE VIDEOS



(a) Before integrating motion estimation

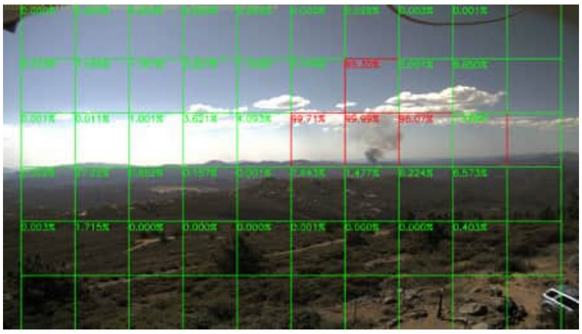
(b) After integrating motion estimation



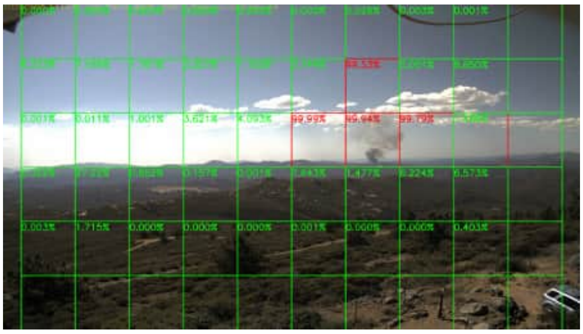
(a) Before integrating motion estimation

(b) After integrating motion estimation

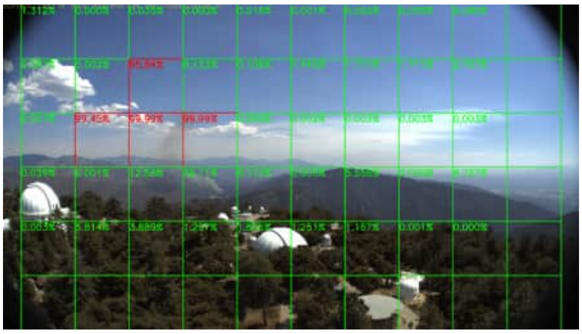
Appendix E (Continued)



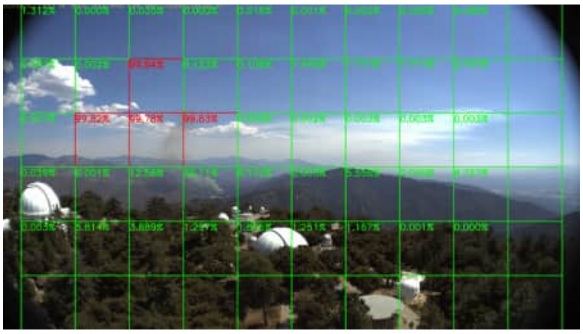
(a) Before integrating motion estimation



(b) After integrating motion estimation

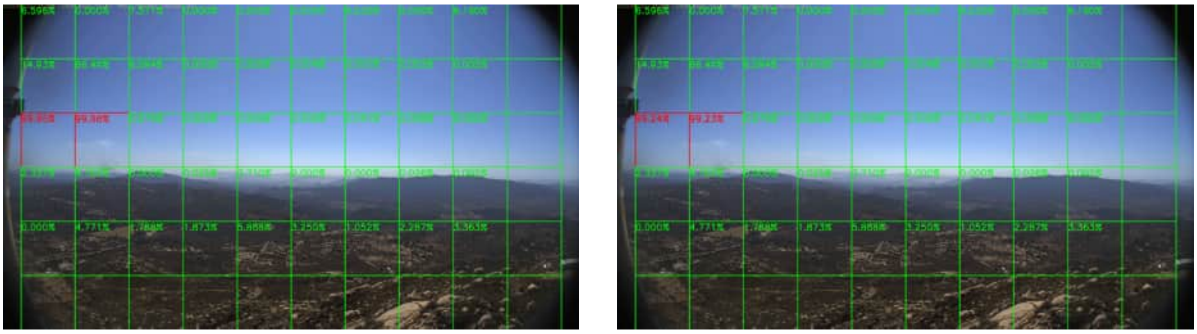


(a) Before integrating motion estimation



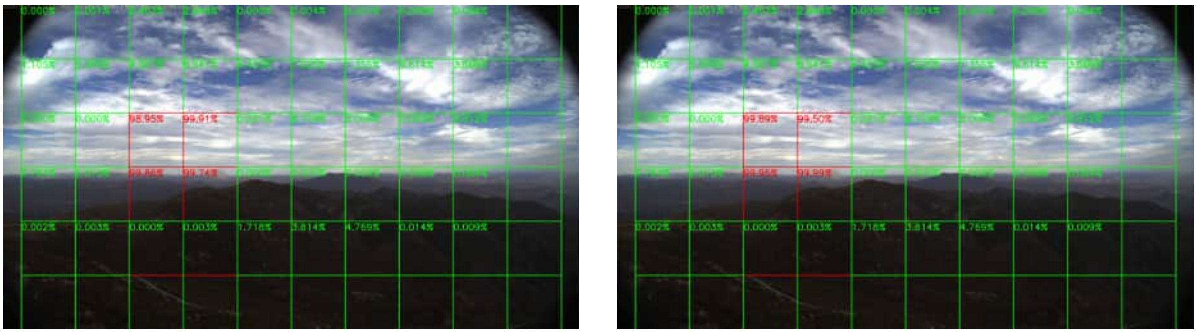
(b) After integrating motion estimation

Appendix E (Continued)



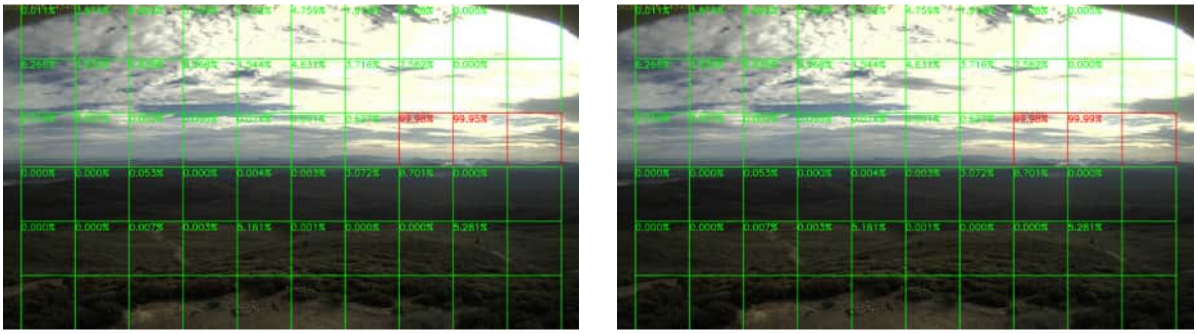
(a) Before integrating motion estimation

(b) After integrating motion estimation



(a) Before integrating motion estimation

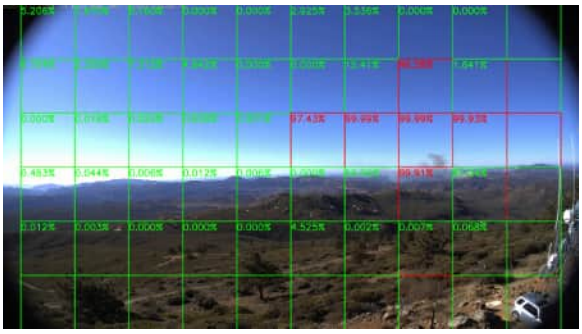
(b) After integrating motion estimation



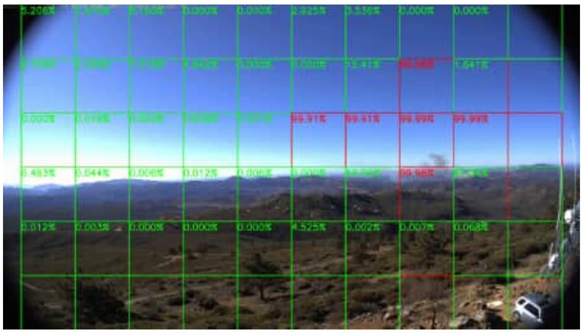
(a) Before integrating motion estimation

(b) After integrating motion estimation

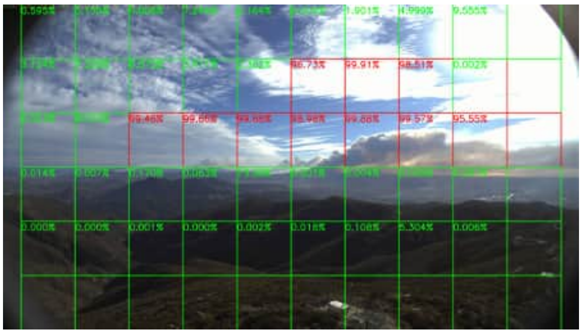
Appendix E (Continued)



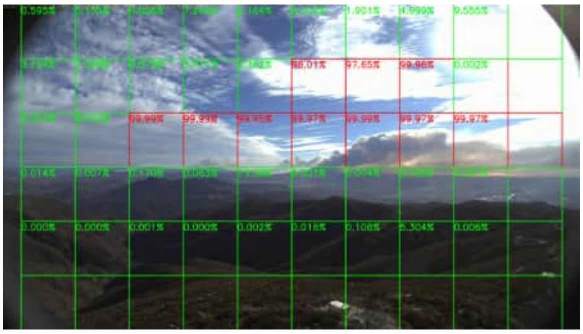
(a) Before integrating motion estimation



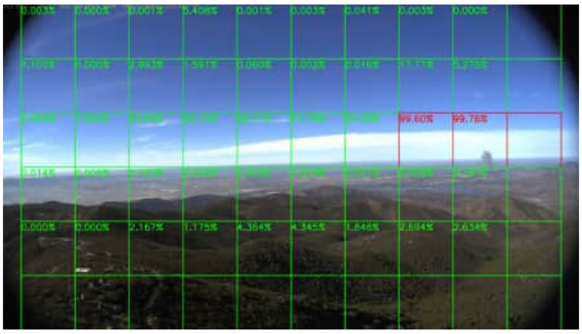
(b) After integrating motion estimation



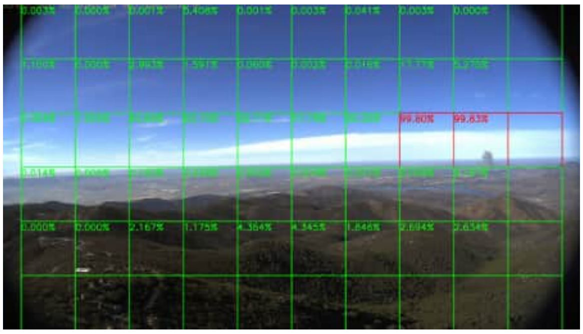
(a) Before integrating motion estimation



(b) After integrating motion estimation



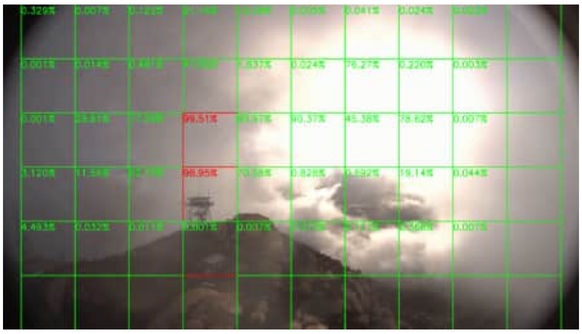
(a) Before integrating motion estimation



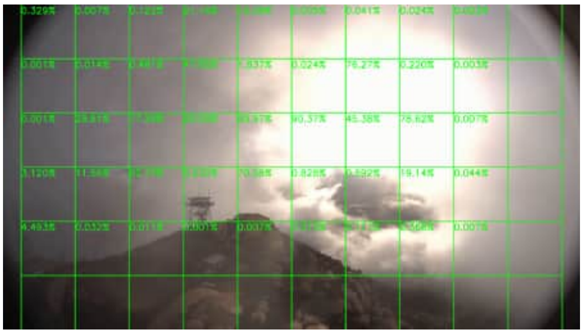
(b) After integrating motion estimation

Appendix F

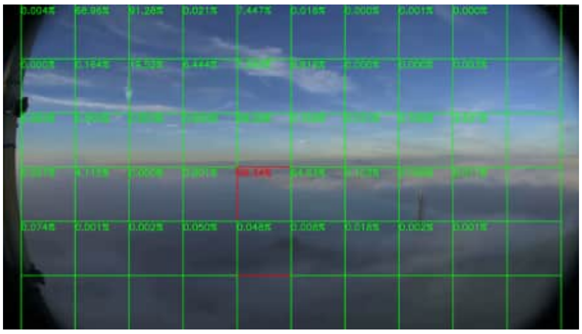
RESULTS ON NON-WILDFIRE VIDEOS



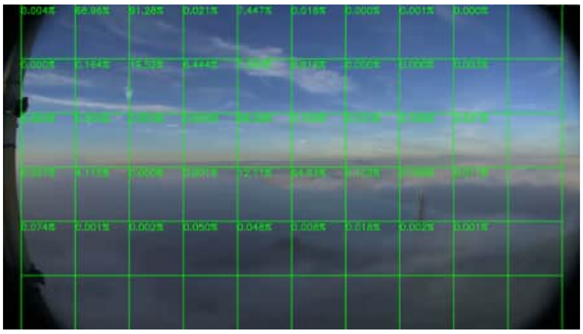
(a) Before integrating motion estimation



(b) After integrating motion estimation

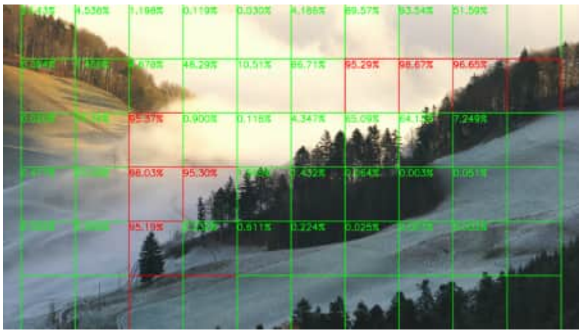


(a) Before integrating motion estimation

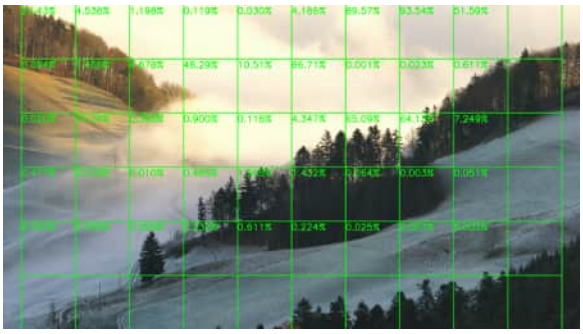


(b) After integrating motion estimation

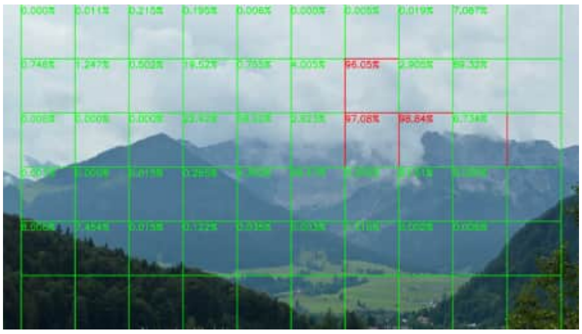
Appendix F (Continued)



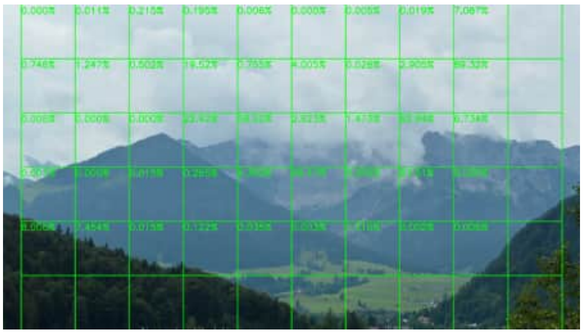
(a) Before integrating motion estimation



(b) After integrating motion estimation

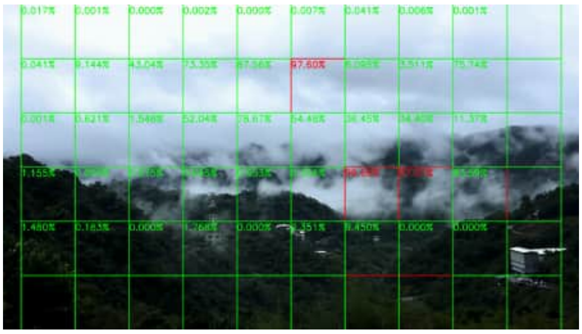


(a) Before integrating motion estimation

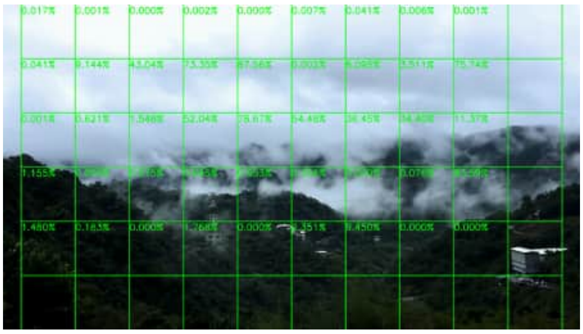


(b) After integrating motion estimation

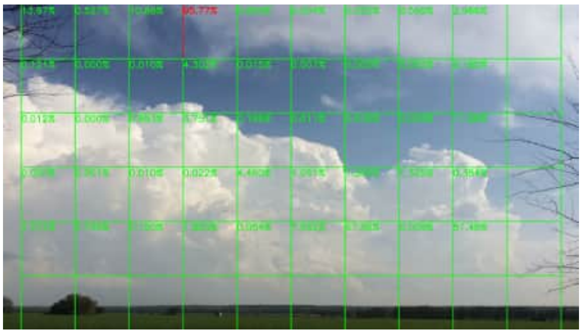
Appendix F (Continued)



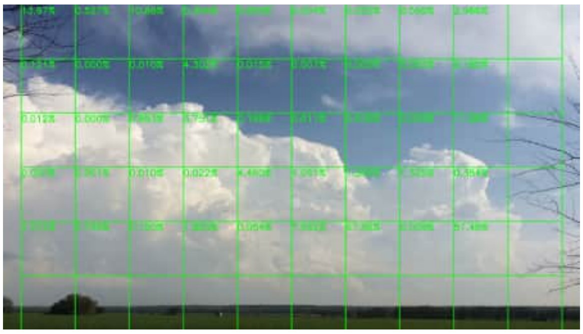
(a) Before integrating motion estimation



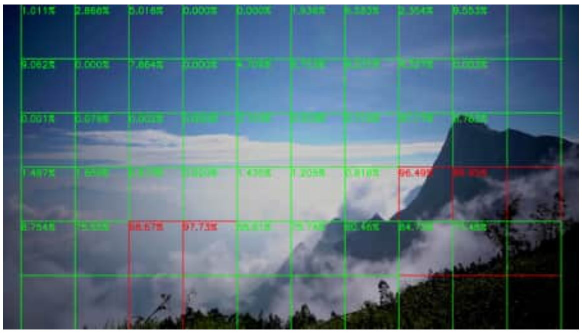
(b) After integrating motion estimation



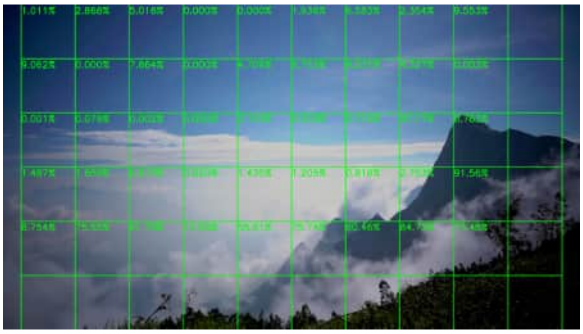
(a) Before integrating motion estimation



(b) After integrating motion estimation

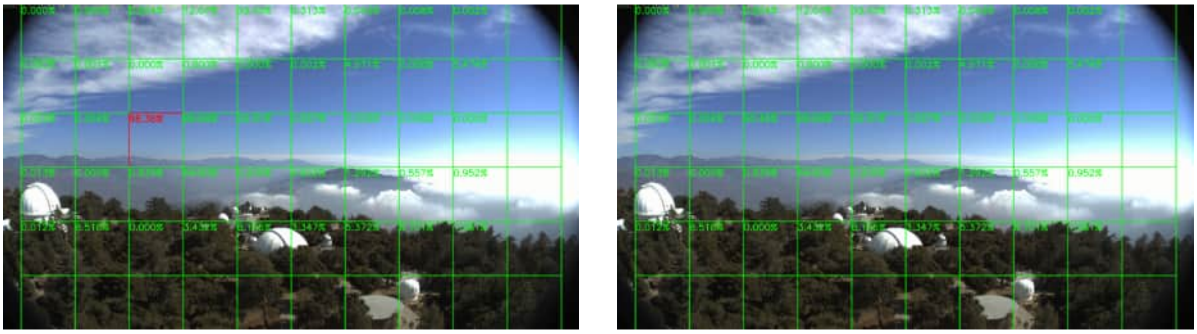


(a) Before integrating motion estimation



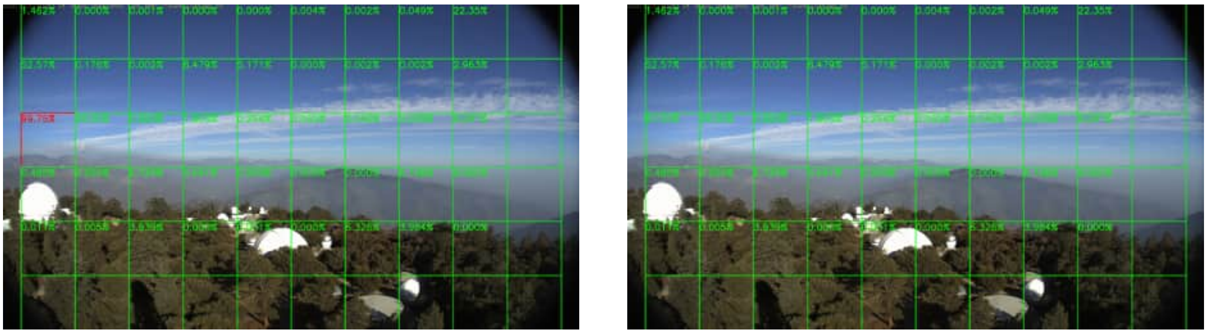
(b) After integrating motion estimation

Appendix F (Continued)



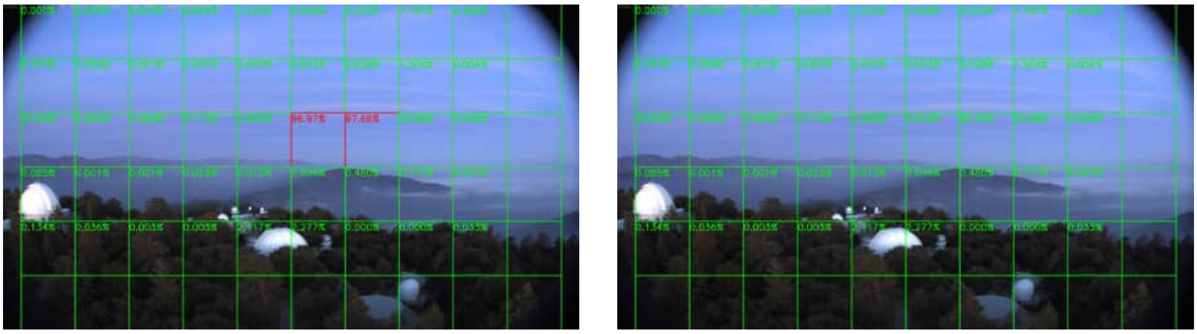
(a) Before integrating motion estimation

(b) After integrating motion estimation



(a) Before integrating motion estimation

(b) After integrating motion estimation



(a) Before integrating motion estimation

(b) After integrating motion estimation

CITED LITERATURE

1. Yan, T., Wan, Z., and Zhang, P.: Fully transformer network for change detection of remote sensing images, 2022.

2. Ramos, L., Casas, E., Bendek, E., Romero, C., and Rivas-Echeverría, F.: Computer vision for wildfire detection: a critical brief review. Multimedia Tools and Applications, pages 1–44, 2024.

3. Hong, Z., Hamdan, E., Zhao, Y., Ye, T., Pan, H., and Cetin, A. E.: Wildfire detection via transfer learning: A survey, 2023.

4. Pan, H., Badawi, D., and Cetin, A. E.: Computationally efficient wildfire detection method using a deep convolutional network pruned via fourier analysis. Sensors, 20(10):2891, 2020.

5. Mohapatra, A. and Trinh, T.: Early wildfire detection technologies in practice—a review. Sustainability, 14(19), 2022.

6. Ulucinar, A. R., Korpeoglu, I., and Cetin, A. E.: A wi-fi cluster based wireless sensor network application and deployment for wildfire detection. International Journal of Distributed Sensor Networks, 10(10):651957, 2014.

7. Varela, N., Jorge L, D.-M., Ospino, A., and Lizardo Zelaya, N. A.: Wireless sensor network for forest fire detection. Procedia Computer Science, 175:435–440, 2020. The 17th International Conference on Mobile Systems and Pervasive Computing (MobiSPC),The 15th International Conference on Future Networks and Communications (FNC),The 10th International Conference on Sustainable Energy Information Technology.

8. Khalid, W., Sattar, A., Qureshi, M. A., Amin, A., Malik, M. A., and Memon, K. H.: A smart wireless sensor network node for fire detection. Turkish J. Electr. Eng. Comput. Sci., 27:2541–2556, 2019.

9. Ammar, M. and Souissi, R.: A new approach based on wireless sensor network and fuzzy logic for forest fire detection. Int. J. Comput. Appl., 89:0975–8887, 2014.

10. Bolourchi, P. and Uysal, S.: Forest fire detection in wireless sensor network using fuzzy logic. In 2013 Fifth International Conference on Computational Intelligence, Communication Systems and Networks, pages 83–87, 2013.

11. Group, N. W. C.: NWCG Standards for Fire Unmanned Aircraft Systems Operations. Washington, DC, USA, National Wildfire Coordinating Group, 2019.

12. Zhao, X., Zhou, Z., Zhu, X., and Guo, A.: Design of a hand-launched solar-powered unmanned aerial vehicle (uav) system for plateau. Applied Sciences, 10(4), 2020.

13. Allison, R. S., Johnston, J. M., Craig, G., and Jennings, S.: Airborne optical and thermal remote sensing for wildfire detection and monitoring. Sensors, 16(8), 2016.

14. Yuan, C., Liu, Z., and Zhang, Y.: Uav-based forest fire detection and tracking using image processing techniques. In 2015 International Conference on Unmanned Aircraft Systems (ICUAS), pages 639–643, 2015.

15. Shi, J., Wang, W., Gao, Y., and Yu, N.: Optimal placement and intelligent smoke detection algorithm for wildfire-monitoring cameras. IEEE Access, 8:72326–72339, 2020.

16. Alert wildfire. `https://www.alertwildfire.org/`, 2014. Accessed on 27 July 2022.

17. Pacific Gas and Electric: News releases, pacific gas and electric. `https://www.pge.com/en_US/about-pge/medianewsroom/news-details.page?pageID=0553327b-df9e-4321-9b19-92b9297ec2d4&ts=1642273313274`, 2021. Accessed on 27 July 2022.

18. Pano ai. `https://www.pano.ai/`, 2019. Accessed on 27 July 2022.

19. Planet: Linking ground and space systems to autonomously assess wildfires. `https://learn.planet.com/rs/997-CHH-265/images/2020-08-25_Moore_MOFD_Case%20Study_Letter.pdf`, 2020. Accessed on 10 July 2022.

20. Interorbital systems. `https://www.interorbital.com/`, 2022. Accessed on 24 July 2022.

21. Afghah, F., Razi, A., Chakareski, J., and Ashdown, J.: Wildfire monitoring in remote areas using autonomous unmanned aerial vehicles. In IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pages 835–840, 2019.

22. Chen, X., Hopkins, B., Wang, H., O'Neill, L., Afghah, F., Razi, A., Fulé, P., Coen, J., Rowell, E., and Watts, A.: Wildland fire detection and monitoring using a drone-collected rgb/ir image dataset. IEEE Access, 10:121301–121317, 2022.

23. Boroujeni, S. P. H., Razi, A., Khoshdel, S., Afghah, F., Coen, J. L., ONeill, L., Fulé, P., Watts, A., Kokolakis, N.-M. T., and Vamvoudakis, K. G.: A comprehensive survey of research towards ai-enabled unmanned aerial systems in pre-, active-, and post-wildfire management. ArXiv, abs/2401.02456, 2024.

24. Chen, W., Zhou, Y., Zhou, E., Xiang, Z., Zhou, W., and Lu, J.: Wildfire risk assessment of transmission-line corridors based on naïve bayes network and remote sensing data. Sensors, 21(2), 2021.

25. Bar, S., Parida, B. R., Pandey, A. C., Shankar, B. U., Kumar, P., Panda, S. K., and Behera, M. D.: Modeling and prediction of fire occurrences along an elevational gradient in western himalayas. Applied Geography, 151:102867, 2023.

26. Jaafari, A., Zenner, E. K., and Pham, B. T.: Wildfire spatial pattern analysis in the zagros mountains, iran: A comparative study of decision tree based classifiers. Ecological Informatics, 43:200–211, 2018.

27. Makowski, D.: Simple random forest classification algorithms for predicting occurrences and sizes of wildfires. Extremes, 26, 12 2022.

28. Zhao, Y., Li, Q., and Gu, Z.: Early smoke detection of forest fire video using cs adaboost algorithm. Optik - International Journal for Light and Electron Optics, 126(19):2121–2124, 2015.

29. Zwirglmaier, K., Papakosta, P., and traub, D.: Learning a bayesian network model for predicting wildfire behavior. 2013.

30. Rodrigues, M., Jiménez-Ruano, A., Peña-Angulo, D., and de la Riva, J.: A comprehensive spatial-temporal analysis of driving factors of human-caused wildfires in spain using geographically weighted logistic regression. Journal of Environmental Management, 225:177–192, 2018.

31. Rezaei Barzani, A., Pahlavani, P., and Ghorbanzadeh, O.: Ensembling of Decision Trees, Knn, and Logistic Regression with Soft-Voting Method for Wildfire Susceptibility Mapping. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, 14W1:647–652, January 2023.

32. Pecha, M., Langford, Z. L., Hořák, D., and Mills, R. T.: Wildfires identification: Semantic segmentation using support vector machine classifier. Programs and Algorithms of Numerical Mathematics 21, 2023.

33. Abu, M. A., Indra, N. H. B., Rahman, A. H. A., Sapiee, N. A., and Ahmad, I.: A study on image classification based on deep learning and tensorflow. 2019.

34. Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q.: Densely connected convolutional networks. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2261–2269, 2017.

35. Tan, M. and Le, Q. V.: Efficientnet: Rethinking model scaling for convolutional neural networks, 2020.

36. Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications, 2017.

37. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A.: Going deeper with convolutions, 2014.

38. Ronneberger, O., Fischer, P., and Brox, T.: U-net: Convolutional networks for biomedical image segmentation, 2015.

39. Badrinarayanan, V., Handa, A., and Cipolla, R.: Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling, 2015.

40. Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs, 2017.

41. Redmon, J., Divvala, S., Girshick, R., and Farhadi, A.: You only look once: Unified, real-time object detection, 2016.

42. Redmon, J. and Farhadi, A.: Yolo9000: Better, faster, stronger, 2016.

43. Liu, H., Sun, F., Gu, J. J., and Deng, L.: Sf-yolov5: A lightweight small object detection algorithm based on improved feature fusion mode. Sensors (Basel, Switzerland), 22, 2022.

44. Li, C., Li, L., Jiang, H., Weng, K., Geng, Y., Li, L., Ke, Z., Li, Q., Cheng, M., Nie, W., Li, Y., Zhang, B., Liang, Y., Zhou, L., Xu, X., Chu, X., Wei, X., and Wei, X.: Yolov6: A single-stage object detection framework for industrial applications, 2022.

45. Wang, C.-Y., Bochkovskiy, A., and Liao, H.-Y. M.: Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, 2022.

46. Wang, Y., Wu, A., Zhang, J., Zhao, M., Li, W., and Dong, N.: Fire smoke detection based on texture features and optical flow vector of contour. 2016 12th World Congress on Intelligent Control and Automation (WCICA), pages 2879–2883, 2016.

47. Avalhais, L. P. S., Rodrigues, J. F., and Traina, A. J. M.: Fire detection on unconstrained videos using color-aware spatial modeling and motion flow. 2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI), pages 913–920, 2016.

48. Yuan, C., Liu, Z., and Zhang, Y.: Aerial images-based forest fire detection for firefighting using optical remote sensing techniques and unmanned aerial vehicles. Journal of Intelligent & Robotic Systems, 88:635 – 654, 2017.

49. Yuan, C., Liu, Z., and Zhang, Y.: Fire detection using infrared images for uav-based forest fire surveillance. In 2017 International Conference on Unmanned Aircraft Systems (ICUAS), pages 567–572, 2017.

50. Hu, C., Tang, P., Jin, W., He, Z., and Li, W.: Real-time fire detection based on deep convolutional long-recurrent networks and optical flow method. In 2018 37th Chinese Control Conference (CCC), pages 9061–9066, 2018.

51. Dang-Ngoc, H. and Nguyen-Trung, H.: Evaluation of forest fire detection model using video captured by uavs. In 2019 19th International Symposium on Communications and Information Technologies (ISCIT), pages 513–518, 2019.

52. Gupta, T., Liu, H., and Bhanu, B.: Early wildfire smoke detection in videos. In 2020 25th International Conference on Pattern Recognition (ICPR), pages 8523–8530, 2021.

53. Wahyono, Harjoko, A., Dharmawan, A., Adhinata, F. D., Kosala, G., and Jo, K.-H.: Real-time forest fire detection framework based on artificial intelligence using color probability model and motion feature analysis. Fire, 2022.

54. Peng, Y. and Wang, Y.: Automatic wildfire monitoring system based on deep learning. European Journal of Remote Sensing, 55(1):551–567, 2022.

55. Shahid, M., Wang, H.-C., Chen, Y.-Y., and Hua, K.-L.: Hybrid cnn-vit architecture to exploit spatio-temporal feature for fire recognition trained through transfer learning. Multimedia Tools and Applications, pages 1–30, 2024.

56. Girshick, R., Donahue, J., Darrell, T., and Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation, 2014.

57. Tan, M. and Le, Q.: Efficientnet: Rethinking model scaling for convolutional neural networks. In International conference on machine learning, pages 6105–6114. PMLR, 2019.

58. Tan, M. and Le, Q. V.: Efficientnetv2: Smaller models and faster training. In International Conference on Machine Learning, pages 10096–10106. PMLR, 2021.

59. Wedel, A. and Cremers, D.: Stereo scene flow for 3d motion analysis. 2011.

60. Farnebäck, G.: Two-frame motion estimation based on polynomial expansion. In Scandinavian Conference on Image Analysis, 2003.

61. He, K., Zhang, X., Ren, S., and Sun, J.: Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.

62. Krizhevsky, A., Sutskever, I., and Hinton, G. E.: Imagenet classification with deep convolutional neural networks. Communications of the ACM, 60:84 – 90, 2012.

63. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L.: Imagenet large scale visual recognition challenge, 2015.

64. University of California San Diego, California, A.: The high performance wireless research and education network., 2019. Accessed December 25, 2022.

65. Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., and He, Q.: A comprehensive survey on transfer learning. Proceedings of the IEEE, 109(1):43–76, 2021.

66. He, Z., Zhang, Z., Feng, H., and Wang, L.: The application of wavelet transform and the adaptive threshold segmentation in image change detection. Advanced Materials Research, 709:547 – 550, 2013.

67. Radke, R., Andra, S., Al-Kofahi, O., and Roysam, B.: Image change detection algorithms: a systematic survey. IEEE Transactions on Image Processing, 14(3):294–307, 2005.

VITA

FATEMEH TAGHVAEI

EDUCATION	M.Sc in Electrical and Computer Engineering, University of Illinois Chicago, USA, 2024
	B.Sc. in Computer Engineering, University of Isfahan, Iran, 2021