

PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA  
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH  
MOHAMED KHIDER UNIVERSITY OF BISKRA



Faculty of Exact Sciences  
**Computer Science Department**  
Dissertation

Presented to obtain the academic master's degree in Computer Science  
Option: Information Systems, Optimization, and Decision (SIOD)  
Theme

---

# Wildfire Detection in Algeria Using Satellite Imagery

---

Presented by:  
**Ouamane Selsabil**  
**Sayad Maroua**  
**Megherbi Sara Anghem**

Under the supervision of:  
**MEADI Mohamed Nadjib**  
**ZERARI Abdelmoumen**

Graduate on 18<sup>th</sup> June in front of the JURY:

**President:** GUEMEIDA Abdelbasset – Computer Science Department  
**Examiner:** ZERDOUMI Oussama – Computer Science Department  
**Examiner:** DJELLAB Mohamed – Management Sciences Department

**Academic year 2024-2025**

# Acknowledgment

We would like to thank God Almighty, whose boundless grace and mercy have been the guiding light throughout this journey. Without His strength and guidance, none of this would have been possible. Every success, every step forward, is a blessing from Him, and for that, we are eternally grateful.

Our deepest gratitude also goes to our supervisor, ***Dr. MEADI Mohamed Nadjib***, whose unwavering support, insightful guidance, and valuable advice have made this work possible. We are particularly grateful for his suggestion of this research topic, which has been both inspiring and enlightening throughout the process.

We would like to thank the committee for their constructive feedback, which contributed to strengthening this work. A special thanks goes to ***Dr.DJELLAB Mohamed*** for his thoughtful and well-placed critiques. His encouragement to improve and carry this work forward was truly motivating, and his perspective added real value to the development of this research.

We are profoundly thankful to our parents and families, whose love, support, and encouragement have been the foundation upon which we have built our dreams. Their belief in us and constant presence have given us the strength to move forward, stay grounded, and remain motivated throughout this journey. We are truly grateful for for that.

We would like to extend our sincere thanks to everyone who has contributed to this work, whether through direct assistance, advice, or inspiration. Your support has been invaluable, and we are deeply appreciative of your help.

***Ouamane Selsabil***

***Sayad Maroua***

***Megherbi Sara Anghem***

# Abstract

Wildfires have become an increasingly urgent global issue, particularly in regions like Algeria, where they cause significant environmental, social, and economic damage. Traditional wildfire detection methods, such as ground observation and aerial surveys, are often limited in coverage and response time, making it crucial to explore advanced solutions for real-time monitoring. In this project, we utilize satellite remote sensing and deep learning techniques to enhance wildfire detection. Using the *Active Fire Detection in Landsat-8 Imagery* dataset, we conducted a comparative analysis of three deep learning models U-Net, SegFormer B2, and YOLO v8. For segmentation tasks, we used the Murphy Mask as our ground truth label. After evaluating the performance of these models, we selected the best one, with U-Net and SegFormer B2 demonstrating the most promising results in accurately detecting fire boundaries and hotspots. These segmentation models provided more precise and dynamic detection compared to the others. Finally, the chosen model was deployed in a web application for early wildfire detection.

**Keywords:** Wildfire, Deep Learning, Satellite Imagery, Remote Sensing, U-Net, SegFormer-B2, YOLOv8, Image Segmentation, Early Detection, Active Fire, Murphy Mask.

# Résumé

Les incendies de forêt sont devenus un problème mondial de plus en plus urgent, en particulier dans des régions comme l'Algérie, où ils causent des dommages environnementaux, sociaux et économiques considérables. Les méthodes traditionnelles de détection des incendies, telles que l'observation au sol et les relevés aériens, sont souvent limitées en termes de couverture et de réactivité, ce qui rend crucial l'exploration de solutions avancées pour une surveillance en temps réel. Dans ce projet, nous utilisons la télédétection par satellite et les techniques d'apprentissage profond pour améliorer la détection des incendies. En utilisant le jeu de données *Active Fire Detection in Landsat-8 Imagery*, nous avons réalisé une analyse comparative de trois modèles d'apprentissage profond : U-Net, SegFormer B2 et YOLO v8. Pour les tâches de segmentation, nous avons utilisé le Murphy Mask comme étiquette de vérité de terrain. Après avoir évalué la performance de ces modèles, nous avons sélectionné le meilleur, avec U-Net et SegFormer B2 montrant les résultats les plus prometteurs pour détecter avec précision les limites des incendies et les foyers chauds. Ces modèles de segmentation ont fourni une détection plus précise et dynamique par rapport aux autres. Enfin, le modèle choisi a été déployé dans une application web pour une détection précoce des incendies de forêt.

**Mots-clés :** Incendie de forêt, Apprentissage profond, Imagerie satellite, Télédétection, U-Net, SegFormer-B2, YOLOv8, Segmentation d'images, Détection précoce, Feu actif, Murphy Mask.





# Contents

Acknowledgment	i
Abstract	ii
Résumé	iii
Contents	v
List of Figures	viii
List of Tables	xi
List of Abbreviations	xii
Introduction	1
<b>1 Satellite Remote Sensing for Wildfire Detection</b>	<b>3</b>
1.1 Introduction . . . . .	4
1.2 Remote Sensing Technologies . . . . .	4
1.2.1 Evolution of Remote Sensing . . . . .	5
1.2.2 Principles of Remote Sensing . . . . .	7
1.2.3 Types of Remote Sensing . . . . .	9
1.3 Satellite Imagery for Wildfire Detection . . . . .	13
1.3.1 Satellite Imagery Overview . . . . .	14
1.3.2 Wildfire Detection . . . . .	19
1.3.2.1 Wildfire Statistics . . . . .	19
1.3.2.2 Traditional Wildfire Detection Techniques in Algeria . . . . .	25

1.3.3	The role of satellite imagery in wildfire detection . . . . .	26
1.3.4	Active fire satellite image Acquisition . . . . .	27
1.3.5	Limitations and Challenges of Satellite-Based Fire Monitoring . . . . .	29
1.4	Related Works . . . . .	30
1.5	Conclusion . . . . .	31
<b>2</b>	<b>Deep Learning Models for Satellite-Based Wildfire Detection</b>	<b>32</b>
2.1	Introduction . . . . .	33
2.2	Deep Learning in remote sensing . . . . .	33
2.3	Deep Learning Fundamentals . . . . .	34
2.3.1	Neural Network . . . . .	36
2.3.2	Deep Learning Architectures . . . . .	37
2.4	Deep Learning for Satellite Imagery . . . . .	41
2.4.1	Rationale for Using DL in Satellite Imagery Analysis . . . . .	41
2.4.2	Deep Learning Techniques and Models for Satellite Imagery Analysis . . . . .	42
2.5	Conclusion . . . . .	51
<b>3</b>	<b>Research Design and Methodology</b>	<b>52</b>
3.1	Introduction . . . . .	53
3.2	Global Architecture . . . . .	53
3.3	Detailed Architecture . . . . .	54
3.3.1	Dataset description and Preparation . . . . .	54
3.3.2	Data Splitting . . . . .	58
3.3.3	Model Construction . . . . .	58
3.3.4	Model Deployment . . . . .	67
3.4	Conclusion . . . . .	73
<b>4</b>	<b>Results and System Deployment</b>	<b>74</b>
4.1	Introduction . . . . .	75
4.2	Environment and Libraries . . . . .	75
4.2.1	Environment . . . . .	75

4.2.2	Programming Language:Python . . . . .	77
4.2.3	Key Libraries and Frameworks . . . . .	77
4.3	Results and Experiments . . . . .	78
4.3.1	Model 1: U-Net . . . . .	79
4.3.2	Model 2: SegFormer B2 . . . . .	81
4.3.3	Model 3: YOLO v8 . . . . .	83
4.4	Discussion . . . . .	88
4.5	Web Application . . . . .	92
4.6	Conclusion . . . . .	96
	<b>Conclusion and perspective</b>	<b>97</b>
	<b>Bibliography</b>	<b>99</b>

# List of Figures

1.1	Timeline of the Evolution of Remote Sensing Technologies (1839–Present)[5].	5
1.2	Earliest surviving aerial photograph taken by Nadar from a hot air balloon above Paris in 1866 [6]. . . . .	6
1.3	ERS-1 satellite in orbit [8]. . . . .	7
1.4	passive and active sensors [11]. . . . .	9
1.5	Drone, UAV, UAS, or RPAS [17]. . . . .	12
1.6	Example of ground-based sensing platforms include fixed sensors and mobile stations [18]. . . . .	12
1.7	Illustration of the MODIS (Moderate Resolution Imaging Spectroradiometer) sensor [19]. . . . .	13
1.8	Satellite-based remote sensing diagram [20] . . . . .	14
1.9	Wildfire Frequency in the United States, 1983–2022 [45]. . . . .	19
1.10	Global temperature records from 1880 to 2020 [47]. . . . .	21
1.11	Yearly Burned Area and Number of Fires—[2002-2023] [49]. . . . .	22
1.12	Avg. Burned Area (ha) / Region Area (km <sup>2</sup> ) and Avg. Nr of Fires / Region Area km-2023] [50]. . . . .	22
1.13	Tree cover loss due to fires in Algeria From 2001 to 2023 [52]. . . . .	24
1.14	Proportion of tree cover loss due to fires [2001-2023][52]. . . . .	24
1.15	Yearly Burned Area by Landcover (100%) - [2002-2023][49]. . . . .	25
2.1	Relationship between AI, Machine Learning, and Deep Learning[67]. . . . .	35
2.2	How deep learning works compared to traditional machine learning [69]. . . . .	36
2.3	Neural Network Architecture: From Image Input to Class Prediction [72]. . . . .	37
2.4	Feed Forward Neural Network[74]. . . . .	38

2.5	Recurrent Neural Networks (RNNs) [76]. . . . .	39
2.6	Transformers architecture [78]. . . . .	40
2.7	CNNs architecture[80]. . . . .	40
2.8	Example of Deep Learning based object detection for wildfire localization (Yolo V8) : The network identifies and marks the precise location of a fire. . . . .	43
2.9	YOLOv8 architecture [61]. . . . .	43
2.10	Example of Deep learning based segmentation for wildfire detection(U-Net) : The network processes a satellite image and generates a pixel-wise mask. . . . .	45
2.11	U-Net architecture[91]. . . . .	46
2.12	Architecture of Segformer [93]. . . . .	47
2.13	Example of deep-learning based image classification for wildfire detection[95]. . . . .	49
2.14	Architecture of EfficientNet[93]. . . . .	50
3.1	system workflow . . . . .	54
3.2	Active Fire Satellite Image vs Its Murphy Mask Label. . . . .	56
3.3	Comparison of RGB image and selected bands image . . . . .	57
3.4	Our Models . . . . .	59
3.5	U-Net architecture used for our segmentatuon task. . . . .	60
3.6	SegFormer-B2 architecture used for wildfire segmentation. . . . .	61
3.7	Use Case diagram . . . . .	68
3.8	Activity Diagram for Login Process . . . . .	69
3.9	Activity Diagram for Map analyse . . . . .	70
3.10	Activity Diagram for active fire Simulation . . . . .	71
3.11	Activity Diagram for Managing Stations . . . . .	72
3.12	Activity Diagram for viewing Alerts . . . . .	73
4.1	Training and Validation Accuracy and Loss Curves for U-Net . . . . .	80
4.2	Confusion Matrix for U-Net on the Test Dataset . . . . .	81
4.3	Training and Validation Accuracy and Loss Curves for SegFormer B2 . . . . .	82
4.4	Confusion Matrix for SegFormer B2 on the Test Dataset . . . . .	83
4.5	Training and Validation Loss Curves for YOLO v8 . . . . .	85

4.6	Recall, Precision, and F1-Score vs Confidence Threshold for YOLO v8 . . . . .	86
4.7	Recall, Precision, and F1-Score vs Confidence Threshold for YOLO v8 . . . . .	86
4.8	Recall, Precision, and F1-Score vs Confidence Threshold for YOLO v8 . . . . .	87
4.9	Confusion Matrix for YOLOv8 at the Bounding Box Level . . . . .	88
4.10	Comparative bar chart of the 3 model performance based on key metrics. . . .	89
4.11	Comparison of the detected masks from our models: U-Net, SegFormerB2, and YOLOv8. . . . .	89
4.12	login page . . . . .	92
4.13	dashboard page . . . . .	93
4.14	analyse map results . . . . .	93
4.15	alerts list page . . . . .	94
4.16	add station . . . . .	94
4.17	existant station . . . . .	95
4.18	Simulation page . . . . .	95
4.19	sub-images when no fire detected . . . . .	96
4.20	sub-images when fire is detected . . . . .	96

# List of Tables

1.1	15 Largest wildfires in world history (ranked by area burned) [46]	20
1.2	Total Burned Area by Wilaya[51]	23
1.3	Burned Area by Vegetation Type [51]	23
1.4	Related Works	31
3.1	Spectral bands of the Landsat 8 OLI and <i>TIRS</i> instruments[97]	57
3.2	Comparison of Model Training Configurations	65
3.3	Confusion Matrix	67
4.1	Performance Metrics of U-Net on the Test Dataset	79
4.2	Performance Metrics of SegFormer B2 on the Test Dataset	82
4.3	Performance Metrics of YOLO v8	84
4.4	Comparison of Key Performance Metrics Across Models	89
4.5	Comparison of Model Performance Metrics Between Our Models and Related Works	91



# List of Abbreviations

**AI** Artificial Intelligence

**BCE** Binary Cross-Entropy

**C2f** Cross-Stage Partial Fusion

**CCD** Charge-Coupled Device

**CLS** Classification loss

**CNNs** convolutional neural networks

**Concat** concatenation

**DFL** Distribution Focal Loss

**DFL Loss** Distribution Focal Loss

**DL** Deep Learning

**FCN** Fully Convolutional Network

**FNNs** Feedforward Neural Networks

**GEO** Geostationary Orbit

**GIS** geographic information systems

**GPU** Graphics Processing Unit

**IoU** Intersection over Union

**LEO** Low Earth Orbit

**LiDAR** Light Detection and Ranging

**lr** learning rate

**mAP** Mean Average Precision

**MBConv** Mobile Inverted Bottleneck Convolutions

**MODIS** Moderate Resolution Imaging Spectroradiometer

**NIR** Near-Infrared

**OLI** Operational Land Imager (for Landsat)

**OLI** Operational Land Imager

**ORM** Object Relational Mapper

**PIL** Python Imaging Library

**ReLU** Rectified Linear Unit

**RNNs** Recurrent Neural Networks

**SAR** Synthetic Aperture Radar

**SGD** stochastic gradient descent

**SPPF** spatial pyramid pooling module

**SSO** Sun-Synchronous Orbit

**SWIR** Shortwave Infrared

**Tanh** Hyperbolic Tangent

**TIRS** Thermal Infrared Sensor

**UI** User Interface

**VIIRS** Visible Infrared Imaging Radiometer Suite

**VRAM** Video Random Access Memory

**YOLO** You Only Look Once



# Introduction

Forests play an indispensable role in the health of our planet. They are vital in absorbing carbon dioxide and releasing oxygen, which helps regulate the global climate. In addition to their role in carbon sequestration, forests provide a habitat for wildlife, protect biodiversity, and regulate the water cycle. They also offer essential resources for humans, from timber to medicinal plants. The ecological, economic, and social significance of forests cannot be overstated. Their importance in maintaining ecological balance is profound, as they provide crucial services such as soil protection, climate regulation, and biodiversity preservation, all of which are vital for sustaining life on Earth.

However, despite these benefits, forests worldwide are increasingly under threat, most notably from a rise in wildfire activity. In recent years, the frequency and intensity of wildfires have grown significantly, driven by a combination of factors such as climate change, human activities, and poor environmental management. This concerning trend is especially evident in regions like northern Algeria, where recurrent fire outbreaks are fueled by dry climatic conditions, extreme heatwaves, and negligence, leading to the regular spread of fires across vast forested areas.

Given the rapid escalation of these fires, timely and efficient detection is crucial for mitigating their impact. Traditional methods of wildfire detection, such as ground-based observation or aerial surveys, have shown certain limitations due to their limited coverage, delayed responses, or inability to monitor large regions at once. Satellite remote sensing offers a powerful tool to meet this need. By using satellites, we can capture high-resolution images of vast areas, even those that are inaccessible, allowing for early detection of wildfires.

The true potential of this approach, however, is realized when combined with the latest advancements in deep learning. Deep learning, a branch of artificial intelligence, has proven to be highly effective in extracting meaningful insights from complex data such as satellite imagery. In our project, we leverage two key deep-learning techniques: segmentation and object detection. These tasks allow us to precisely delineate fire boundaries, track fire hotspots, and ensure no critical information is overlooked. This offers a more accurate and dynamic solution compared to classification methods. To accomplish this, we evaluate the performance of U-Net, SegFormer B2, and YOLO v8 models. These models are well-suited for the tasks of segmentation and object detection, and our goal is to assess their efficiency and accuracy in wildfire detection.

The success of this system, however, depends on the availability of high-quality datasets. The satellite imagery used must be accurate, providing the necessary data for training and fine-tuning deep learning models. In our project, we rely on carefully selected datasets that offer high-resolution images of wildfire-affected areas, ensuring that our models can detect fires with precision.

Finally, we will compare the performance of our models with those from related work to assess their effectiveness in wildfire detection and highlight key differences in accuracy and approach.

## Memo Organization and Structure

This memo comprises an introduction, four chapters, and a general conclusion, organized as follows:

- **Chapter 1:** Satellite Remote Sensing for Wildfire Detection ;this chapter covers the evolution and key principles of remote sensing technology, focusing on satellite sensors for wildfire detection. It highlights global and Algerian wildfire trends, emphasizing the urgency of the issue, and reviews traditional wildfire detection methods in Algeria, setting the stage for the application of advanced technologies in wildfire management.
- **Chapter 2:** Deep Learning Models for Satellite-Based Wildfire Detection ;this chapter explores the rise of deep learning in remote sensing, highlighting the use of various deep learning models like U-Net, SegFormer B2, and YOLO v8. It also discusses the specific tasks of segmentation and object detection in wildfire detection and why these approaches are essential.
- **Chapter 3:** Research Design and Methodology ;describes the proposed system architecture, including dataset description, model construction, preprocessing techniques, and performance metrics.
- **Chapter 4:** Results and System Deployment ;this chapter presents the results of the model evaluation, compares the performance of the U-Net, SegFormer B2, and YOLO v8 models. It also explains the deployment of the solution and any interfaces for the admin and user.
- **General Conclusion:** Summarizes the key findings from the project, evaluates the effectiveness of using deep learning for wildfire detection, and provides recommendations for future improvements and research in the field of wildfire monitoring using satellite imagery.

Chapter

I

# Satellite Remote Sensing for Wildfire Detection

## 1.1 Introduction

Wildfires profoundly disrupt the balance of natural ecosystems, inflicting extensive and persistent damage on forests and their surrounding environments. In many cases, full recovery can take years, sometimes even decades. As these events become more frequent and severe, it is crucial to develop and implement effective strategies that minimize their impact. We must prioritize reliable, sustainable solutions to manage this growing crisis, safeguarding both the environment and the communities that depend on it.

This chapter reviews the evolution and fundamental principles of remote sensing technology, focusing on satellite sensors used in wildfire detection. It examines global and Algerian wildfire trends to highlight the scale and urgency of the problem. Additionally, the chapter explores traditional wildfire detection techniques in Algeria, laying the groundwork for understanding how advanced technologies can enhance wildfire management and response.

## 1.2 Remote Sensing Technologies

Remote sensing is the science of obtaining information about the Earth's surface and atmosphere from a distance. This is accomplished by detecting and measuring electromagnetic energy that is reflected or emitted from objects or areas, typically through sensors mounted on satellites, aircraft, or other platforms. The core principle involves capturing electromagnetic radiation—ranging from visible light to infrared, thermal, and microwave wavelengths—and interpreting the variations in the energy to identify and analyze the characteristics of the observed targets. Remote sensing systems can be classified as passive, relying on natural sources of radiation such as the sun, or active, where the sensor emits its energy (e.g., radar) to illuminate the target[1][2].

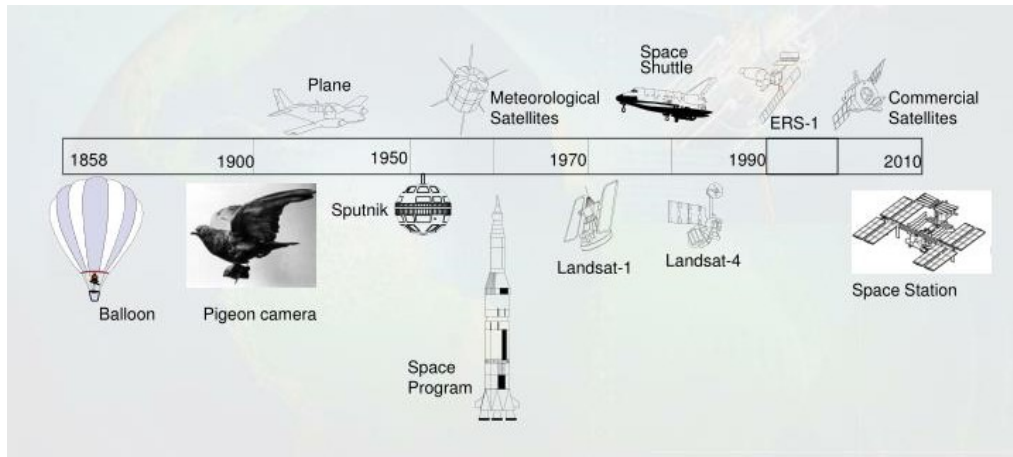
The significance of remote sensing lies in its ability to provide consistent, timely, and large-scale data that is indispensable for monitoring environmental changes, managing natural resources, and supporting decision-making across diverse sectors such as agriculture, forestry, urban planning, disaster management, and climate science. By enabling observation over vast and often inaccessible areas, remote sensing facilitates the detection of land cover changes, vegetation dynamics, water quality, and urban growth, among other applications. The integration of remote sensing data with Geographic Information Systems (GIS) and advanced image processing techniques further enhances the extraction of valuable information for sci-

entific research and practical applications.

Remote sensing is inherently multidisciplinary, combining optics, spectroscopy, photography, computer science, electronics, telecommunications, and space technology into a comprehensive system. The process involves several stages: emission of electromagnetic radiation (from the sun or transmission to the Earth's surface), interaction with surface materials (reflection, absorption, emission), transmission back to the sensor, and data output for analysis. The diversity of sensors and platforms, coupled with advances in digital processing and modeling, continues to expand the capabilities and applications of remote sensing technology [3].

### 1.2.1 Evolution of Remote Sensing

Understanding the evolution of remote sensing is essential for appreciating how this powerful tool has transformed into a cornerstone of modern Earth observation. From its origins in aerial photography to the advanced satellite technologies used today, the field reflects continuous innovation in sensors, data processing, and applications. These advancements have enabled broader coverage, faster acquisition, and higher precision, serving vital domains such as environmental monitoring, agriculture, urban planning, military intelligence, and disaster response [4].



**Figure 1.1** – Timeline of the Evolution of Remote Sensing Technologies (1839–Present)[5].

The history of remote sensing is broadly segmented into three historical periods:

- **The Early Age (1839–1907)**

This period marks the foundational phase of remote sensing, beginning with the invention of photography.

- In 1839, Louis Daguerre introduced the daguerreotype process, enabling visual documentation of the Earth's surface.
- In 1858, Gaspard-Félix Tournachon (Nadar) captured the first known aerial photograph from a tethered balloon over Paris.
- Innovations followed with Arthur Batut's kite photography in 1889 and Julius Neubronner's use of pigeons equipped with cameras in 1906.
- By 1909, aerial photography from airplanes emerged, enhancing image quality and acquisition range.



**Figure 1.2** – Earliest surviving aerial photograph taken by Nadar from a hot air balloon above Paris in 1866 [6].

### — The Medieval Age (1908–1945)

This era saw remote sensing adopted for strategic military purposes.

- Aerial reconnaissance became a critical component during World War I (1914–1918), improving battlefield mapping and surveillance.
- In the 1930s and 1940s, Germany pioneered the development and application of radar, extending sensing capabilities beyond the visible spectrum. Radar's extensive deployment during World War II laid the groundwork for post-war meteorological applications and long-range detection technologies [7].

### — The Modern Age (1946–present)

The launch of artificial satellites initiated the modern era of remote sensing.

- In 1957, the Soviet Union launched *Sputnik 1*, marking the beginning of space-based



Earth observation.

- In 1959, the *Explorer-6* satellite returned the first space-based image of Earth.
- With the launch of *TIROS-1* in 1960, consistent meteorological observation became possible.
- A major milestone occurred in 1972 with the launch of *Landsat-1*, equipped with multi-spectral scanners to monitor the Earth's surface.
- The 1980s and 1990s introduced radar satellites such as *ERS-1* and *Radar-sat*, as well as high-resolution commercial satellites like *IKONOS* and *QuickBird*.
- In recent decades, the integration of UAVs (drones), machine learning algorithms, and open-access satellite data (e.g., NASA and ESA) has revolutionized remote sensing applications, making them faster, more accurate, and accessible on a global scale [7].



**Figure 1.3** – ERS-1 satellite in orbit [8].

From early experiments with photography to cutting-edge multispectral satellite systems and AI-powered analytics, the historical trajectory of remote sensing illustrates the convergence of science, technology, and global needs for Earth observation.

### 1.2.2 Principles of Remote Sensing

Remote sensing refers to the acquisition of information about the Earth's surface and atmosphere from a distance, typically by detecting and measuring electromagnetic radiation reflected or emitted by objects. By eliminating the need for direct contact, remote sensing provides a powerful means of monitoring environmental processes at multiple scales. Understanding the underlying principles of remote sensing is essential for the correct interpretation and effective use of the data obtained. These principles govern how sensors operate and directly influence the quality, accuracy, and usability of information [9].

The core principles of remote sensing can be broadly categorized into three fundamental dimensions:

— **Spectral Resolution:**

Spectral resolution describes how well a sensor can differentiate between various wavelengths or spectral bands of electromagnetic energy. Sensors with high spectral resolution can detect very narrow ranges of wavelengths, which is essential for detailed analysis of land surfaces and environmental conditions. For instance, Landsat satellites are equipped with multispectral sensors that capture data across several broad bands, such as visible and near-infrared. Hyperspectral sensors go even further by recording data across hundreds of narrow, contiguous bands. This allows for more accurate identification of materials like different vegetation species, minerals, and water quality indicators. The choice of spectral bands is crucial: near-infrared bands are often used to assess plant health due to their sensitivity to chlorophyll, while specific visible and infrared wavelengths are employed to monitor water bodies [9] [10].

— **Temporal Resolution:**

Temporal resolution refers to how often a sensor revisits and captures data for the same geographic area. This factor is particularly important for observing changes that happen over time, such as crop growth, seasonal shifts, or natural disasters. Sensors with high temporal resolution can provide daily or even hourly data.

A notable example is the MODIS instrument aboard NASA's Terra and Aqua satellites, which offers daily global coverage—ideal for tracking phenomena like wildfire spread or flood development. On the other hand, satellites like Landsat typically revisit the same location every 16 days, which is sufficient for monitoring longer-term changes like deforestation or urban expansion. Depending on the study's goals, researchers must balance temporal resolution with spectral and spatial factors [9].

— **Spatial Resolution:**

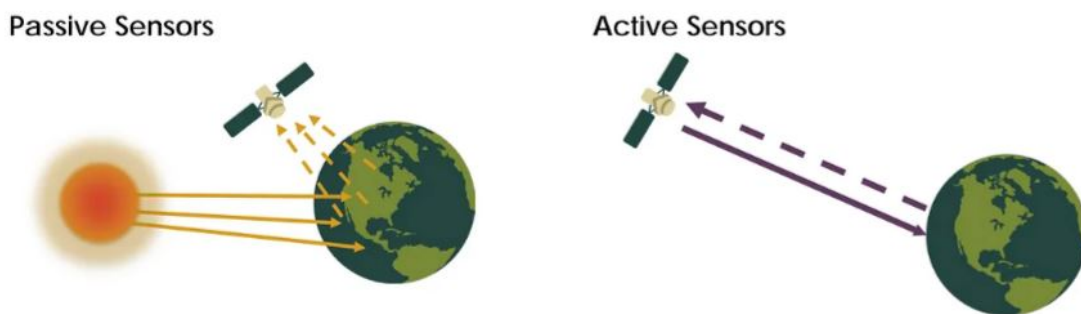
Spatial resolution defines the smallest feature that a sensor can detect on the Earth's surface. It is usually expressed in terms of pixel size; for example, a resolution of 10 meters means that each pixel represents a 10x10 meter area on the ground. High spatial

resolution is essential when detailed observation is required, such as in city planning or precision farming. Some commercial satellites, such as WorldView-3, provide imagery with resolutions as fine as 0.3 meters, making it possible to identify individual buildings and land plots. In contrast, sensors such as NOAA’s AVHRR have a coarser resolution of about 1 kilometer, making them more suitable for large-scale applications such as tracking sea surface temperature or vegetation patterns between regions. Ultimately, the spatial resolution dictates the level of detail available for analysis [9].

### 1.2.3 Types of Remote Sensing

Remote sensing systems are commonly classified in two main ways, based on how they work and where they are used. This classification reflects the fact that every system involves two key elements: the way it collects data, either by using natural energy (passive) or by emitting its signal (active), and the platform it operates from, whether it’s a satellite, an aircraft, a drone, or a ground-based setup. These two perspectives offer a structured understanding of the operational mechanisms and application domains of remote sensing technologies.

**A. Classification by Physical Principle** This classification is based on the physical principle behind how remote sensing systems collect data. It focuses on the type of energy involved in the sensing process. In this context, there are two main methods: passive remote sensing, which depends on natural sources of energy like sunlight, and active remote sensing, which uses artificial energy generated by the sensor itself. Each method has specific advantages depending on the application and environmental conditions.



**Figure 1.4** – passive and active sensors [11].

#### 1.Passive Remote Sensing

Systems rely on natural energy, mainly sunlight, to collect information reflected or emitted from the Earth’s surface. They operate across various spectral bands—like visible,

near-infrared, and thermal—and are used in satellites such as MODIS and Landsat for monitoring vegetation, land cover, and environmental changes; for example, MODIS, VIIRS, Sentinel-2, and the Landsat series [12].

**Advantages:**

- Large-scale coverage: Satellites can capture large areas, including polar regions, continents, and oceans; this is ideal for global monitoring.
- Cost-effective: for repetitious tasks or large areas, satellites are more economical, reducing the need for physical fieldwork.
- Long-term data archive: Some satellites offer historical data that enables analyses (deforestation, glacier retreat, etc.) [13].

**Limitations:**

- Lower spatial resolution: some satellites may not capture details, making them not suitable for analysis that requires high detail.
- High initial cost: the development and launching and maintaining of satellites are expensive and require significant investment.
- Atmospheric interference: Haze, cloud cover, aerosols, and atmospheric conditions can affect the quality of data and obscure data.
- Affected by cloud cover and dependent on daylight [14].

**2.Active Remote Sensing**

Systems emit their energy toward the target and measure the reflected or backscattered signal. These sensors operate independently of sunlight and can function effectively under cloudy or nighttime conditions. The most common active systems include radar and LiDAR technologies [12].

**Advantages:**

- All-weather and day/night capability: active sensors generate their energy, allowing data collection regardless of sunlight, cloud cover, or time of day.
- High-resolution 3D mapping: LiDAR systems can produce detailed three-dimensional models of terrain and structures, useful for topographic and infrastructure analysis.
- Vegetation and surface penetration: radar systems, such as SAR, can penetrate vegetation canopies or shallow surfaces, revealing hidden features.

- Independence from natural light: active systems operate independently of solar energy, enabling nighttime data acquisition.

**Limitations:**

- High cost and energy demand: Active remote sensing systems are more expensive to develop, operate, and maintain due to complex instruments and energy requirements.
- Complex data processing: Interpreting active sensor data often requires advanced technical knowledge and processing software.
- Signal interference: data quality may be affected by electronic interference or environmental noise.
- Limited spatial coverage (for non-satellite platforms): aircraft or drone-mounted systems typically cover smaller areas compared to satellites and may require multiple passes for wide coverage [15].

**B. Classification by Platform or Source** Remote sensing systems can also be classified based on the platform or source from which data is collected. This approach focuses on the physical location of the sensor—whether it is mounted on a satellite, an aircraft, or a drone, or positioned at ground level. Each platform comes with its own strengths and limitations in terms of spatial coverage, resolution, and operational flexibility, making them suitable for different types of environmental and geospatial applications. These platforms are often integrated with specific sensor technologies such as LiDAR and radar to enhance their capabilities.

**1. Aerial Remote Sensing**

Aerial platforms include crewed aircraft—such as fixed-wing planes, helicopters, and specially equipped survey planes—and Unmanned Aerial Vehicles (UAVs) or drones. UAVs are autonomous aircraft controlled via remote or preprogrammed flight paths and can carry a variety of sensors. They are especially valuable for applications requiring high-resolution imagery at a local scale, such as precision agriculture, disaster response, and infrastructure monitoring. Aircraft, on the contrary, are used for higher-altitude surveys that demand more advanced sensors [16].

**Advantages:**

- High spatial detail.

- flexibility in deployment.

**Limitations:**

- Limited spatial coverage per flight
- regulatory constraints.



**Figure 1.5** – Drone, UAV, UAS, or RPAS [17].

## 2. Ground-Based Remote Sensing

These systems involve sensors installed on or near the Earth’s surface, such as wireless sensor networks, thermal cameras, or fixed spectrometers. Ground-based systems are ideal for continuous, high-frequency measurements at fine spatial scales. They are often used for the validation of aerial and satellite data and for localized studies in agricultural, forest, and urban environments. Ground platforms may also incorporate radar and LiDAR technologies for enhanced detail.

**Advantages:**

- Real-time
- high-precision data.

**Limitations:**

- Small area coverage
- installation limitations.



**Figure 1.6** – Example of ground-based sensing platforms include fixed sensors and mobile stations [18].

### 3. Satellite-Based Remote Sensing

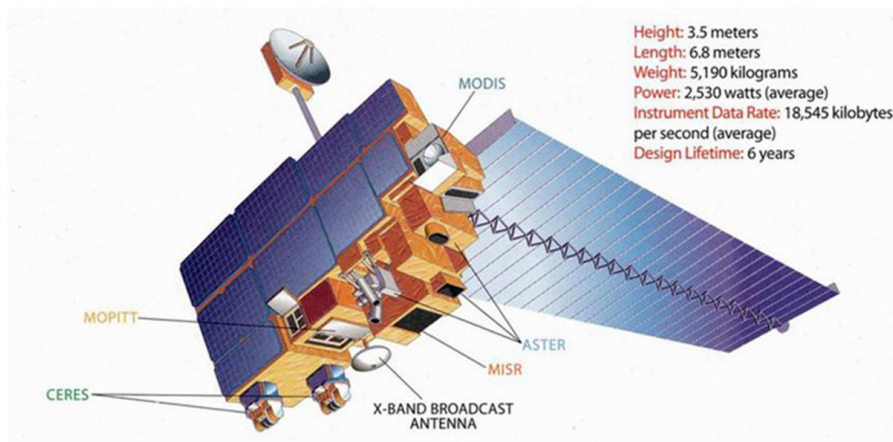
Satellite platforms host sensors that orbit the Earth and provide repetitive, large-scale, and long-term observations. They are essential for monitoring phenomena such as deforestation, urban expansion, climate change, and natural disasters. These systems enable consistent data collection across different time scales and geographic regions, making them highly valuable for temporal change detection and trend analysis. Well-known missions include Landsat, Sentinel-2, and MODIS. These platforms can carry both passive and active sensors, including radar and LiDAR.

#### Advantages:

- Broad coverage.
- Long-term monitoring.

#### Limitations:

- Moderate spatial resolution.
- Revisit time constraints.



**Figure 1.7** – Illustration of the MODIS (Moderate Resolution Imaging Spectroradiometer) sensor [19].

## 1.3 Satellite Imagery for Wildfire Detection

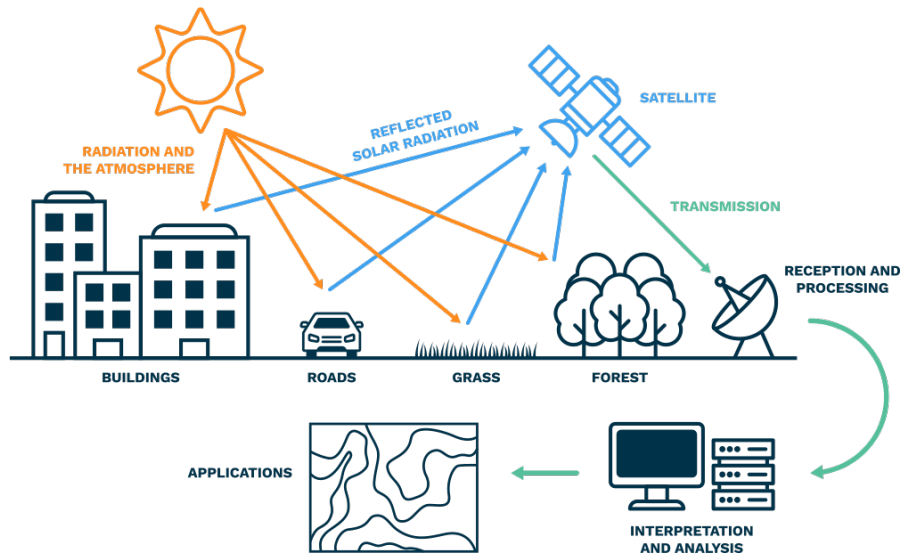
Wildfires increase daily due to many reasons, such as climate change and human activities, so early detection is an important step. In this section, we will discuss one of the most widely used techniques for fire detection, which is the use of satellite imagery.



### 1.3.1 Satellite Imagery Overview

Satellite imagery is images (visual data) collected and captured of plants, forests, Earth, etc. Equipped with satellites using onboard sensors and then transmitted to Earth for analysis, they are used to monitor phenomena.

Satellite imaging companies sell those images to governments and businesses [20]



**Figure 1.8** – Satellite-based remote sensing diagram [20] .

**A. The Capture of Earth’s Land Images by Satellites** Satellites capture images of the Earth using specialized sensors that detect electromagnetic radiation (light) reflected or emitted from the surface. The process involves several key components and steps:

- **Step 1: Satellite Orbit and Positioning:** A satellite is launched into space; it enters and is placed in a specific orbit around Earth, which is a balance between its forward velocity and Earth’s gravitational pull. This balance keeps the satellite moving around Earth without falling back or flying off into space. [21]

The type of orbit influences the frequency, coverage, and resolution of the images. Common orbits for Earth observation include:

- **Low Earth Orbit (LEO):** Satellites like Sentinel-2 or Landsat operate at altitudes of 600–800 km, completing an orbit every 90 minutes. LEO is ideal for high-resolution imaging due to its proximity to Earth.
- **Sun-Synchronous Orbit (SSO):** A type of LEO where satellites pass over the same spot on Earth at the same local time each day, ensuring consistent lighting for comparable images. Sentinel-2, for example, uses SSO for consistent monitoring.



- Geostationary Orbit (GEO): At 35,800 km, satellites like GOES remain fixed over one location, providing continuous coverage but lower resolution (1 km or coarser). These are less common for detailed land imaging [22].
- **Step 2: Image Capture Using Onboard Sensors:**
  - As the satellite moves along its orbit, its sensor scans Earth’s surface in swaths.
  - The sensor captures reflected light/radiation, capturing data in multiple spectral bands (e.g., red, green, blue, infrared). (Light reflected from Earth enters the sensor’s optics.)
  - Detectors (e.g., CCD arrays) convert light into digital signals.
  - Data is processed onboard or transmitted to ground stations when ensuring it is without significant errors [23].
- **Step 3: Data Processing and Image Creation:** Ground stations receive the satellite data and process it to create images.
  - Processing involves calibrating sensor data, correcting distortions caused by the satellite’s motion and Earth’s curvature, and stitching together multiple swaths to form continuous images or maps.
  - The result is detailed images of Earth’s land surface, which can be used for environmental monitoring, land use analysis, disaster response, and scientific research[24].

**B. Widely Used Satellite Sensors:** Satellite sensors are critical tools in Earth observation and remote sensing; these sensors fall into several main categories based on technology used and data type.

(a) **Optical Sensors**

They detect and capture reflected sunlight in visible, NIR, and SWIR. Examples of optical sensors:

- **Landsat Multispectral Scanner (MSS) and Thematic Mapper (TM)/Enhanced Thematic Mapper Plus (ETM+):** Operated by NASA/USGS, used for land use, agriculture, and forestry. [25]
- **MODIS (Moderate Resolution Imaging Spectroradiometer):** works on NASA’s Terra and Aqua satellites with 36 spectral bands and 250 m–1 km

resolution; it is used for global vegetation, fire detection, ocean color, smoke plumes, and burn scars in near-real time [26]

- **Sentinel-2:** Developed by the European Space Agency, with 13 spectral bands, it provides high-resolution multispectral imagery (10–60 m resolution) across visible, NIR, and SWIR regions, useful for agriculture, forestry, and land use studies also it is used for land monitoring and disaster management. [27].

(b) **Radar Sensors (Synthetic Aperture Radar—SAR)**

Emit microwave signals and measure their reflection, enabling them to penetrate clouds.

- **Sentinel-1 SAR:** ESA’s Copernicus program with C-band SAR and 5–40 m resolution transmits microwave signals and measures their return to monitor soil moisture and is used for flood mapping, maritime surveillance, and deformation monitoring. [28]
- **RADARSAT-2:** by the Canadian Space Agency with C-band SAR and 3–100 m resolution, it is used for ice monitoring and oil spill detection. [29]
- **ALOS-2 PALSAR-2** By Japan’s JAXA, with L-band SAR, 3–100 m resolution is used for forest monitoring and earthquake analysis. [30]

(c) **Thermal Infrared Sensors**

Measure emitted thermal radiation and detect long-wavelength infrared radiation; it is useful for temperature mapping, heat signatures, and volcanic activity.

- **Landsat Thermal Infrared Sensor (TIRS):** On Landsat 8/9 with 100 m resolution and 2 thermal bands, it is used for water resource management and urban heat islands. [31]
- **VIIRS (Visible Infrared Imaging Radiometer Suite):** Mounted on JPSS satellites, it captures 22 spectral bands with enhanced spatial resolution (350 m and 750 m at nadir), suitable for fire detection and radiative power estimation, nighttime light detection and thermal anomalies, and temperature. [32]

(d) **Microwave Radiometers and Altimeters**

Measure microwave emissions or altimetry to monitor ocean surface temperatures.

- **AMSR-E/AMSR2 (Advanced Microwave Scanning Radiometer):** On NASA’s Aqua and JAXA’s GCOM-W1, with 6–89 GHz frequencies and 5–50

km resolution for sea ice, soil moisture, and precipitation. [33]

- **GPM Microwave Imager (GMI):** By the Global Precipitation Measurement mission, with 10–183 GHz and 5–25 km resolution, it is used for rainfall and snowfall measurement. [34]
- **SMAP (Soil Moisture Active Passive):** For the NASA missions, with an L-band radiometer and 40 km resolution, it is used for soil moisture and drought monitoring. [35]

**C. Satellite Imagery Representation** The way to visualize the data captured by satellites depends on many factors, such as data format and spectral bands. These representations are significant in the monitoring and management of disasters.

- **Panchromatic (Grayscale):** uses a single broad-spectrum band to produce high-resolution (0.5–1 m/pixel) grayscale images, with some commercial satellites like WorldView-3 achieving resolutions of around 30 centimeters. This makes it ideal for extracting fine details such as building identification and mapping [36].
- **Multispectral:** captures data across multiple wavelengths (3–10) broad bands, including multiple colors of light, including visible light, near-infrared (NIR), and shortwave infrared (SWIR) bands [37].
- **Thermal Infrared:** measures heat signatures and surface temperatures emitted by objects on the Earth’s surface; it uses wavelengths from 8–15 m [38].
- **Hyperspectral:** captures data across hundreds of narrow, contiguous bands (e.g., 10–20 nm width), providing a detailed spectral analysis (mineralogy, pollutant detection) [39].
- **Synthetic Aperture Radar (SAR):** an active remote sensing technology using microwave signals.
  - Works regardless of daylight or weather conditions.
  - It captures information about surface characteristics and changes, such as ground deformation and deforestation[40].

## D. Applications of Satellite Imagery

1. **Environmental Monitoring** Satellites enable continuous, large-scale monitoring of environmental parameters such as air quality, land use, and water conditions. They provide diverse data types—including optical, thermal, and hyperspectral imagery—that facilitate timely detection and assessment of pollution and ecosystem changes, thereby supporting effective environmental management [41].

- **Agriculture and Food Security** Satellite imagery plays a crucial role in monitoring crop health, predicting yields, and managing agricultural resources to enhance food security. It allows rapid and reliable assessment of crop conditions over large areas, capturing spatial and temporal variability beyond the reach of traditional ground surveys. Satellite data supports early warning systems for agricultural anomalies and droughts, aiding timely decision-making to improve crop productivity and sustainability [42].

- **Oceanography and Maritime Surveillance** Satellite remote sensing is extensively utilized for ocean monitoring, including measuring sea surface temperature, analyzing ocean color for primary productivity, assessing water quality, and detecting marine pollution. It helps identify potential fishing zones by analyzing chlorophyll concentrations and temperature patterns indicative of fish abundance. Satellites also monitor coastal hazards like storm surges and provide frequent synoptic observations essential for understanding ocean dynamics and climate interactions. The integration of machine learning with satellite data is advancing oceanographic research and maritime surveillance [43].

2. **Urban Planning and Infrastructure Development** Satellite imagery combined with Geographic Information Systems (GIS) enhances urban planning by supplying up-to-date spatial data on land use, urban growth, and infrastructure development. This integration supports multi-scale urban planning and improves decision-making and management efficiency [44].

3. **Disaster Response and Recovery** Satellite imagery offers rapid situational awareness over large and often inaccessible areas, enabling efficient emergency response and resource allocation. It provides high-resolution spatial data critical for damage mapping, impact assessment, and monitoring reconstruction following

natural disasters such as floods, earthquakes, landslides, tsunamis, hurricanes, and wildfires [41].

- **Case Study:** Wildfires are indeed one of the natural disasters we have chosen as our case study. In the next subsection, we will delve deeper into wildfire-related topics, further exploring their impact and analysis.

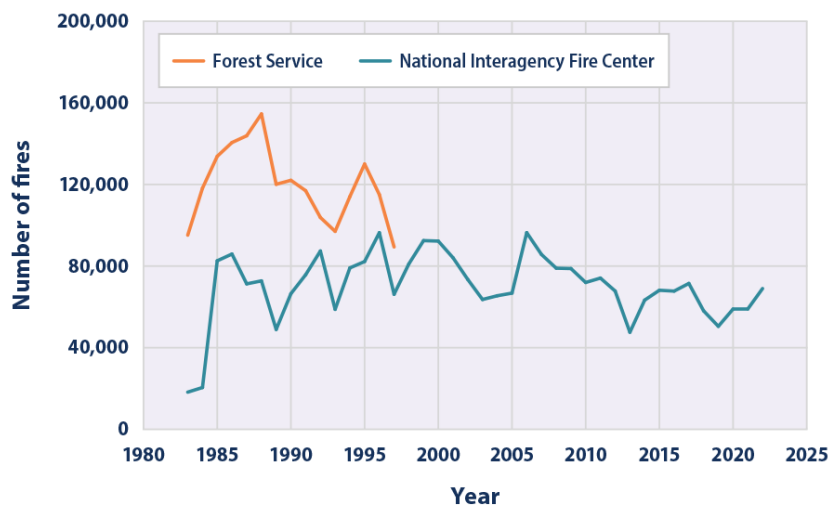
### 1.3.2 Wildfire Detection

Wildfires are a major environmental concern that requires careful monitoring. To effectively manage wildfires, it is important first to gain a clear understanding of their characteristics and behavior. In the following subsections, we will examine wildfire statistics to better comprehend these trends, as well as review traditional techniques used for fire detection. This will provide a foundation for advancing wildfire management strategies.

#### 1.3.2.1 Wildfire Statistics

##### A. Global Trends

Various factors, including human activity, lightning, and dry vegetation, cause wildfires. One major contributor is the rising global temperature since the Industrial Revolution, primarily driven by human activities such as greenhouse gas emissions. This increase in temperature intensifies evaporation, reduces soil moisture, and dries vegetation, making landscapes more flammable. Warming also worsens droughts and changes rainfall patterns, creating better conditions for fires to start.



**Figure 1.9** – Wildfire Frequency in the United States, 1983–2022 [45].

Figure 1.9 shows the total number of wildfires reported each year from 1983 to 2022, including even the smallest fires that cover just a few acres. It uses two different reporting systems, with the orange line showing data from the Forest Service. However, the Forest Service stopped collecting this data in 1997.

The data presented in the following table (Table 1.1) provides information on the geographical impact of wildfires globally, displaying the total area of land affected by each fire, ranked by burned area in millions of hectares. The table ranges from the New South Wales Bushfires (1974–75), which burned 117 million hectares, to the 2010 Russian Wildfires, which affected 0.3 million hectares.

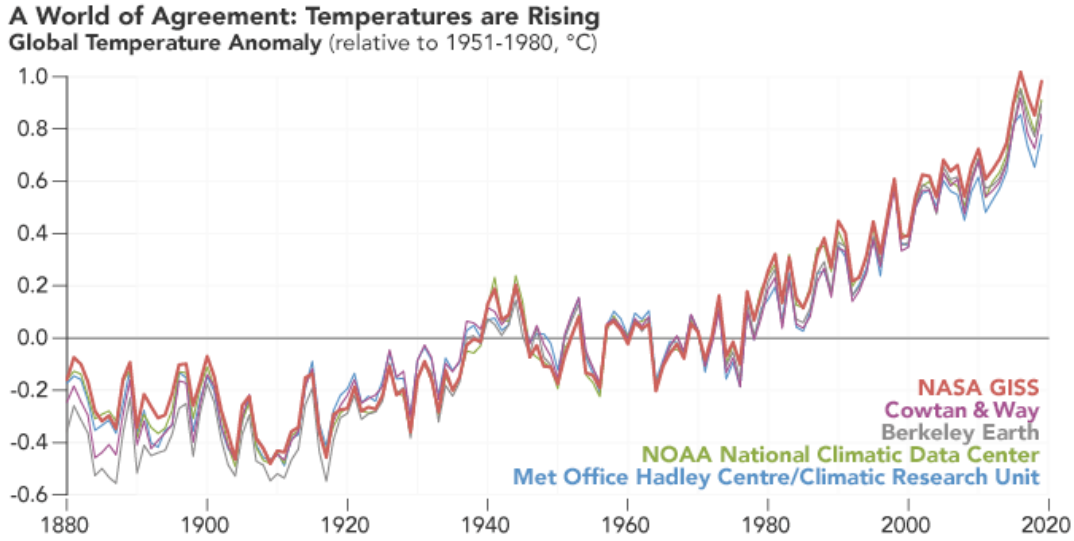
**Table 1.1** – 15 Largest wildfires in world history (ranked by area burned) [46]

Wildfire Event	Year(s)	Area Burned (ha)
New South Wales Bushfires	1974-1975	117,000,000
2019-2020 Australian Bushfires	2019-2020	24,300,000
Siberian Taiga Fires	2003	22,000,000
2021 Russian Wildfires	2021	18,000,000
Black Dragon Fire	1987	7,300,000
1989 Manitoba Fires	1989	3,200,000
Black Friday Bushfires	1939	2,000,000
Chinchaga Fire	1950	1,400,000 – 1,700,000
Great Fire of 1919	1919	2,000,000
The Big Burn	1910	1,200,000
Great Midwest Fires of 1871	1871	1,200,000
Taylor Complex Fire	2004	526,000
Smokehouse Creek Fire	2024	428,000
Yellowstone Fires	1988	321,000
2010 Russian Wildfires	2010	300,000

According to NASA’s analysis, the global average temperature has increased by approximately 1.1°C (1.9°F) since 1880, with the majority of this warming occurring since 1975 at a rate of 0.15 to 0.20°C per decade.

Global temperature records begin in 1880 due to limited data prior to this period. The following plot (Figure 1.10) illustrates temperature anomalies from 1880 to 2020, based on data from various sources, including NASA and the Met Office. Despite yearly fluctuations, all records show a clear trend of rapid warming, with

the last decade being the warmest [47].



**Figure 1.10** – Global temperature records from 1880 to 2020 [47].

This warming is not occurring uniformly—land areas are heating faster than oceans, and some regions, such as the Arctic, are experiencing much more rapid changes. These changes have made places that were once too cold or wet for wildfires more vulnerable to them. As a result, the geography of wildfire-prone areas is shifting, expanding the regions where fires can occur [47].

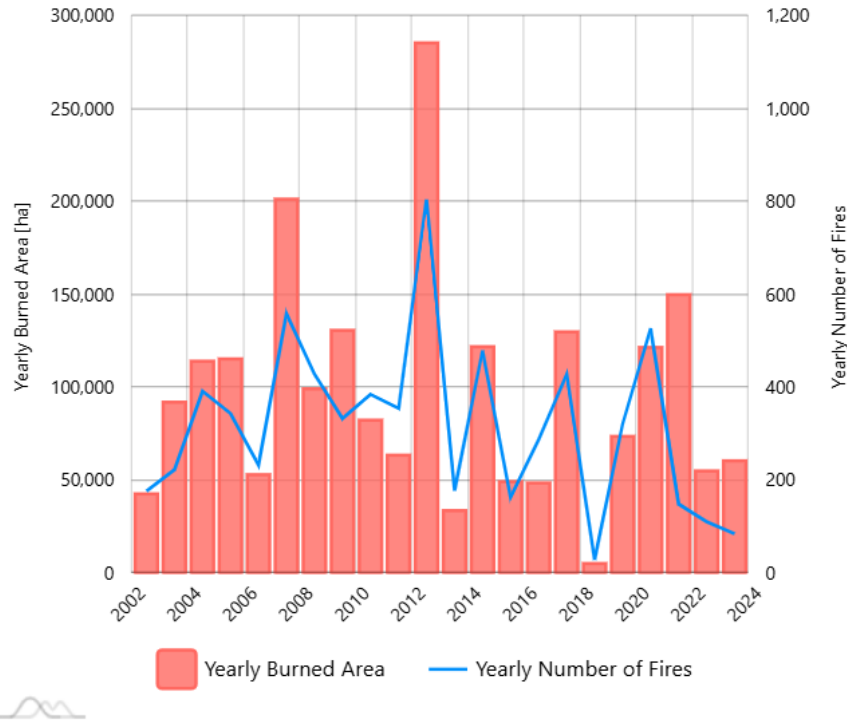
Since global warming affects regions differently, it is important to assess wildfire risks at the local level. In the next subsection, we will focus on Algeria as a case study and explore its wildfire statistics in detail.

## B. Wildfires in Algeria

Forest fires are a recurring phenomenon that has consistently impacted Algerian forests. Between 1985 and 2010, a total of 42,555 wildfires were recorded, affecting approximately 910,640 hectares of forest land. This equates to an annual average of 1,637 fires and 35,025 hectares burned, with an average of 21.39 hectares affected per fire [48].

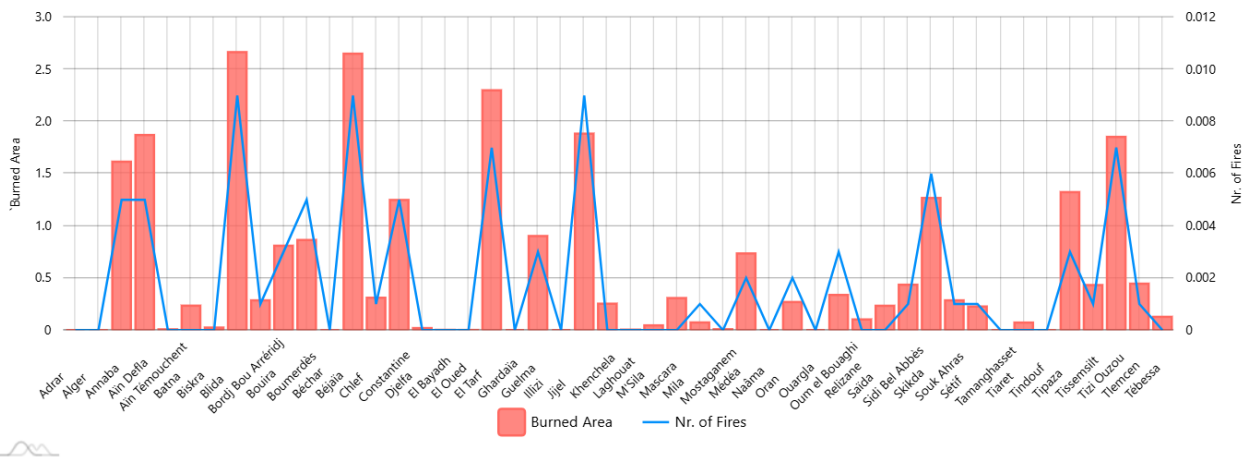
Transitioning to more recent years, Figure 1.11 tracks the yearly changes in the burned area and the number of fires in Algeria between 2002 and 2023, providing

more recent temporal trends.



**Figure 1.11** – Yearly Burned Area and Number of Fires—[2002-2023] [49].

To better understand the geographic distribution of wildfires, Figure 1.12 presents the average burned area and number of fires normalized by regional size between 2002 and 2023. This spatial analysis identifies regions with a higher density of fires and greater burned areas per square kilometer, which can inform targeted resource allocation for fire management and firefighting efforts.



**Figure 1.12** – Avg. Burned Area (ha) / Region Area (km<sup>2</sup>) and Avg. Nr of Fires / Region Area (km<sup>2</sup>)—[2002-2023] [50].



According to a public report published by the Algerian Press Service (APS) in 2022, summarizing the outcomes of a national forest protection commission meeting held at the Ministry of Agriculture and Rural Development, Algeria experienced devastating wildfires during the summer of 2021, affecting numerous wilayas and large areas of natural vegetation [51].

As highlighted in Table 1.3, forests were the most affected vegetation type, with an estimated 260,135 hectares destroyed, accounting for 26% of the total burned area. The Kabyle region, known for its dense forests, was especially impacted. According to Table 1.2, Tizi Ouzou and Bejaia suffered the most damage, with 43,398 and 13,174 hectares burned, respectively. Other wilayas, such as Khenchela, Guelma, El Tarf, and Annaba, also recorded substantial losses, each with over 5,000 hectares affected.

**Table 1.2** – Total Burned Area by Wilaya[51]

Wilaya (Province)	Burned Area (ha)
Tizi Ouzou	43,398
Bejaia	13,174
Khenchela	9,837
Guelma	5,927
El Tarf	5,090
Annaba	5,024

**Table 1.3** – Burned Area by Vegetation Type [51]

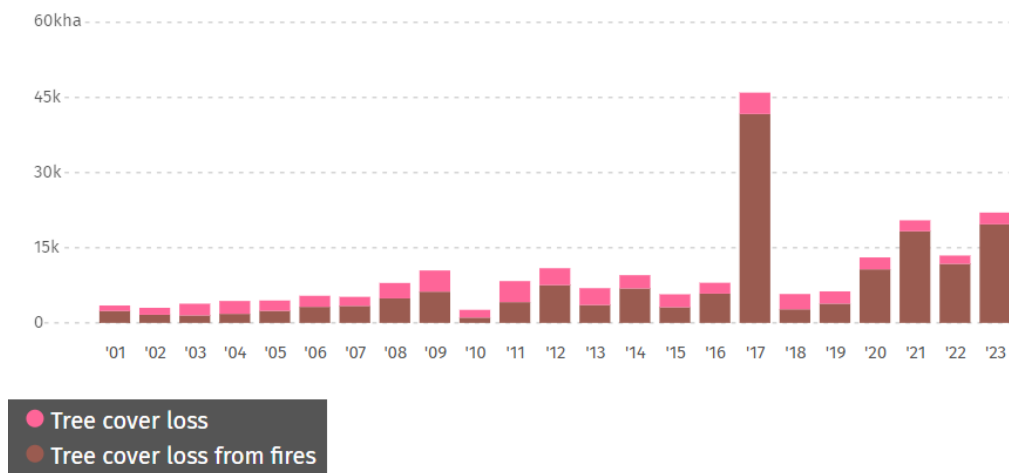
Vegetation Type	Burned Area (ha)	Percentage (%)
Forests	260,135	26
Shrubs	21,040	21.5
Bushland	16,415	16.5
Fruit Trees	16,160	36
Alfa Grass	352	0.5

- **Tree cover loss**

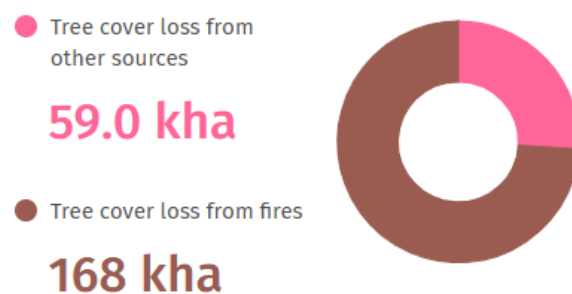
From 2001 to 2023, Algeria lost 168 kha of tree cover from fires and 59.0 kha from all other drivers of loss. The year with the most tree cover loss due to fires during

this period was 2017, with 41.7 kha lost to fires, 91% of all tree cover loss for that year (Figure 1.13).

Moreover, Figure 1.14 illustrates that fires were responsible for 74% of the total tree cover loss during this entire period, underscoring the dominant role of wildfires in deforestation.

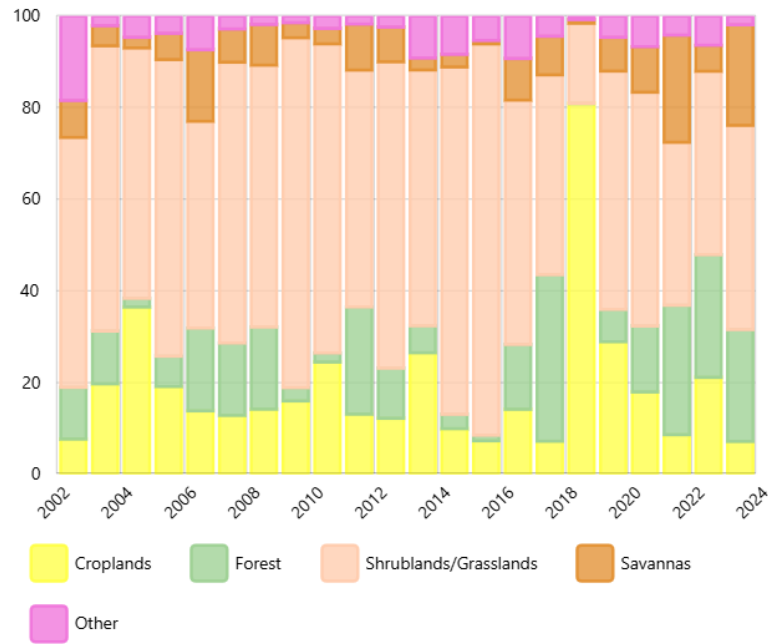


**Figure 1.13** – Tree cover loss due to fires in Algeria From 2001 to 2023 [52].



**Figure 1.14** – Proportion of tree cover loss due to fires [2001-2023][52].

Figure 1.15 shows the percentage of the total burned area each year distributed by different land cover types, highlighting how much of the fire-affected area is forest compared to other land covers from 2002 to 2023.



**Figure 1.15** – Yearly Burned Area by Landcover (100%) - [2002-2023][49].

These statistics highlight the urgent need for robust wildfire monitoring systems. Effective monitoring and proactive management are essential to minimize the devastating impact of wildfires on Algeria’s forests and natural environment, ultimately reducing their frequency and severity.

### 1.3.2.2 Traditional Wildfire Detection Techniques in Algeria

**Community Reporting and Volunteer Networks** Local populations, including villagers and forest visitors, frequently report fires as soon as they notice smoke or flames. This grassroots vigilance is critical in the initial detection and rapid communication to authorities or firefighting units. The Algerian Red Crescent (ARC), which has a widespread network of local units covering all wilayas and access to 24,000 volunteers, relies heavily on community engagement for initial fire assessments and interventions [53].

**Ground Patrols and Surveillance Towers** Algeria has traditionally depended on ground patrols and manned surveillance towers to detect forest fires. Observers stationed in lookout towers scan the horizon for smoke or fire signs and monitor weather conditions that could increase fire risk. However, these methods are often inefficient due to limited visibility caused by weather, terrain obstacles, and the

reliance on human vigilance, which can delay detection and reporting [54].

**Aerial Surveillance** Another traditional technique involves aerial patrols, where pilots and observers fly over forested and remote areas during high-risk periods, such as dry seasons or after storms. This method allows for broader coverage and quicker reporting compared to ground patrols. However, aerial surveillance can be hindered by poor visibility due to smoke, weather, or night conditions, limiting its effectiveness [54].

- **Challenges:**

- **Delayed Detection:** Human observation is slower and less reliable, especially in vast or inaccessible forest areas.
- **Limited Coverage:** Towers and patrols cover limited areas and cannot provide continuous monitoring.

All these traditional methods, which rely heavily on human observation, face significant limitations that have prompted the adoption of more advanced technological solutions. In the next section, we will explore the importance of one such advanced technique (satellite imagery) for wildfire detection and monitoring.

### 1.3.3 The role of satellite imagery in wildfire detection

The satellite imagery plays an important role in multiple stages of wildfire.

Monitoring the forest fire:

- **Pre-fire:** facilitate the detection and monitoring of regions susceptible to wildfire hazards by detecting high temperature and signs of moisture stress that are considered early indicators of potential wildfire, also pinpointing the vulnerable populations
- **During active fire:** satellites enable real-time tracking of the wildfire by determining the spread direction and estimating the space covered by fire and the burned area. This critical information helps combat advancing flames.
- **Post-fire:** helps to detect burned vegetation and soil, enabling detailed mapping of fire severity and tracking vegetation recovery [55].

### 1.3.4 Active fire satellite image Acquisition

It is essential to outline the methods for acquiring satellite imagery relevant to wildfire detection, including the platforms utilized, manual retrieval techniques, and the evaluation of publicly available pre-existing datasets.

**A. Satellite Imagery Sources** Before discussing datasets, it is essential to know the platforms that provide satellite images and active fire hotspot data.

#### 1. Platforms that Provide Active Fire Hotspot Coordinates

Several authoritative sources offer geolocated fire hotspot data detected via satellite sensors:

- **NASA FIRMS (Fire Information for Resource Management System):** Provides fire hotspot coordinates detected by MODIS and VIIRS sensors, with data accessible via CSV, shapefiles, and APIs.
- **NOAA VIIRS Active Fire Data:** Fire hotspot data available through NOAA portals, derived from VIIRS sensors.
- **European Space Agency (ESA) Sentinel Hotspots:** Fire detection products derived from Sentinel satellites (Sentinel-2, Sentinel-3).
- **Global Fire Emissions Database (GFED):** Combines multiple satellite products to compile global fire hotspot data.
- **Copernicus Emergency Management Service (EMS):** Offers rapid mapping and fire hotspot detection based on Sentinel satellite data.

#### 2. Platforms Providing Satellite Imagery

- **Sentinel Hub**<sup>1</sup> Provides multispectral satellite imagery from Sentinel-1, Sentinel-2, Landsat, and others with customizable APIs and interfaces.
- **USGS Earth Explorer**<sup>2</sup> Grants access to Landsat, MODIS, and other satellite imagery collections.
- **NASA Worldview**<sup>3</sup> An interactive tool to visualize near real-time imagery from MODIS, VIIRS, and other satellites.

---

1. <https://sentinel-hub.com>

2. <https://earthexplorer.usgs.gov>

3. <https://worldview.earthdata.nasa.gov>

- **Google Earth Engine**<sup>4</sup> A cloud-based platform with extensive satellite datasets, including Landsat, Sentinel, and MODIS, supporting large-scale data analysis.
- **Copernicus Open Access Hub**<sup>5</sup>: Provides Sentinel satellite data searchable by location and date.
- **Amazon Web Services (AWS) Open Data Registry**<sup>6</sup> Hosts openly accessible Landsat and Sentinel imagery on cloud infrastructure.

**B. Manual Fire Satellite Image Retrieval** It involves downloading precise geographic coordinates of active fire locations from trusted sources such as NASA FIRMS. These coordinates are then used to obtain corresponding satellite images, for example, via Sentinel Hub.

• **Limitations of this approach :**

- The process can be time-consuming due to manual querying and downloading.
- Image quality and resolution may vary depending on satellite revisit intervals and sensor specifications.

**C. Pre-existing Datasets** Several publicly available datasets were evaluated for this study, including:

- **Active Fire Detection in Landsat-8 Imagery** The publicly available large-scale dataset for active fire recognition consists of 8,194 Landsat-8 images collected globally during August 2020. The dataset includes significant wildfire events across diverse regions such as the Amazon, Africa, Australia, and the United States. It comprises a total of 146,214 image patches, approximately 192 GB in size, featuring 10-band multispectral images in TIFF format. Additionally, the dataset provides outputs generated by three well-established handcrafted active fire detection algorithms (Schroeder et al. [2016], Murphy et al. [2016], and Kumar and Roy [2018]), which serve as valuable references for model evaluation. [56]
- **Wildfire Prediction Dataset (Satellite Images)** This dataset comprises

---

4. <https://earthengine.google.com>

5. <https://dataspace.copernicus.eu/>

6. <https://registry.opendata.aws/>

satellite images of regions in Canada that have previously experienced wildfires, sourced from the Government of Canada's Forest Fires Open Data Portal under the Creative Commons 4.0 Attribution (CC-BY) license for Quebec. The dataset includes 42,850 images of size 350x350 pixels, categorized into two classes: 22,710 images depicting wildfire-affected areas and 20,140 images without wildfire presence. Using longitude and latitude coordinates corresponding to wildfire spots with burned areas exceeding 0.01 acres, satellite images were extracted via the MapBox API. This preprocessing step transformed the raw wildfire location data into a structured image dataset, optimized for deep learning applications aimed at predicting wildfire risk. [57]

- **Datasets on Roboflow**<sup>7</sup> Roboflow provides a diverse collection of datasets relevant to active fire detection, which includes satellite imagery and annotated data. Among these, some datasets consist of raw satellite images capturing active fire events without annotations, serving as valuable input data for training models. Other datasets include labeled images where fire-affected regions are annotated to provide ground truth masks for supervised learning.

### 1.3.5 Limitations and Challenges of Satellite-Based Fire Monitoring

Despite their capabilities, satellite systems for fire detection have several limitations:

- Cloud cover and smoke can obscure optical sensors, limiting the accuracy of fire detection in thick atmospheric conditions [58].
- Spatial resolution limitations mean that small fires or those beneath forest canopies may go undetected, particularly by low-resolution sensors like MODIS.
- Revisit frequency can delay detection or miss short-lived fire events, especially in systems with lower temporal resolution like Landsat.

To overcome these limitations, multiple satellite datasets are often integrated, and ground validation is used where possible.

---

7. <https://roboflow.com/>

## 1.4 Related Works

The detection of wildfires has become an essential focus in remote sensing and machine learning applications. Several studies have explored different methodologies and models for effectively identifying fire-affected areas. This section reviews relevant works that employed various machine learning techniques, highlighting their approaches, datasets, and results. By examining these studies, we aim to contextualize our approach and identify the strengths and limitations of existing methods in wildfire detection.

- Boumaiza and Brahimi (2024) [59] used the **U-Net** architecture with **Landsat-8** imagery to detect wildfires. They evaluated performance on three types of fire masks: *Intersection*, *Voting*, and *Murphy*. Out of the configurations tested, the Murphy mask-based model performed best, achieving **95% precision**, **74% recall**, and an **F1-score of 83%**, using a dataset of **45,376 satellite images with their labeled masks**.
- James et al. (2023) [60] implemented **MobileNetV2**, a lightweight CNN, via transfer learning for binary wildfire classification ("*Fire*" vs. "*No Fire*") using **Sentinel-2** imagery from *San Isabel National Park*. Trained on a small dataset of **614 images**, the model achieved **95% accuracy** with a **2% error rate**, showing that efficient CNNs can still deliver strong performance on wildfire detection even with limited data.
- Choutri et al [61] (2023) proposed a UAV-based wildfire detection and geo-localization system using **YOLO-based models**, with a specific focus on evaluating multiple versions before selecting **YOLO-NAS** as the most effective. Their system utilized a two-stage pipeline: offline detection using annotated aerial images and online real-time detection with geo-localization, implemented through stereo vision on a UAV platform.

The dataset comprised over **12,000 labeled aerial images**, collected from diverse sources such as *BowFire*, *FiSmo*, *FLAME*, and additional UAV-acquired imagery. Images were annotated into three distinct classes: *fire*, *non-fire*, and *smoke*, which significantly helped in minimizing false detections—a crucial factor for practical deployment.



In terms of performance, the **YOLO-NAS** model achieved a **mean Average Precision (mAP@50)** of approximately **0.71** and an **F1-score** of **0.68**. For the real-time detection and localization component, the system was successfully integrated into a UAV equipped with stereo cameras and a Pixhawk flight controller, enabling 3D geo-localization of wildfire occurrences during flight.

A comparative summary of the reviewed studies is presented in Table 1.4

Study	Architecture Type	Model Name	Approach Type	Dataset Type	Dataset size
Boumaiza & Brahimi (2024)	CNN	U-Net	Segmentation	Landsat-8 satellite images with labeled masks	45,376
James et al. (2023)	CNN	MobileNetV2	Binary Classification	Sentinel-2 (San Isabel National Park)	614
Choutri et al. (2023)	CNN	YOLO-NAS	Object Detection	BowFire, FiSmo, FLAME, and UAV aerial images (annotated)	12,000+

**Table 1.4** – Related Works

## 1.5 Conclusion

In this chapter, we reviewed the key aspects of satellite remote sensing and its application in wildfire detection. We also explored historical wildfire trends and traditional detection methods, providing a foundational understanding of how technological advancements support modern wildfire management.

The next chapter will delve into advanced computational techniques in Earth observation, with a focus on deep learning and its expanding role in satellite data analysis. We will highlight how these innovative approaches are being applied to improve environmental monitoring capabilities.

Chapter

II

# Deep Learning Models for Satellite-Based Wildfire Detection

## 2.1 Introduction

The integration of deep learning with remote sensing technology has revolutionized the detection and analysis of wildfires from satellite imagery. This chapter provides an in-depth overview of the foundational concepts necessary to understand these advances. We begin by discussing the evolution of deep learning within remote sensing, highlighting its growing importance and key applications in environmental monitoring.

The chapter then delves into the fundamental principles of deep learning, including neural network structures and the main architectural families employed in image analysis. Special emphasis is placed on the diverse types of deep learning architectures and their relevance to satellite-based image interpretation.

Finally, we examine how these advanced computational methods are specifically applied to satellite imagery for wildfire detection. By outlining representative techniques, object detection, segmentation, and classification, the chapter builds a comprehensive understanding of how deep learning improves the capabilities of satellite systems to monitor and manage wildfires.

## 2.2 Deep Learning in remote sensing

Deep learning has brought a major shift in how satellite and aerial images are analyzed within remote sensing. Although remote sensing itself has been around since the 19th century, starting with aerial photography and later evolving with satellite imagery in the mid-20th century, it wasn't until the 2010s that deep learning began to play a major role in this field. Although early machine learning methods started appearing in geographic information systems (GIS) around 2009, deep learning only began to truly reshape remote sensing analysis with the arrival of powerful convolutional neural networks (CNNs) and large labeled datasets like ImageNet [62]. By 2020, pre-trained deep learning models had become readily available, making it easier to adopt these techniques for tasks such as land cover classification, object detection, and environmental monitoring [63].

- **The rise of deep learning in Remote Sensing**

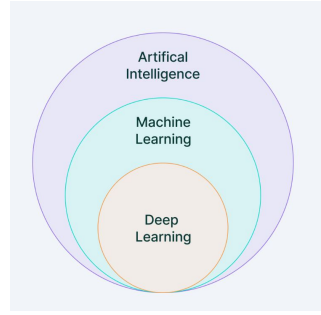
Deep learning has become critically important in remote sensing due to its ability to automatically learn and extract complex features from vast amounts of remote sensing data. Unlike traditional methods that rely heavily on handcrafted features and expert knowledge, deep learning models can capture mid- and high-level abstract features directly from raw images. This capability significantly improves tasks such as image registration, classification, object detection, and semantic segmentation in remote sensing, enabling more accurate and efficient analysis of satellite and aerial imagery[64].

Compared to traditional remote sensing methods, deep learning offers several advantages. Traditional approaches often depend on manual feature engineering, which can be time-consuming and less adaptable to diverse data types. Deep learning, being fully data-driven, can automatically learn discriminative features from large-scale datasets, improving performance in complex scenarios such as hyperspectral image classification and change detection. In addition, deep learning models can integrate information from multiple data sources and modalities, such as combining optical and SAR images, which is challenging for conventional methods. This leads to better generalization, higher accuracy, and the ability to handle large data effectively [65].

The rise of deep learning in remote sensing has been rapid and transformative. Since around 2014, there has been an exponential increase in research and applications that leverage deep learning techniques in this field. This surge is driven by advances in computational power, the availability of large annotated datasets, and the success of deep learning in adjacent fields like computer vision. Remote sensing scientists are increasingly adopting deep learning to tackle unique challenges such as geospatial data fusion, 3D reconstruction, and environmental monitoring. The trend is expected to continue as new satellite missions provide richer data and as deep learning architectures evolve to better suit the spatial and temporal characteristics of remote sensing data.

## 2.3 Deep Learning Fundamentals

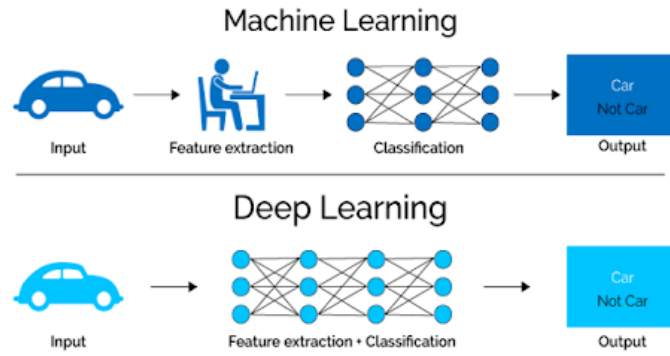
Deep learning is a specialized branch of machine learning that uses algorithms known as artificial neural networks to process data in complex ways [66], which itself is a subset of artificial intelligence, as illustrated in Figure 2.1.



**Figure 2.1** – Relationship between AI, Machine Learning, and Deep Learning[67].

Deep learning has become famous for its ability to handle complex tasks like recognizing images or understanding speech. It differs from traditional rule-based systems by using what are called artificial neural networks—computational systems made up of layers of connected “neurons” that process information, a bit like how the human brain works [68]. Neural networks are the foundational building blocks of deep learning; they enable computers to learn from data by recognizing patterns and making decisions automatically.

What makes deep learning “deep” is the use of many layers, not just one or two. The information passes through these layers, and at each stage, the system learns more abstract details from the data. During training, deep learning models figure out how to get better by adjusting their internal settings in response to mistakes—essentially learning from experience. A big difference between deep learning and traditional machine learning is that deep learning can automatically find important patterns in huge amounts of raw data without needing people to tell it exactly what to look for. That’s why deep learning is so effective in areas like image recognition, speech understanding, and language translation, where data is messy and complicated [67].



**Figure 2.2** – How deep learning works compared to traditional machine learning [69].

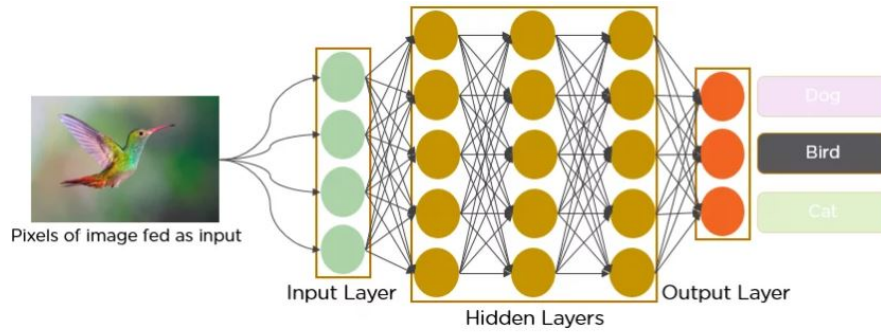
Neural networks are the foundational building blocks of deep learning; they enable computers to automatically learn from data by recognizing patterns and making decisions.

### 2.3.1 Neural Network

A neural network is a core concept in artificial intelligence, inspired by the way the human brain processes information. These models are designed to learn from data by recognizing patterns, making predictions, and solving problems across a variety of tasks. At their core, neural networks consist of many simple computing units called artificial neurons, which are connected to one another and communicate by passing signals [70].

The structure of a neural network is organized into layers. There is an input layer, which receives the initial data; one or more hidden layers, where complex transformations and computations take place; and an output layer, which produces the final prediction or result. Each neuron in these layers performs a simple calculation, and the network as a whole adjusts its connections based on the data it sees during training. This layered design enables neural networks to capture intricate relationships within data and to learn automatically from experience, making them fundamental to modern deep learning approaches. [71]

The structure of a neural network consists of several interconnected layers of artificial neurons (also called nodes or units). Each layer serves a different role in processing information:



**Figure 2.3** – Neural Network Architecture: From Image Input to Class Prediction [72].

### 2.3.2 Deep Learning Architectures

Deep learning architectures refer to the various ways neural networks are structured to learn complex patterns from data. While the structure involves fundamental elements—such as layers, neurons, and their interconnections—the architecture defines how these components are organized and applied to specific tasks. Selecting the appropriate architecture is crucial, as it directly impacts a model’s ability to handle different types of data and problem domains, such as image recognition, natural language understanding, or time-series forecasting [73].

The diversity of deep learning architectures has emerged to meet the unique demands of various applications. Each architecture offers specialized capabilities tailored to particular challenges and data characteristics. This variety is especially relevant in fields like wildfire detection from satellite imagery, where precise pattern recognition is essential.

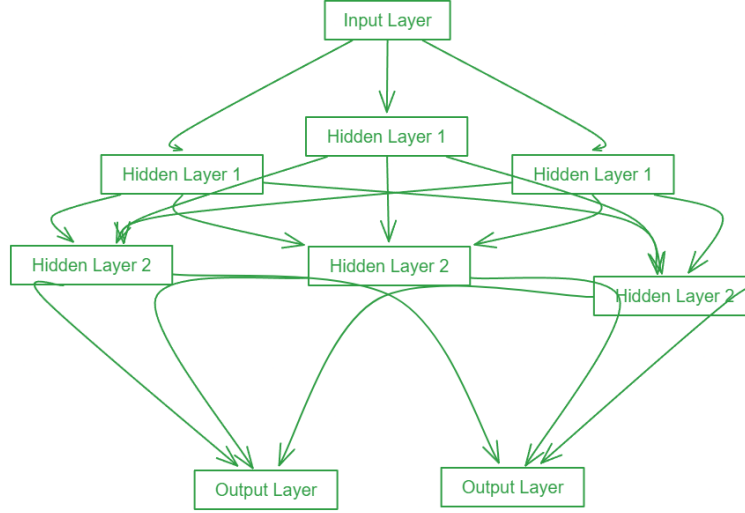
In the following section, we provide a brief overview of major deep learning architectures and highlight what distinguishes each, particularly in relation to solving complex problems such as wildfire detection.

#### A. Feedforward Neural Networks (FNNs)

Feedforward Neural Networks (FNNs), also known as multilayer neural networks, are artificial neural networks where information flows in one direction—from the input layer, through hidden layers, to the output layer, with no cycles or loops. Data is processed step by step as it moves through each layer, with each neuron applying weights, biases, and activation functions to transform the input and produce a final output [74].

— **Types of Data Suited for FNNs:**

- Tabular or structured data (e.g., spreadsheets, databases).
- Simple classification or regression tasks.
- Data where feature order or spatial structure is not important.



**Figure 2.4** – Feed Forward Neural Network[74].

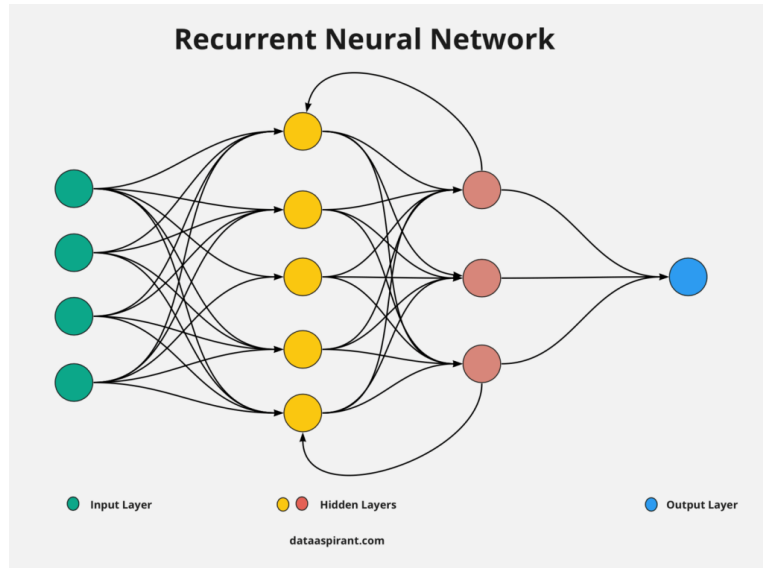
## B. Recurrent Neural Networks (RNNs)

Recurrent Neural Networks (RNNs) are specialized neural networks designed to process sequential data by maintaining connections that form cycles. Unlike feed-forward networks, RNNs allow information to persist and be reused across steps in a sequence, enabling the network to learn patterns that depend on order or context. Each neuron in the hidden layers receives input not only from the current data point but also from its own previous output, making RNNs well-suited for data where timing and sequence matter [75].

— **Types of Data Suited for RNNs:**

- Any data where the order of information is important.
- Sequential or time-series data (e.g., speech, audio, sensor readings).
- Text and language (e.g., translation, sentiment analysis).





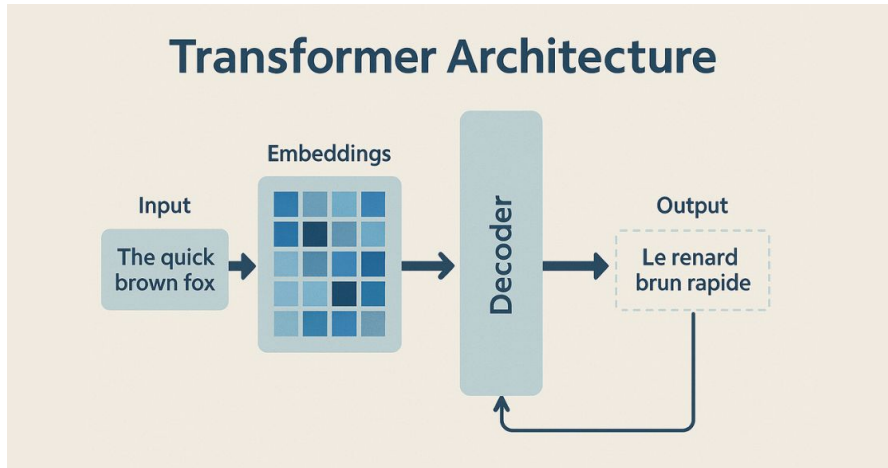
**Figure 2.5** – Recurrent Neural Networks (RNNs) [76].

### C. Transformers

Transformers are a modern deep learning architecture designed to handle sequential data by using a mechanism called attention. Unlike RNNs, Transformers process entire sequences at once, allowing the model to capture relationships between any two elements in the sequence, regardless of their position. This architecture excels at learning complex dependencies and handling large datasets efficiently [77].

#### — Types of Data Suited for Transformers:

- Natural language data (text, translation, question-answering).
- Time-series and sequential data
- Images (when adapted as Vision Transformers)
- Any data where understanding relationships between elements is important



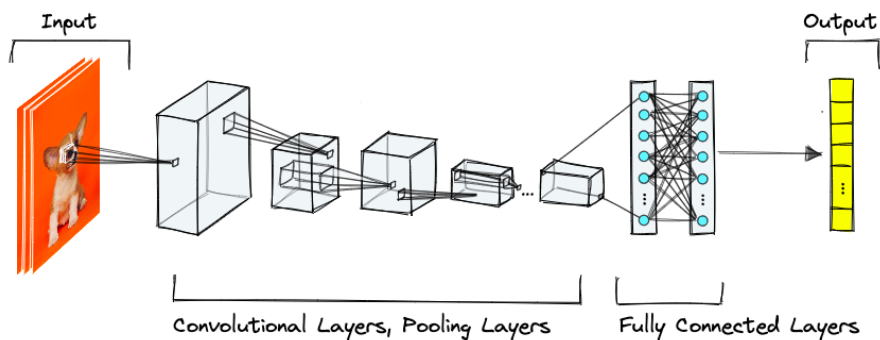
**Figure 2.6** – Transformers architecture [78].

#### D. Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) are deep learning architectures designed to automatically recognize patterns and features in grid-like data, such as images. CNNs use convolutional layers to scan input data with filters, making them especially effective for visual tasks [79].

##### — Types of Data Suited for CNNs

- Image data (satellite, medical, photographs).
- Spatial data where patterns and position matter.
- Relevance to Wildfire Detection: CNNs are at the core of wildfire detection from satellite imagery, as they can accurately identify fire patterns, smoke, and other related features in large volumes of image data.



**Figure 2.7** – CNNs architecture[80].

## 2.4 Deep Learning for Satellite Imagery

Deep learning has become an essential tool for analyzing satellite imagery by enabling the automated extraction of complex patterns and features from large and diverse datasets. These methods use artificial neural networks that learn directly from the images, eliminating the need for manual feature engineering. This approach makes it possible to process and interpret data from various sensors, like optical, infrared, or radar, captured by satellites, and to identify intricate structures and subtle changes that might be overlooked with traditional analysis.

The strength of deep learning lies in its ability to integrate multiple layers of information, such as different spectral bands and spatial resolutions, into a single analytical framework. This allows for a more nuanced and comprehensive understanding of the content of satellite images. Because satellite data are often high-dimensional and complex, deep learning's flexibility and adaptability are crucial for extracting meaningful information in a reliable and efficient manner [81].

### 2.4.1 Rationale for Using DL in Satellite Imagery Analysis

Deep learning has proven to be highly effective in satellite imagery analysis. Its ability to automatically learn and extract meaningful features from complex visual data makes it ideal for this task, as demonstrated by the following key points:

- **Hierarchical Feature Learning:**

CNNs extract low-level to high-level features (edges → smoke, fire shapes) from satellite images, essential for wildfire detection [82].

- **Spatial Invariance:**

They're robust to changes in wildfire position and orientation, crucial in satellite images with shifting perspectives [83].

- **End-to-End Learning:**

CNNs learn directly from raw data, with no need for manual feature engineering, saving time and improving flexibility [83].

- **Transfer Learning:**

Pre-trained CNNs can be fine-tuned for wildfire detection, boosting performance even with limited labeled data[84].

- **Efficiency with Large Images:**

CNNs handle large satellite images by focusing on local patterns (convolutions), making analysis scalable.

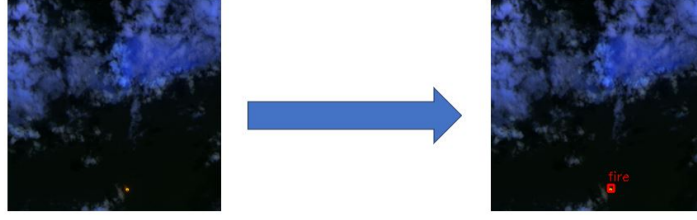
## 2.4.2 Deep Learning Techniques and Models for Satellite Imagery Analysis

In the context of analyzing satellite images for the detection of wildfires, various deep learning-based techniques play an essential role [85]. These techniques rely on the ability to automatically learn and extract important features from the complex visual data in satellite images. Each technique, whether it involves object detection, image segmentation, or classification, uses different models and architectural strategies to locate and classify features relevant to wildfire monitoring. In this section, we will examine how each technique works and highlight representative models that illustrate their effectiveness and practical applications.

### A. Object Detection

Object detection involves not only recognizing what is in an image but also locating it by drawing bounding boxes around specific objects, such as wildfires, vehicles, or buildings. This process is achieved by training the model to predict both the class of an object and its position within the image, often identifying multiple objects at once [86]. The network learns to detect patterns and features that distinguish different objects and then outputs coordinates for bounding boxes as well as their labels. Object detection is especially valuable in satellite imagery, where identifying the exact location of events like wildfires enables rapid response and resource allocation. Popular models for this task include YOLO (You Only Look Once) and Faster R-CNN, which are optimized for speed and accuracy, making them ideal for analyzing large volumes of high-resolution satellite images in real time [87].

To illustrate this, a figure demonstrating DL-based object detection in wildfire imagery (Figure 2.8) will be provided in the following figure.



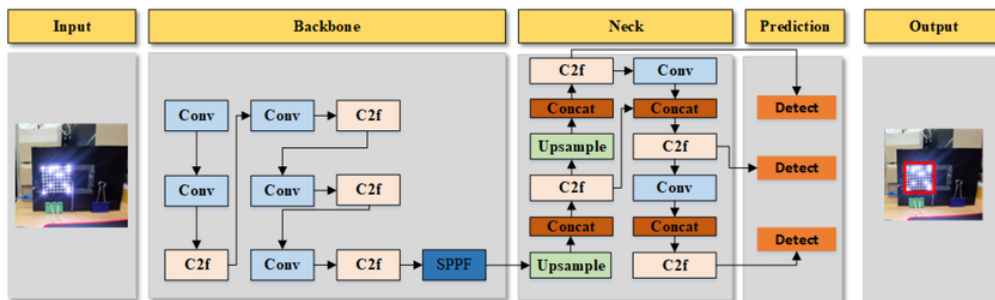
**Figure 2.8** – Example of Deep Learning based object detection for wildfire localization (Yolo V8) : The network identifies and marks the precise location of a fire.

Among the popular models for object detection is YOLO (You Only Look Once):

— **YOLO (You Only Look Once):**

YOLO is a deep learning object detection model that processes an entire image in one pass. It predicts object classes and bounding boxes simultaneously, enabling real-time, accurate detection in a single evaluation[87]. Over time, multiple versions of YOLO have been developed to improve speed and accuracy, addressing new challenges in real-time object detection and adapting to different applications. For example, YOLOv2 (2016) introduced batch normalization, anchor boxes, and dimension clusters to boost detection accuracy and stability. YOLOv8 (2023), developed by Ultralytics, brought significant improvements in performance, flexibility, and efficiency for diverse vision tasks. The most recent YOLO11 by Ultralytics delivers state-of-the-art performance for various tasks, including detection, segmentation, tracking, and classification, making it highly versatile across different AI applications.

**Architecture of YOLOv8:**



**Figure 2.9** – YOLOv8 architecture [61].

The architecture of YOLOv8, as shown in Figure 2.9, consists of several key components designed to efficiently process input images and produce accurate detection results [88]. This architecture includes the following main parts:

- **Input:** The model takes an input image, which could contain multiple objects of interest.
- **Backbone:** This section extracts hierarchical features from the input image using a series of convolutional layers (Conv), the C2f block, and a spatial pyramid pooling module (SPPF). These layers are responsible for progressively capturing spatial and semantic information from the image at multiple scales.
- **Neck:** This part of the network integrates features from different levels of the backbone. It includes C2f and Conv blocks for feature refinement, along with upsampling and concatenation (Concat) layers to merge multi-scale information. This fusion of features is crucial for accurately detecting objects of varying sizes.
- **Prediction:** The refined features are passed to the prediction head, which includes detection layers (Detect). These layers simultaneously predict the bounding box coordinates, objectness score, and class probabilities for each detected object.
- **Output:** The final output consists of the detected objects in the image, complete with bounding boxes and classification labels.

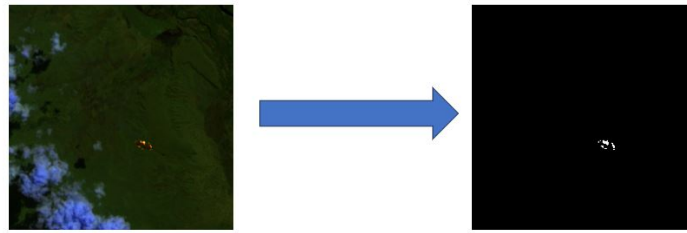
This structured explanation covers all components of the YOLOv8 architecture and how they contribute to the model's ability to detect objects in real time and accurately.

## B. Segmentation

Segmentation takes the analysis a step further by dividing the image into segments or regions and classifying each pixel instead of just drawing a box around an object. In satellite imagery, this could mean identifying every pixel that belongs to fire, water, forests, or buildings, resulting in a detailed and precise map of all features of the image. This pixel-level approach is especially valuable when it is important to know the exact shape, size, and boundaries of objects or regions, rather than simply knowing their locations. Unlike object detection, which provides bounding boxes and general positions

of objects, segmentation delivers a full mask that outlines the complete area covered by each class. Semantic segmentation models like U-Net and DeepLab use CNNs to achieve this fine-grained classification, enabling more accurate monitoring and assessment of wildfire spread, land cover, and other environmental factors in satellite images [89].

To illustrate this, a figure demonstrating DL-based segmentation in wildfire imagery 2.10 will be provided in the following figure.



**Figure 2.10** – Example of Deep learning based segmentation for wildfire detection(U-Net) : The network processes a satellite image and generates a pixel-wise mask.

One of the most widely used models for segmentation is

- **U-Net:** U-Net is a convolutional neural network architecture made mainly for image segmentation. It features a U-shaped design, with a contracting path (encoder) that reduces the image size and captures context using convolution and max-pooling layers, and an expansive path (decoder) that upsamples the data and combines features through skip connections to achieve precise localization. U-Net builds on fully convolutional networks but is modified to handle smaller training datasets while still achieving high accuracy in segmentation. It avoids fully connected layers and uses mirroring to deal with image borders, enabling accurate pixel-level classification. This balance of precision and efficiency has made U-Net a popular choice in medical imaging and various segmentation applications. [90].

**Architecture of U-Net:**

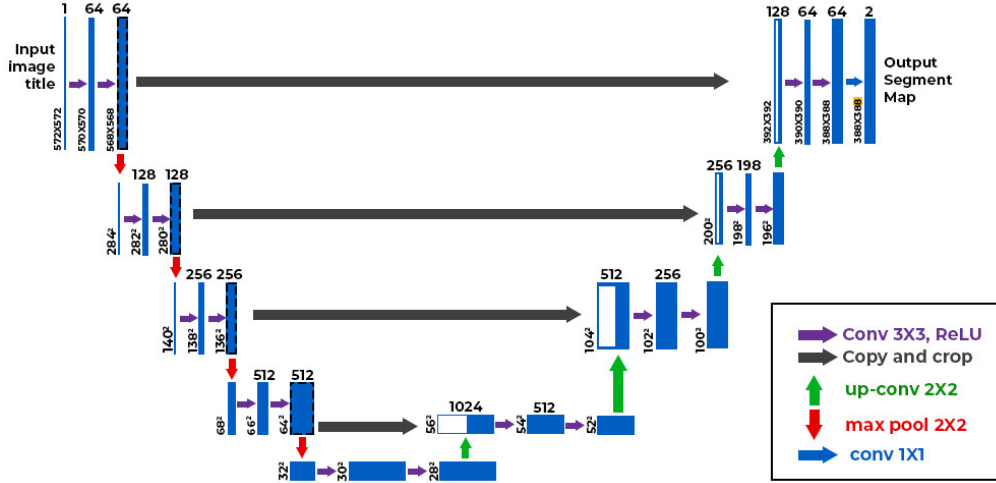


Figure 2.11 – U-Net architecture[91].

The architecture of U-Net, as depicted in Figure 2.11, consists of several essential components designed to process input images effectively and produce accurate segmentation results.

This architecture includes the following main parts:

- **Input:** The U-Net architecture begins with an input image (e.g., 572×572 pixels). This image contains the raw data that will be processed to create a pixel-wise segmentation map.
- **Encoder (Contracting) Path:** The encoder compresses the input image, extracting increasingly abstract features.

At each stage of the encoder, there are convolutional layers (3×3, ReLU) (purple arrows) that learn patterns and refine features, and max pooling (2×2, stride 2) (red arrows) reduces the image size (downsampling) to capture larger-scale features.

The number of feature maps (channels) increases as you move deeper: 64, 128, 256, 512, 1024. This part of the network learns to understand the context and important structures in the image.

- **Bottleneck:** At the deepest part of the U, known as the bottleneck, the architecture has the most abstract and compressed features (1024 channels). This stage captures the global context and high-level information of the image.
- **Decoder (Expanding) Path:** Reconstructs the segmentation map by pro-



gressively increasing spatial resolution.

Up-convolutions (Up-conv  $2 \times 2$ ) (green arrows) double the size of feature maps to restore spatial detail.

Skip connections (copy and crop, gray arrows) link encoder layers to decoder layers at the same resolution.

They combine precise spatial details from the encoder with abstract information from the decoder.

Each decoder block includes convolutional layers ( $3 \times 3$ , ReLU) to refine and smooth the segmentation map.

- **Output:** The final layer uses a  $1 \times 1$  convolution to map feature channels to the desired number of classes (e.g., foreground vs. background).

This creates the output segmentation map, which has the same spatial dimensions as the input, with each pixel classified.

- **SegFormer:** SegFormer is a cutting-edge semantic segmentation model that delivers high accuracy and efficiency across diverse datasets. It can handle challenging tasks like urban scene parsing and satellite image analysis while maintaining computational efficiency. SegFormer's strength lies in its ability to capture local and global dependencies, as well as process contextual information across multiple scales and resolutions. This allows it to produce precise segmentations even in complex and cluttered images, making it a popular choice in remote sensing, environmental monitoring, and medical imaging applications [92].

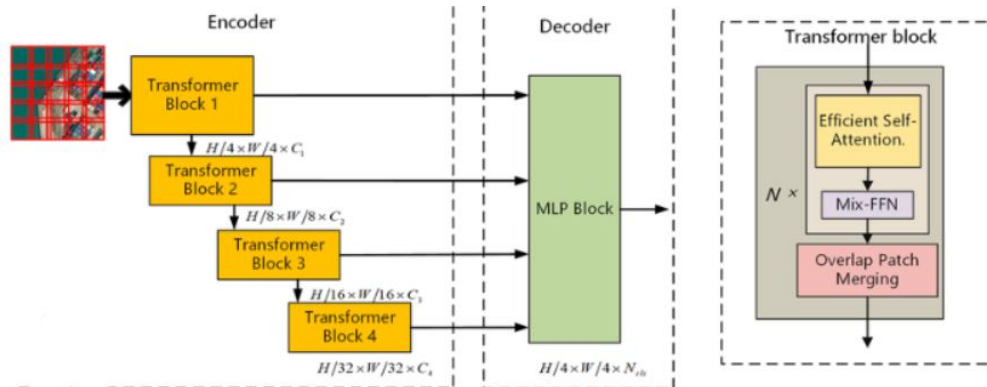


Figure 2.12 – Architecture of Segformer [93].

The architecture of SegFormer, as depicted in Figure 2.12, consists of several essential components [93].

### Architecture of SegFormer:

- **Input Image:** The model begins with a high-resolution input image, which is first divided into patches.
- **Encoder:** The encoder has four hierarchical stages (Transformer Blocks 1–4), each of which:
  - Reduces the spatial resolution of the input (down-sampling).
  - Increases the number of feature channels to capture more abstract, high-level information.

Each transformer block includes Overlap Patch Merging: Merge neighboring patches to form coarser representations while preserving overlap (helps retain fine details).

Efficient Self-Attention: Uses a self-attention mechanism to learn relationships between distant pixels and capture global context.

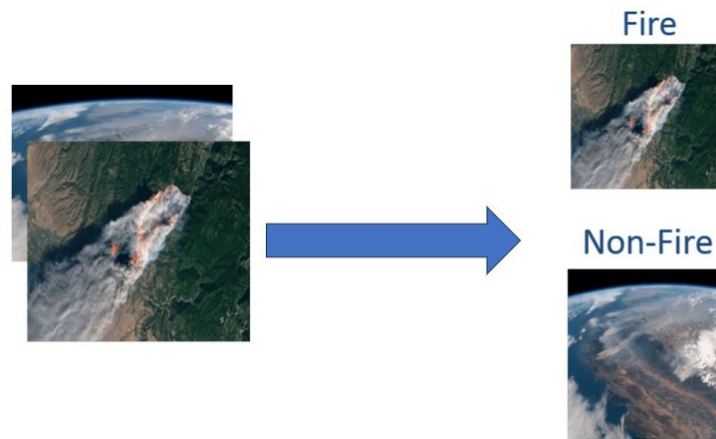
Mix-FFN (Feed Forward Network): Introduces non-linearity and projects features into a richer space.
- **Multi-Level Feature Fusion:** The encoder’s four outputs are downsampled differently ( $1/4$ ,  $1/8$ ,  $1/16$  and  $1/32$  of the original image size), allowing the model to capture features on multiple scales.
- **Decoder:** Consists of a simple Multi-Layer Perceptron (MLP) block.
  - This MLP decoder fuses the multi-level features from the encoder to generate a final, unified representation.
  - Unlike traditional decoders with complex upsampling and skip connections, this lightweight MLP focuses on combining and refining multi-scale features.
- **Output:** The final output of the decoder is a pixel-wise classification map, where each pixel in the original image is assigned a class label (segmentation map).

This architecture is highly efficient, balancing fine detail (local information) and global context to achieve accurate segmentation while remaining lightweight com-

pared to conventional CNN-based architectures.

### C. Classification

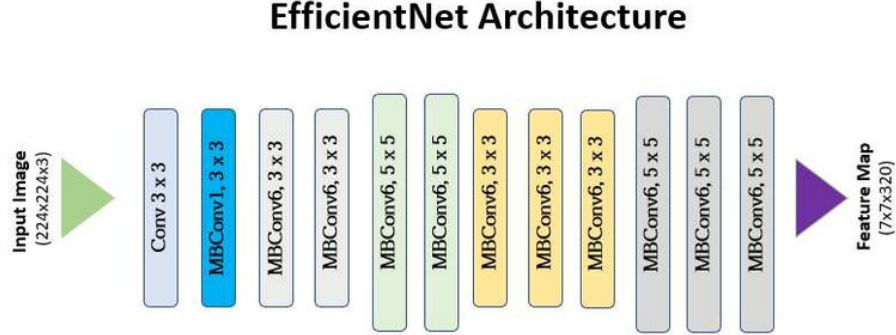
Image classification involves the model analyzing the entire satellite image and assigning it a single label or category, such as 'fire detected' or 'no fire.' This is the simplest form of image analysis using CNNs and is often used as a first step or for rapid assessments when a quick response is needed. Unlike object detection, which identifies and locates multiple objects within an image by drawing bounding boxes, and segmentation, which classifies every pixel to produce detailed maps of each region, classification only provides an overall summary for the whole image [94]. It does not reveal where the fire is located or how much area is affected but tells you whether or not a particular feature (like wildfire) is present. This makes image classification particularly useful for filtering large datasets or for triggering more detailed analyses when a certain class is detected. To illustrate this, a figure demonstrating DL-based Classification in wildfire imagery 2.13 will be provided in the following figure.



**Figure 2.13** – Example of deep-learning based image classification for wildfire detection[95].

- **EfficientNet:** is a deep learning model family for image classification that balances network depth, width, and resolution using a technique called compound scaling. Developed by Google AI in 2019, EfficientNet achieves high accuracy while being computationally efficient, outperforming many traditional models like ResNet and Inception. Its architecture uses mobile inverted bottleneck convolutions (MB-Conv) and squeeze-and-excitation optimization to reduce computational cost with-

out compromising performance. EfficientNet models range from B0 to B7, offering scalability for various applications, from mobile devices to powerful servers.



**Figure 2.14** – Architecture of EfficientNet[93].

The architecture of EfficientNet, as depicted in Figure 2.14, consists of several essential Components[96].

#### Architecture of EfficientNet:

- **Input Image:** The input image size is  $224 \times 224 \times 3$  (RGB image).
- **Initial Convolution:** A  $3 \times 3$  convolutional layer is applied, which helps in capturing low-level features.
- **MBConv Blocks:** After the initial convolution, the core of EfficientNet architecture uses MBConv (Mobile Inverted Bottleneck Convolution) blocks, which are highly efficient and adapted from MobileNetV2.

These blocks consist of:

Depthwise separable convolutions

Squeeze-and-excitation modules for channel-wise recalibration.

Residual connections for easier training.

- **Block Details:** There are multiple MBConv blocks with different configurations, alternating between  $3 \times 3$  and  $5 \times 5$  kernel sizes to capture multi-scale features.

The number of channels and feature map sizes change progressively through these blocks, capturing increasingly abstract and high-level representations.

- **Feature Map:** The final output of these layers is a feature map of size  $7 \times 7 \times 320$ , which can be used for various tasks like classification, object detection, or segmentation.

Deep learning offers a diverse toolkit of Object detection, segmentation, and classification, each plays a vital role in DL, offering complementary methods to extract meaningful insights from complex data. Together, they enable advanced analysis and open new opportunities across fields like environmental monitoring and automated systems.

## 2.5 Conclusion

In this chapter, we explored the intersection of deep learning and remote sensing. It presented key deep learning tasks, object detection, segmentation, and classification used in satellite-based wildfire detection. It detailed how each task works, their applications, and their differences in output and precision. We also explored well-known models such as YOLOv8, U-Net, SegFormer, and EfficientNet, outlining their architectures and roles in processing satellite imagery. This foundation helps in understanding which models are best suited for specific detection needs and prepares us for the next steps in system development.

A comparative overview of deep learning techniques, object detection, segmentation, and classification demonstrated how these methods contribute to more accurate and timely wildfire detection. This foundational knowledge sets the stage for the next chapter, where we transition from theory to practice, detailing the step-by-step methodology for developing an operational wildfire detection system based on satellite imagery.

Chapter



# Research Design and Methodology

## 3.1 Introduction

The rapid advancements in artificial intelligence, particularly in the domain of computer vision, have paved the way for the development of intelligent systems capable of monitoring environmental conditions and supporting disaster management efforts. Among these applications, wildfire detection has emerged as a critical area due to the increasing frequency and intensity of wildfires worldwide.

This chapter presents the methodology employed in the design of a wildfire detection system using satellite data. We begin by outlining the overall architecture of the detection pipeline. Subsequently, we delve into the core components of the system, including dataset acquisition and preprocessing, model architecture and training strategies, and the evaluation metrics used to assess system performance. Finally, we illustrate the system's deployment and interaction flow through UML diagrams.

## 3.2 Global Architecture

The global architecture of the wildfire detection system using satellite imagery is designed as a modular and scalable pipeline that ensures accurate and near real-time detection of wildfires. This end-to-end architecture integrates multiple stages including data collection, preprocessing, model development, evaluation, and deployment, as depicted in the diagram.

The process begins with the satellite imagery dataset, which serves as the raw input. These images are subjected to essential preprocessing operations such as resizing, normalization, and data augmentation to ensure the quality and consistency of input for the deep learning models.

Following preprocessing, the dataset is split into training, validation, and testing sets, which supports reliable model training and robust performance evaluation. Three deep learning models are developed and trained using optimized hyperparameters. These models are assessed based on standard metrics such as precision, recall, and the Dice coefficient (F1 Score), ensuring an objective comparison of their effectiveness.

The best-performing model is then integrated into a web application, allowing end-users to access wildfire detection outputs in near real time.

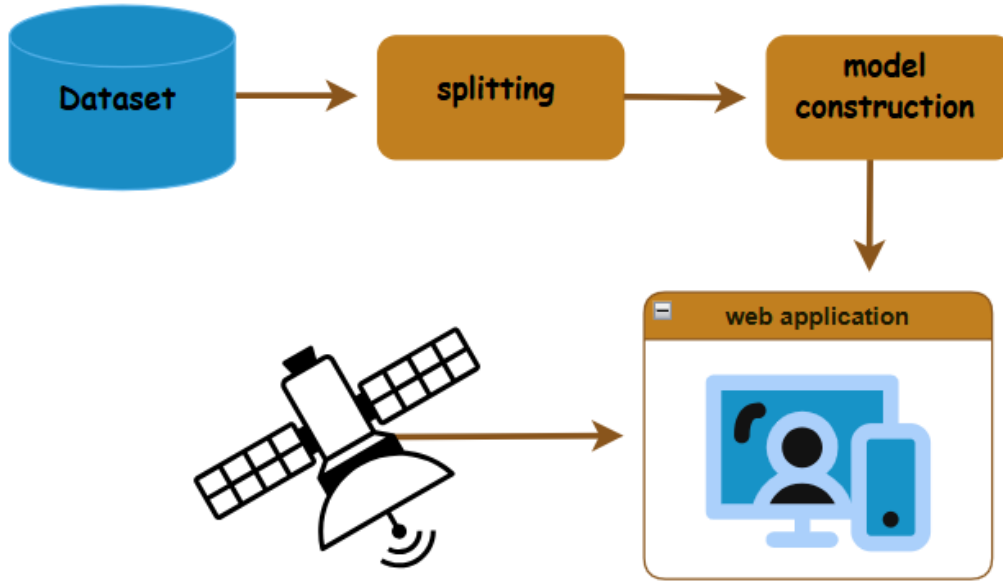


Figure 3.1 – system workflow

### 3.3 Detailed Architecture

This section outlines the architecture and methodology for processing the satellite imagery, from data collection to model construction. It ensures the data is properly prepared for machine learning models through a systematic approach. The following subsections detail the key steps, including data preparation, model selection, and evaluation, to ensure accurate and meaningful analysis.

#### 3.3.1 Dataset description and Preparation

As we previously discussed in Chapter 1, regarding the process of obtaining satellite images for fire detection, we relied on publicly available datasets. Based on this approach, we chose to use the *"Active Fire Detection in Landsat-8 Imagery"* dataset for several reasons. Unlike other datasets, such as the burned areas dataset from Canada, which mainly focuses on post-fire burned regions, this dataset specifically targets active fire events. Additionally, it offers both input images and corresponding labeled masks, enabling supervised learning and precise model training. The dataset also stands out for its broad geographic coverage, capturing wildfire events from diverse regions worldwide, and for its high-quality 10-band multispectral images. These characteristics make it particularly well-suited for developing robust deep learning models for detecting active fires.



**Study Area and Label Selection** The dataset we are using provides satellite imagery from various regions around the world. For our analysis, we specifically focused on the African region, which experienced a significant number of wildfires during the time the images were captured. Given our focus on Algeria, we filtered the dataset to include only images from Africa and paired them with the corresponding Murphy masks.

To prepare the data for model training, satellite images must be processed by an algorithm that generates corresponding fire labels. The images are passed through one of the fire detection algorithms—Murphy, Kumer, or Schroeder—which analyze reflectance values across different channels to generate binary masks labeling the fire-affected pixels. These masks are then paired with the satellite images to create a dataset suitable for training our fire detection model.

The dataset includes several types of masks, such as Kumer, Rumer, and Murphy. Below is a brief explanation of each:

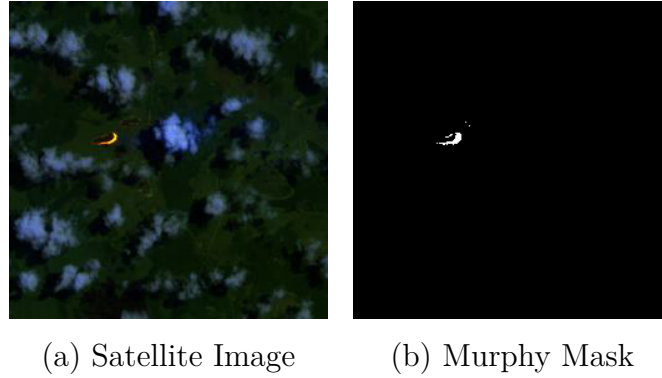
- **Murphy Mask:** The Murphy algorithm identifies fire-affected pixels using spectral ratios from channels c5, c6, and c7. It does not rely on contextual analysis or neighboring pixels, making it a straightforward method for detecting fires. The algorithm directly identifies unambiguous fire pixels and classifies nearby pixels as fire-affected using a 3x3 pixel window. This simplicity and efficiency make the Murphy mask ideal for our study[56].

- **Kumer Mask:** The Kumer algorithm (Kumer and Roy, 2018) uses channels c2 to c7 and performs contextual analysis within a 61x61 pixel window. While this method is sensitive to regional context, it is computationally intensive, and its complexity is not required for our study. Therefore, we chose the Murphy mask due to its simplicity and better performance[56].

- **Schroeder Mask:** The Schroeder algorithm (Schroeder et al., 2016) relies on conditions based on seven Landsat-8 channels and also incorporates contextual analysis. However, it is more complex and prone to false positives, particularly in regions with high variability. Given these limitations, we found the Murphy mask to be more reliable and effective for detecting fire pixels in the African region[56].

Among these, we chose the Murphy mask for our study due to its ability to effectively identify fire-affected areas and distinguish between fire types. Unlike other algorithms, Murphy’s approach directly identifies unambiguous fire pixels, making it particularly suited for our model’s labeling needs. An example of a satellite image and its corresponding Murphy mask is

presented in Figure 3.2.



**Figure 3.2** – Active Fire Satellite Image vs Its Murphy Mask Label.

Initially, we filtered the dataset to focus on images from the African region, resulting in a diverse set of fire events. However, a second filtering step was applied to ensure that only images with significant fire-affected pixels were included. Specifically, we retained only images with at least four fire pixels in their corresponding masks. This step was crucial to eliminate images with minimal fire activity, ensuring that the remaining dataset contained sufficient information for meaningful model training. After this filtering, the dataset was reduced to **5,701 images**, all containing substantial fire pixels, ready for model training.

**Band Composition and Selection** As discussed in Chapter 1, satellite images can be represented in various ways depending on the sensor and dataset characteristics. For our chosen dataset, we use Landsat 8 multispectral imagery provided in *(.tif)* format.<sup>1</sup>

While the Landsat 8 sensor officially captures data across 11 spectral bands, our dataset comprises 10 bands in TIFF format. Table 3.1 summarizes the spectral bands available on Landsat 8, illustrating the diverse wavelength ranges each band covers. It is important to note that each satellite sensor is designed with a unique set of spectral bands tailored for various remote sensing applications, which can vary between missions.

For our fire detection task, we focus on extracting just three bands: **Band 7 (SWIR2)**, **Band 6 (SWIR1)**, and **Band 2 (Blue)**. This choice is motivated by the unique properties of these bands in highlighting fire-related phenomena. Bands 6 and 7, which correspond to the shortwave infrared (SWIR) regions, regions experiencing active fires appear brighter,

---

1. TIFF (Tagged Image File Format) is a widely used raster image format that supports multispectral satellite data with multiple bands and lossless compression, commonly employed in remote sensing, GIS, and professional imaging.

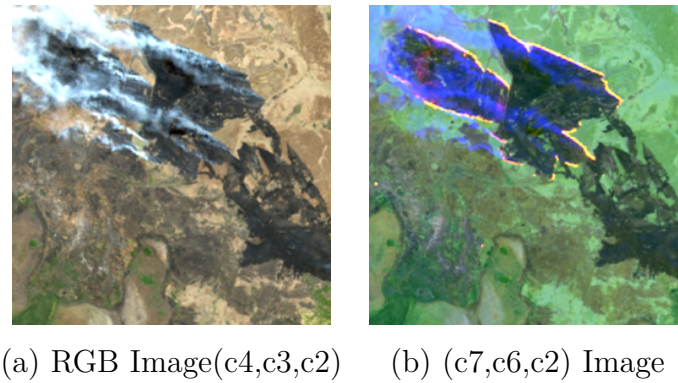
**Table 3.1** – Spectral bands of the Landsat 8 OLI and *TIRS* instruments[97]

Band Number	Band Description	Wavelength Range ( $\mu\text{m}$ )	Resolution (m)
Band 1	Coastal/Aerosol	0.435 – 0.451	30
Band 2	Blue	0.452 – 0.512	30
Band 3	Green	0.533 – 0.590	30
Band 4	Red	0.636 – 0.673	30
Band 5	NIR	0.851 – 0.879	30
Band 6	SWIR-1	1.566 – 1.651	30
Band 7	SWIR-2	2.107 – 2.294	30
Band 8	Pan	0.503 – 0.676	15
Band 9	Cirrus	1.363 – 1.384	30
Band 10	<i>TIR-1</i>	<i>10.60 – 11.19</i>	100
Band 11	<i>TIR-2</i>	<i>11.50 – 12.51</i>	100

Spatial resolution indicates the ground area represented by each pixel in the satellite image. For example, 30 m means each pixel covers 30 m by 30 m on the Earth's surface.

indicating strong thermal activity. Band 7, in particular, offers a clear distinction of active fire areas, thus improving fire detection effectiveness.[98].Meanwhile, the Blue band (Band 2) helps differentiate between smoke and atmospheric effects[99].

Figure3.3 presents a visual comparison between the standard RGB satellite image and the selected spectral bands (C7, C6, and C2), illustrating the enhanced fire visibility provided by our chosen band combination.

**Figure 3.3** – Comparison of RGB image and selected bands image .

The selected bands are extracted from the original multispectral TIFF images and saved separately as PNG images to facilitate further image processing and model training. This targeted selection allows us to focus the model on the most relevant spectral information for accurate and efficient fire detection.

**Data Availability** The dataset used for this study, which includes satellite imagery and corresponding fire masks, is publicly available on GitHub. It can be accessed via the following link: <https://github.com/pereira-gha/activefire>

### 3.3.2 Data Splitting

Data splitting is a fundamental step to ensure the model’s ability to generalize beyond the training data. To mitigate risks of overfitting and to enable reliable performance evaluation, we adopted a consistent data-splitting strategy across all three models used in this study. Specifically, the dataset was partitioned into 70% for training, 15% for validation, and 15% for testing.

This approach allows us to properly tune the models during development (using the validation set), while objectively assessing their performance on unseen data (using the test set).

### 3.3.3 Model Construction

#### 1.3.3.1 Pre-processing

Preprocessing is a critical step in deep learning pipelines before feeding images into our models cause it ensure clean , consistent and suitable input for the model [100] ,and for that we apply this preprocessing steps: • **Resizing:** we resize all images to a fixed dimension to ensure uniform input size, cause may dataset contain different sizes , in our models for each experiment we use different input: in experiment 1 we use (256x256), in the second experiment (512x512) and in last experiment (640x640).

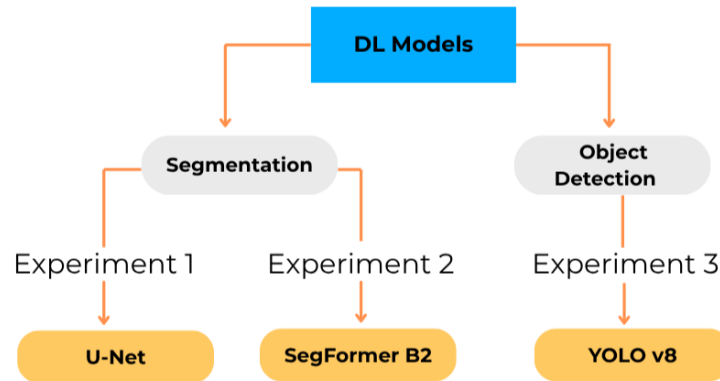
- **Data Normalization:** Normalization plays a crucial role in adjusting the pixel values to a common scale, allowing the model to better learn the patterns in the data. *In Experiment 1* , a simpler normalization technique was applied, where the pixel values of the image were scaled from the range of 0–255 to a floating-point range of 0–1. This was done by dividing each pixel value by 255. *For Experiment 2* , a more advanced normalization method was used: mean normalization. In this case, the pixel values were scaled to the range of [-1, 1] by applying a mean of 0.5 and a standard deviation of 0.5 for each three channel. This technique was intended to enhance model performance by ensuring that pixel values have a more consistent scale across different channels.

- **Data Augmentation:** (Applied Only to Training Data)

we make some random transformations by increasing the diversity of training data, helping the model generalize better new data, our augmentations in both 3 experiments include these common transformations: Random Crop: forces model to recognize fires from partial views and on different regions, and this helps model learn partial objects. Sharpen: enhances edges and details to help the model detect boundaries more clearly especially for small fire borders. Flips: by mirroring images horizontally and vertically to make the model invariant to orientation which means learns that direction doesn't matter. RandomBrightnessContrast: adjust lighting conditions which makes model robust to different times of day/night/clouds.

### 1.3.3.2 Used Models

In this work, we explore various models to address the challenge of detecting forest fires using satellite imagery. After extensive experimentation, we found that two segmentation models and one object detection model yielded promising results. These models are presented in the following subsection. (See 3.4)

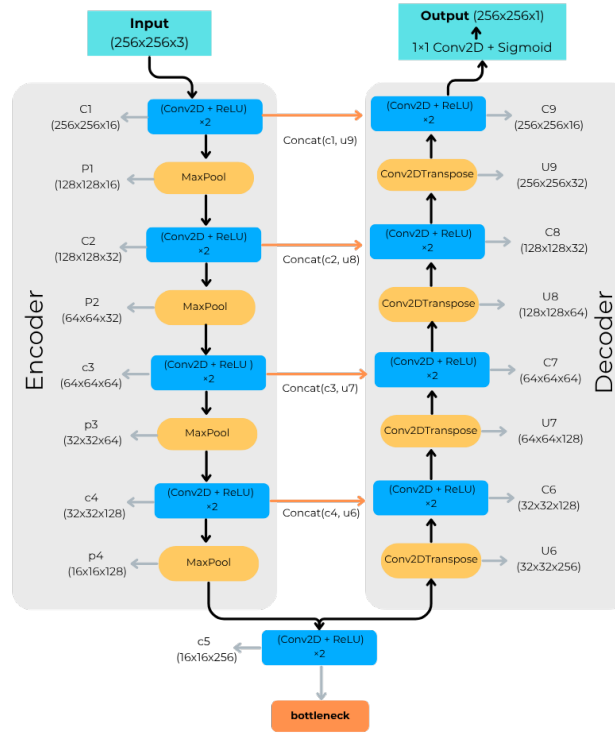


**Figure 3.4 – Our Models**

#### 1. U-Net

In our wildfire segmentation task, we adopted a U-Net architecture, a well-known convolutional neural network architecture used for segmentation tasks; it excels in pixel-level classification. As a global architecture of U-Net, it follows a symmetric encoder-decoder structure with skip connections, as illustrated in Figure 3.5.

— **Input:** Satellite images of shape  $(256 \times 256 \times 3)$ .



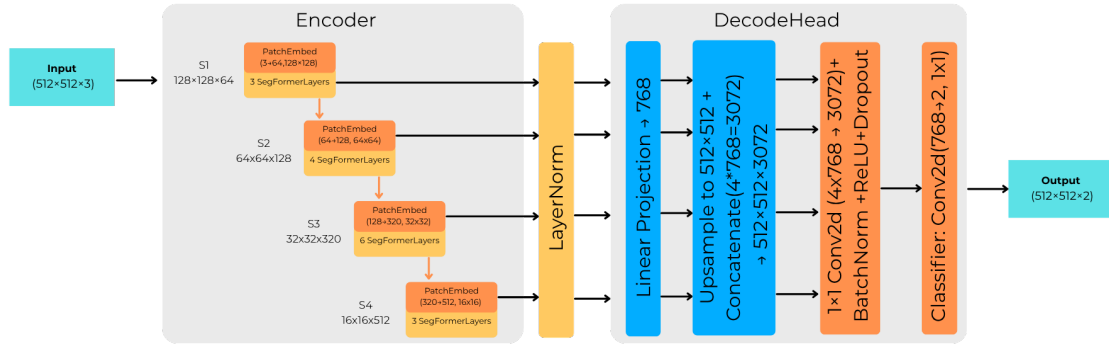
**Figure 3.5** – U-Net architecture used for our segmentation task.

- **Contracting Path (Encoder):** Each block consists of two convolutional layers with ReLU activation, Batch Normalization, and MaxPooling for downsampling. The number of filters doubles at each level, starting from a base of 16.
- **Expanding Path (Decoder):** Upsampling is done using transposed convolutions. Each upsampled output is concatenated with the corresponding encoder feature map (skip connection), followed by two convolutions.
- **Output Layer:** A final  $1 \times 1$  convolution reduces the feature map to a single output channel for binary segmentation. It combines all feature channels at each pixel location into one value. A sigmoid activation is applied to produce a per-pixel probability between 0 and 1.

- One of the most strength point of u-net is precision in localization when it excels at delineating fire boundaries due to its pixel-wise segmentation capability also, it can handles Low-Resolution Data thanks to skip connections that help recover spatial details, which is crucial for detecting small wildfires in satellite images, in the other hand, it achieves high accuracy in segmentation tasks.

## 2. SegFormer B2

In our wildfire segmentation task, we also employed **SegFormer-B2**, a recent transformer-based semantic segmentation model that combines hierarchical *Vision Transformers* (*ViTs*) with a lightweight and efficient decoder. As a global architecture, SegFormer follows an encoder-decoder structure that is well-suited for pixel-wise classification, as illustrated in Figure 3.6.



**Figure 3.6** – SegFormer-B2 architecture used for wildfire segmentation.

- **Input:** Satellite images of shape  $(512 \times 512 \times 3)$ .
- **Hierarchical Encoder (SegFormerEncoder):** The encoder is composed of four hierarchical stages. Each stage reduces spatial resolution and increases channel depth through patch embedding and SegFormerLayers with efficient self-attention. The outputs from all stages represent multi-scale contextual features, normalized by LayerNorm.
- **Decoder (SegFormerDecodeHead):** The decoder takes the outputs from all encoder stages and projects each to 768 channels using linear layers. These are upsampled to  $512 \times 512$ , then concatenated ( $4 \times 768 \rightarrow 3072$  channels). A  $1 \times 1$  convolution reduces the feature map to 768 channels, followed by BatchNorm2d, ReLU, and Dropout ( $p = 0.1$ ). A final  $1 \times 1$  convolution classifier maps the result

to 2 channels, producing the segmentation output.

- **Output Layer:** The final segmentation map of shape  $512 \times 512 \times 2$  assigns each pixel a class label, enabling high-resolution binary segmentation of wildfire-affected areas.
- This model is useful in Captures Global Context: Self-attention helps identify fires spread across large areas, which CNNs might miss.

### 3. YOLO v8

Real-time object detection model that predicts bounding boxes and class probabilities with a single forward pass, enabling fast and efficient detection. In its architecture, it consists of a backbone, which is a CNN that extracts features from the input image. And also neck is a Feature Pyramid Networks (FPN) aggregate multi-scale features for detecting objects of varying sizes, and lastly, a head for bounding box regression and classification that predicts class labels.

- YOLO models have been proven effective in detecting wildfire flames and smoke from UAV images with good precision and recall

#### 1.3.3.3 Model Hyperparameters and Configuration

##### — Learning Rate (lr):

**Definition:** The learning rate controls the step size for updating model weights during optimization based on the loss gradient[101]. **Impact:** Too high a value may cause unstable training and overshoot the optimal parameters, while too low a value slows convergence[102]. **Tuning:** To use an appropriate learning rate, we should experiment or use learning rate schedules that can help adjust the learning rate dynamically during training [101].

##### — Batch Size:

**Definition:** The number of training examples is processed in one iteration before updating parameters, affecting memory usage and training stability [103]. **Impact:** Small batch sizes often lead to more stable and accurate gradients and improve generalization, but slow the training. Larger batch sizes speed up training but use more memory and may produce a noisier gradient [104].



— **Epochs:**

**Definition:** Defines the number of complete passes through the training dataset, ensuring that every data sample is used to update the model's parameters [105].

**Impact:** The number of epochs can affect convergence, using multiple epochs:

- can improve Model Performance by allowing it to learn better from data.
- We can apply early stopping to avoid overfitting.
- Optimize the training process by enabling gradual learning.

Using too many epochs can lead to overfitting, where the model performs well on training data but poorly on unseen data.

**Tuning:** Experimentation and early stopping help find the balance between underfitting and overfitting [106].

— **Loss Functions:** They are fundamental components in machine learning that quantify the difference between a model's predicted outputs and the actual target values (ground truth).

They provide a numerical measure of error or "loss," which the model aims to minimize during training to improve its performance. Some of these loss functions are [107]:

1. **Focal Loss:** it is designed to address the problem of class imbalance in object detection/segmentation, it modifies the standard cross-entropy loss by adding a modulating factor that down-weights well-classified examples and focuses on hard misclassified samples [108].

**Formulation:**

$$FL(p_t) = -\alpha_t(1 - p_t)^\gamma \log(p_t)$$

Where:

- $p_t$ : Model's estimated probability for the true class.
- $\alpha_t$ : Weighting factor for class balancing (typically  $\alpha \in [0, 1]$  for class 1,  $1 - \alpha$  for class 0).
- $\gamma$  (gamma): Focusing parameter ( $\gamma \geq 0$ ); higher values reduce the loss for easy examples [109].

**Key Features:**

- Reduces contribution from easy negatives (common in imbalanced datasets).

- Improves performance on rare classes (e.g., foreground vs. background in object detection) [108]

2. **Dice Loss:** It is designed to stabilize training in segmentation tasks with missing/empty labels.

**Formulation:**

$$\mathcal{L}_{Dice} = 1 - \frac{2|Y \cap \hat{Y}| + \epsilon}{|Y| + |\hat{Y}| + \epsilon}$$

Where:

- $Y$  = Ground truth mask
- $\hat{Y}$  = Predicted mask
- $\epsilon$  = Smoothing parameter (usually a small value like  $1 \times 10^{-5}$  to  $1 \times 10^{-7}$ )

The  $\epsilon$  prevents division by zero and stabilizes training, especially when dealing with empty or missing labels. Proper tuning of this parameter helps maintain numerical stability and improves segmentation accuracy[general knowledge] [110].

3. **Binary Cross-Entropy (log loss):** is a loss function commonly used in binary classification tasks in machine learning; it measures the difference between the actual class labels (0 or 1) and the predicted probabilities output by a model [111].

The function is smooth and differentiable, making it suitable for gradient-based optimization methods like stochastic gradient descent. Also, this loss function penalizes incorrect classifications more severely as the predicted probability diverges from the true label, encouraging the model to be calibrated (predictions reflect true probabilities) [112].

4. **Box Loss:** Measures the error in bounding box predictions.
5. **CLS Loss:** Classification loss (how well objects are classified into categories).
6. **DFL Loss :** Distribution Focal Loss (DFL) for better box regression precision.

— **Optimization Algorithms:**

Optimization algorithms are fundamental components enabling models to learn, they guide how model parameters (weights and biases) are updated to find the optimal

values that give best performance and minimize the loss function .

- **Adam (Adaptive Moment Estimation):** is one of the most widely used optimization algorithms that combines the benefits of AdaGrad (adaptivity for sparse gradients) and RMSProp (moving average of squared gradients), from its advantages, adaptive learning rates reduce the need for manual tuning. Works well with sparse gradients and non-stationary objectives. it also has fast convergence for many deep learning tasks due to adaptive learning rates and momentum [113]

The following table summarizes the hyperparameter values used for each model.

**Table 3.2** – Comparison of Model Training Configurations

Model	Epochs	Batch	Learning Rate	Early Stopping	Loss Function	Optimizer
U-NET	20	8	1e-3	Yes (patience=5)	BCE	Adam
SegFormer B2	30	8	1e-4	Yes (patience=7)	Focal + Dice	Adam
YOLO V8	100	16	1e-3	Yes (patience=20)	DFL + BCE+ CLS	AdamW

#### 1.3.3.4 Performance Metrics

1. **Recall:** Measures the proportion of actual positive pixels/regions correctly identified out of all actual positive pixels.

The recall is defined by:

$$\text{Recall} = \frac{TP}{TP + FN} \quad (3.1)$$

where:

- TP = True Positives (correctly predicted positive pixels)
- FN = False Negatives (positive pixels missed by the model)
- It is ideal and critical for safety-critical applications where high recall ensures that nearly all relevant objects or regions are detected, which is essential in fields like wildfire detection.

2. **Precision:** Measures the proportion of correctly predicted positive pixels/regions relative to all predicted positives. The precision is defined by:

$$\text{Recall} = \frac{TP}{TP + FP} \quad (3.2)$$

where:

- TP = True Positives (correctly predicted positive pixels)
- FP = False Positives (The number of negative pixels incorrectly predicted as positive pixels)

3. **Dice Coefficient (F1 Score):** It quantifies the similarity between two sets (predicted and ground truth segmentation masks) by calculating the ratio of twice the intersection to the sum of their sizes. The Dice Coefficient, equivalent to the F1 Score, is defined as:

$$\text{DSC} = \text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3.3)$$

- It is ideal for balancing the trade-off between false positives (precision) and false negatives (recall)
4. **Intersection over Union (IoU):** It is a metric primarily used to evaluate the accuracy of an object detection or segmentation model. It measures the overlap between the predicted bounding box (or segmentation) and the ground truth bounding box (or segmentation). It is crucial for assessing how well a model localizes objects, rewarding predictions with higher spatial overlap, even if the bounding box coordinates do not exactly match the ground truth

$$\text{IoU} = \frac{\text{Area of Union}}{\text{Area of Overlap}} \quad (3.4)$$

Where

- Area of Overlap: The region where the predicted and actual bounding boxes overlap.
- Area of Union: The total area covered by both the predicted and actual bounding boxes combined [114].
- IoU values range from 0 to 1, where 1 indicates perfect overlap. It is a key metric in tasks such as image segmentation and object detection, used to quantify how well the model's predictions match the true objects [115].

5. **Confusion Matrix:** It is a table used to visualize the performance of a classification model by comparing predicted labels against true labels [116].

For a binary classification, the confusion matrix is a 2x2 table with the following components:

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

**Table 3.3** – Confusion Matrix

Where

- True Positive (TP): Correctly predicted positive cases.
- False Negative (FN): Actual positives incorrectly predicted as negative.
- False Positive (FP): Actual negatives incorrectly predicted as positive.
- True Negative (TN): Correctly predicted negative cases.
- It allows the calculation of various performance metrics, such as accuracy, precision, recall, and other performance metrics.

### 3.3.4 Model Deployment

Deployment is a critical phase that involves transitioning a model from a development environment to an operational one, where it can interact with end-users. This process ensures that the model is not only available but also performs effectively in practical applications, providing real-time insights. Deploying a model is essential for transforming it into a fully operational system that can deliver real-time outcomes and meet the needs of its users [117].

For our project, the wildfire detection model must be integrated into a real-time **web application**, allowing users to monitor fire outbreaks using satellite imagery. The web application will provide an interface for users to interact with the system, facilitating the visualization of the detection results.

To understand the conceptual framework of this project, we will examine two types of UML diagrams:

- a. **Use Case Diagram** The shown Figure 3.7 presents the use case diagram. the admin actor can perform the multiple use cases such as uploading an image, enter coordinates, manage stations. Meanwhile, the station actor can view alerts and is involved in receiving alerts sent by the system. The system also includes a use case to get zone images from the map, which supports the run simulation use case, and to send alerts to stations.

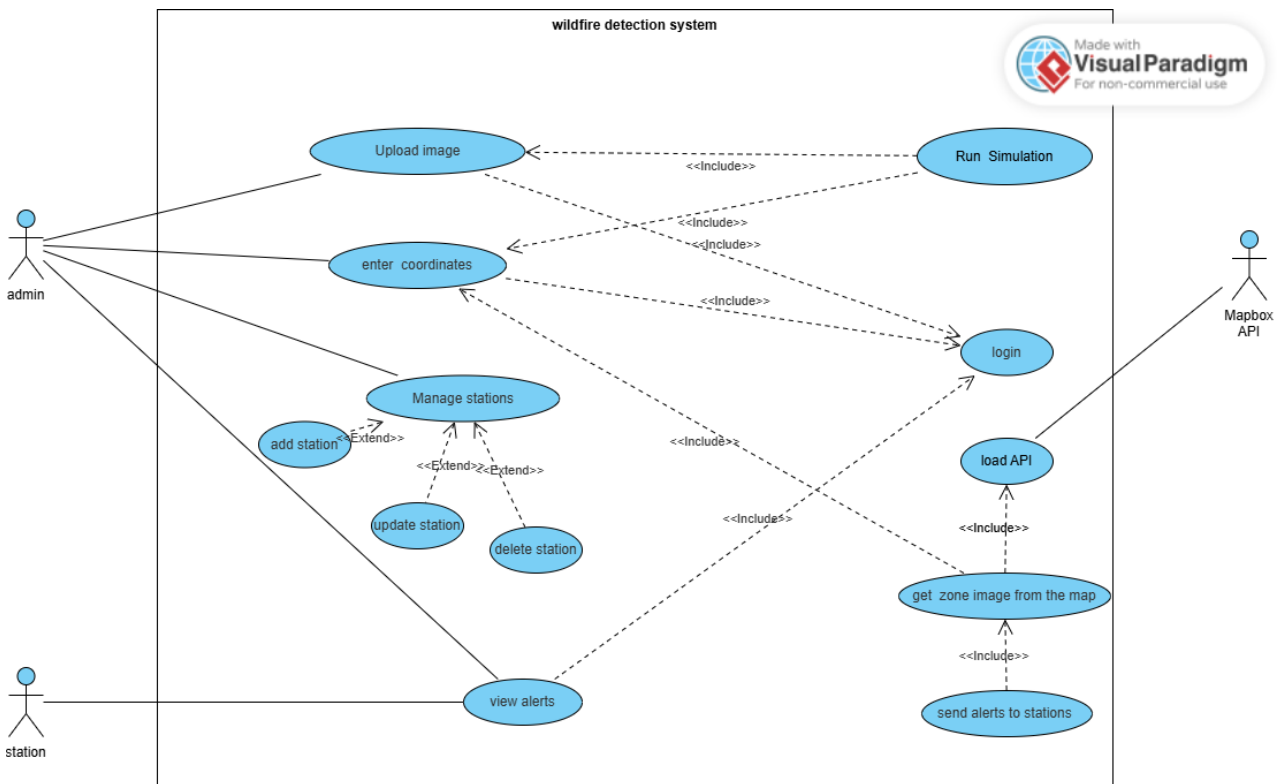


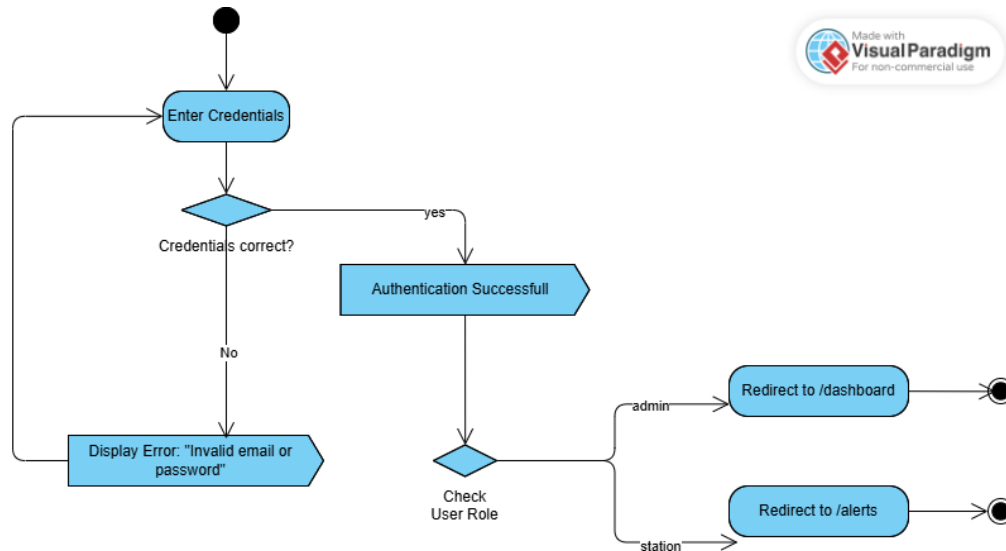
Figure 3.7 – Use Case diagram

- b. **Activity Diagram** This diagram will provide a deeper understanding of the workflow of the system, from receiving satellite images to generating and displaying the detection results.

#### — CASE 1: Login

The figure below 3.8 shows the process when a user enters their credentials (email and password) on the login page. The system checks if the credentials are correct. If they are not correct, an error message ("Invalid email or password") is displayed, and the user is prompted to re-enter their credentials. If the credentials are correct,

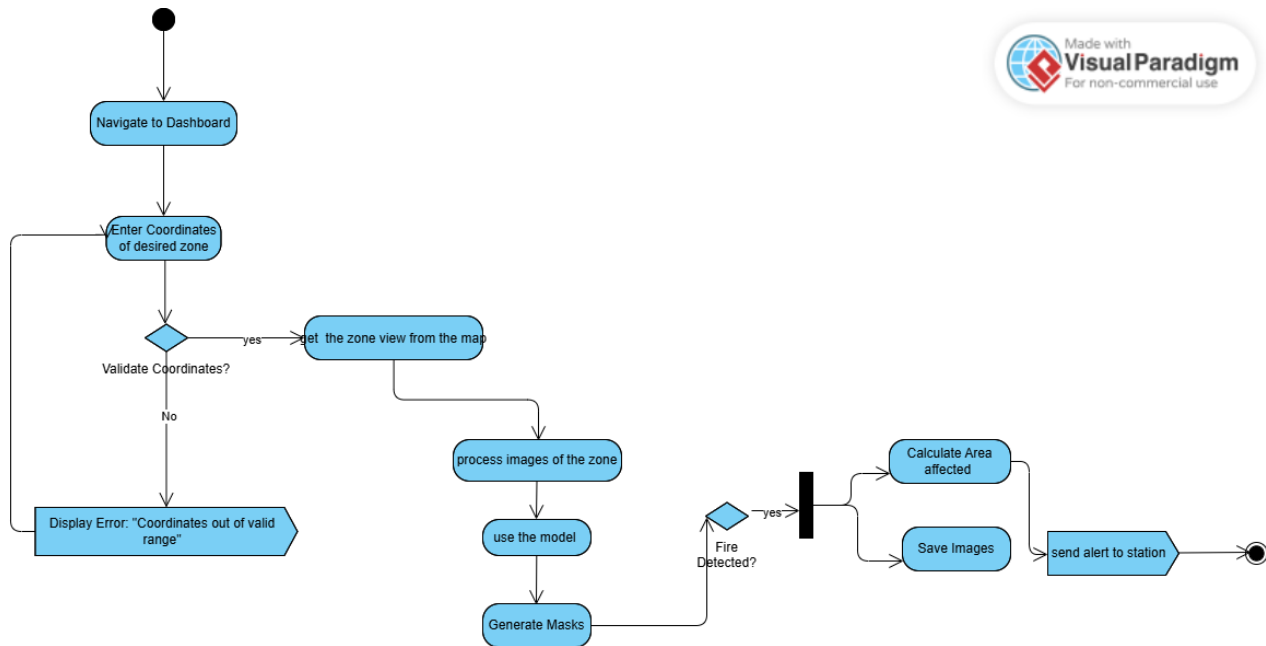
authentication is successful. After successful authentication, the system checks the user's role. If the user is an admin, they are redirected to the dashboard; if the user is a station (receiver), they are redirected to the alerts page. The process ends once the user is redirected to the appropriate page.



**Figure 3.8** – Activity Diagram for Login Process

### — CASE 2: Analyse Map

Figure 3.9 shows the analysis map process when the admin navigates to the dashboard. It enters the coordinates of the desired zone. The system validates them to ensure they are within a valid range. If they are invalid, an error message ("Coordinates out of valid range") is displayed, and the user returns to enter new coordinates. If they are valid, the system retrieves a zone view from the map. Next, it processes the images of the zone, applies a model to analyze them, and generates masks to identify potential fire areas. The system then checks if a fire is detected. If no fire is detected, the process ends. If fire is detected, it calculates the affected area, saves the images, and sends an alert to the station. The process concludes after the alert is sent or if no fire is detected.

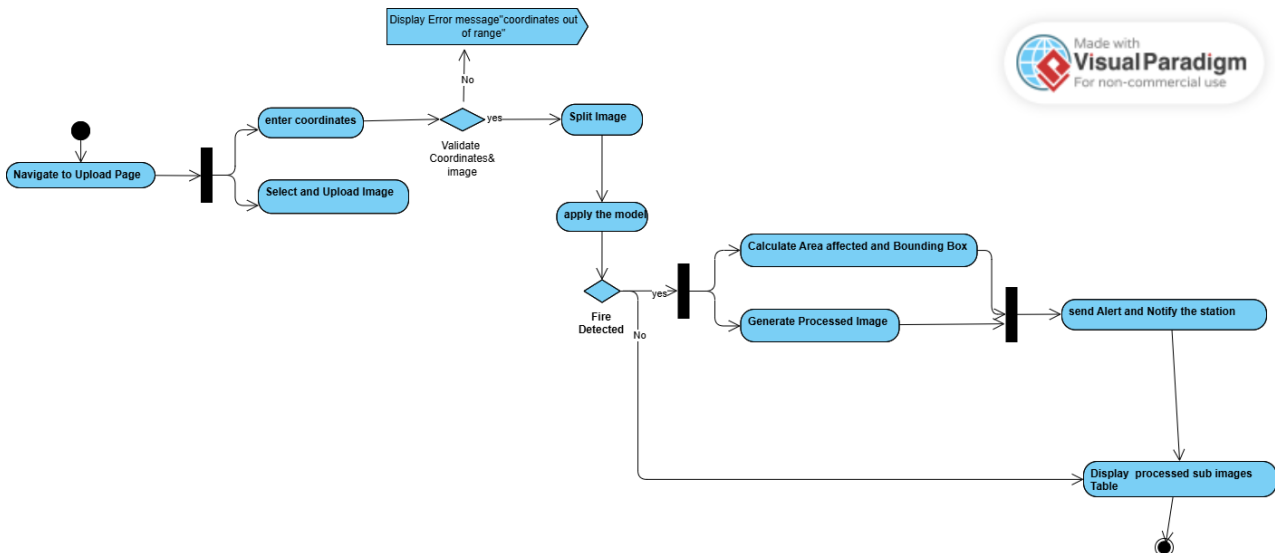


**Figure 3.9** – Activity Diagram for Map analyse

### — CASE 3: Run simulation

In figure 3.10 when an admin navigates to the upload page. The admin enters coordinates and selects an image to upload. The system validates the coordinates and the image. If validation fails, an error message ("coordinates out of range") is displayed, and the admin is prompted to re-enter the coordinates. If validation succeeds, the image is split into manageable sections. The system then applies a model to analyze the split image. Next, it checks if fire is detected. If no fire is detected, the process ends. If fire is detected, the system calculates the affected area and bounding box, generates a processed image, sends an alert to the station, and finally displays the processed sub-images in a table. The process concludes after the results are displayed.





**Figure 3.10** – Activity Diagram for active fire Simulation

#### — CASE 4: Manage stations

In figure below 3.11 when an admin navigates to the stations page. It can view existing stations. From there, the admin chooses an action: delete a user, edit a receiver, or add a new station. If the admin selects "delete user," the system prompts for confirmation of deletion. If confirmed, a success message is displayed, and the process ends. If not confirmed, the process returns to viewing existing stations. If the admin selects "edit receiver," the system allows updating the station form. The system validates the update form; if valid, a success message is displayed, and the process ends. If invalid, an error message is shown, and the admin can retry. If the admin selects "add a new station," they fill out the station form. The system validates the add form; if the email is already registered, an error message ("Email already registered") is displayed, and the admin can retry. If the form is valid, the new station is saved, a success message is displayed, and the process ends.

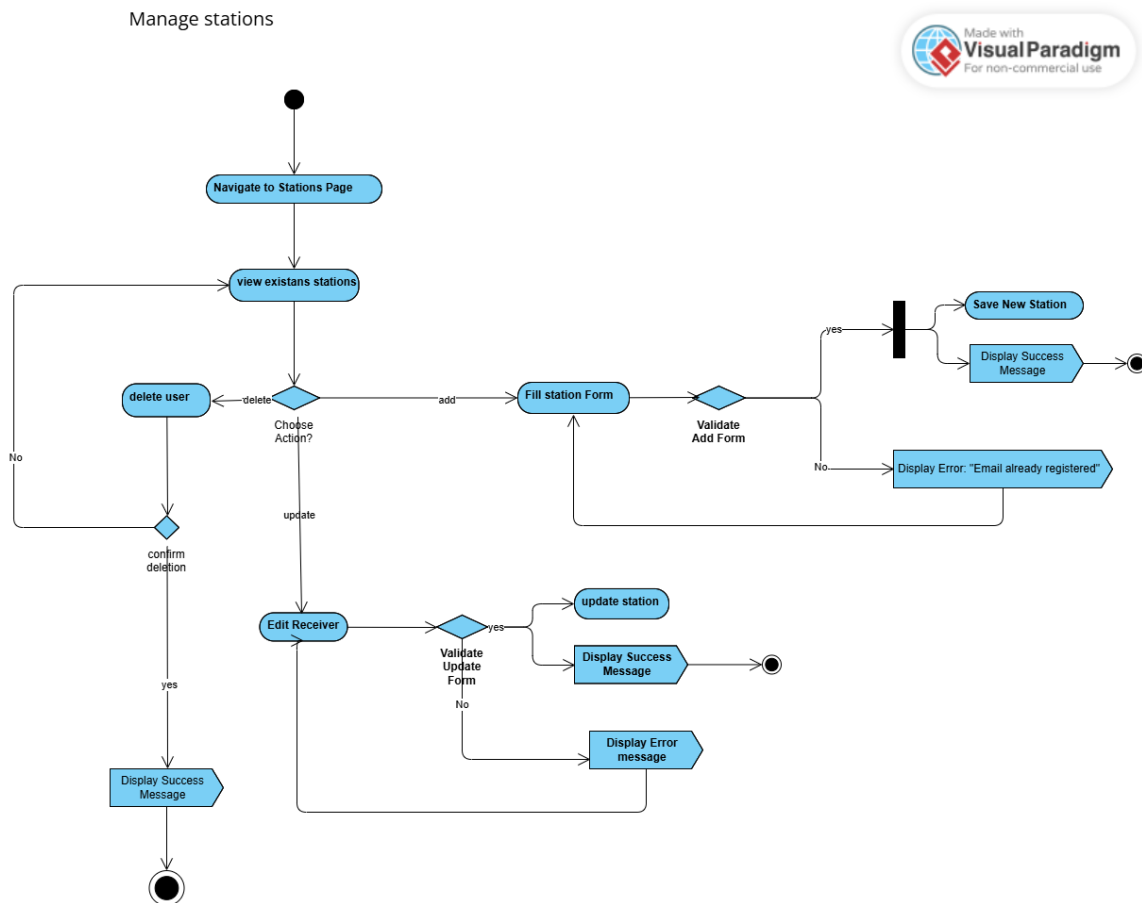
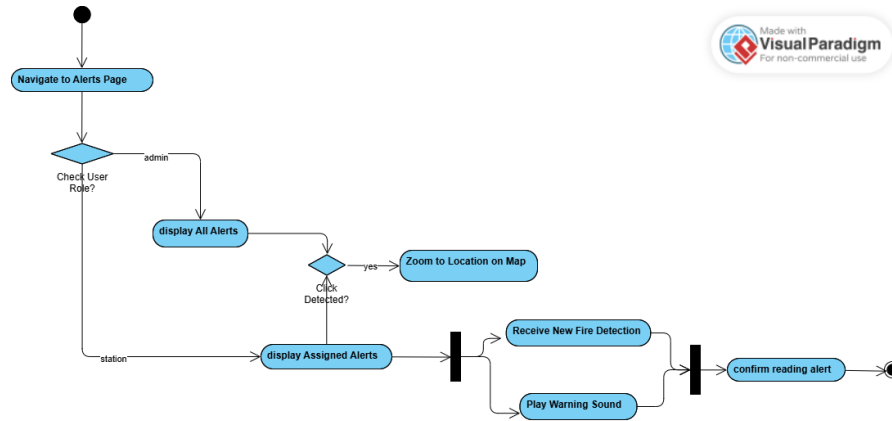


Figure 3.11 – Activity Diagram for Managing Stations

#### — CASE 5: View alerts

Figure 3.12 when a user navigates to the alerts page. The system checks the user's role to determine if they are an admin or a station. If the user is an admin, all alerts are displayed. If the user is a station, only the assigned alerts are displayed. For displayed alerts, the system checks if the user clicks on a detected alert. If yes, the map zooms to the location associated with that alert. Additionally, if a new fire detection occurs while on the alerts page, the station receives a new alert, and a warning sound is played. The process ends after confirming the reading of the alert.



**Figure 3.12** – Activity Diagram for viewing Alerts

### 3.4 Conclusion

This chapter aimed to present the design and methodology behind the development of a wildfire detection system that utilizes satellite imagery. We explored the key stages of the process, including data acquisition, preprocessing, model selection, and training configurations. To aid in understanding the system's design, we also explored two UML diagrams: the use case diagram and the activity diagram, which helped visualize the system's interactions and workflow.

The next chapter includes analyzing the quantitative and qualitative outcomes of the models and comparing their suitability for deployment. Additionally, we will describe the deployment process, shedding light on how the selected model was integrated into a real-time web application for effective wildfire monitoring in real-world scenarios.

Chapter

IV

# Results and System Deployment

## 4.1 Introduction

In this chapter, we present the experimental results and performance evaluation of our deep learning-based wildfire detection system. The chapter begins by outlining the development environments and key libraries used throughout the project. We then present both quantitative results, such as F1 Score, Recall, and Precision, and qualitative outputs, including training curves, confusion matrices, and visual comparisons. Additionally, the chapter includes a comparative analysis that justifies the selection of the most suitable model for deployment in a wildfire monitoring application. Finally, the chapter concludes with a description of the model deployment process, detailing how the selected model was integrated into a web-based application designed to support wildfire monitoring and visualization.

## 4.2 Environment and Libraries

In this section, we describe the various environments and libraries used during the development and deployment of our wildfire detection system.

### 4.2.1 Environment

Here, we discuss the three primary environments used in the project: Google Colab, Kaggle, and VS Code. Each environment provided specific tools and configurations suited to various aspects of model development, training, and evaluation.

#### 1. Google Colab

Google Colab<sup>1</sup> is a cloud-based platform that allows users to run Jupyter notebooks in an interactive environment with access to powerful computational resources like GPUs and TPUs. It is especially beneficial for deep learning projects, where the training process can be resource-intensive.

For this project, we used Google Colab to train and evaluate two of our deep learning models. Google Colab provides free access to GPUs, which were crucial for training the models on large satellite datasets. Specifically, we configured Colab to use an NVIDIA Tesla T4 GPU, known for its high performance in deep learning tasks.

---

1. <https://colab.research.google.com/>

- **Google Colab Configuration:**

- **GPU:** NVIDIA Tesla T4
- **RAM:** 13 GB
- **Disk Usage:** 100 GB (via Google Drive integration)
- **Free Usage:** it offers up to 4 hours every 2 days or 1.5 hours every day of GPU runtime

The configuration with a GPU significantly reduced training time, enabling faster experimentation and model iteration. Colab's integration with Google Drive made it easy to manage and share data. Additionally, its support for TensorFlow and Keras ensured smooth execution of our deep learning workflows.

## 2. Kaggle

Kaggle<sup>2</sup> is another cloud-based platform widely used for data science and machine learning tasks. It provides access to a wide range of datasets, competition environments, and cloud-based computational resources, including free access to GPUs.

For training our third model, we utilized Kaggle due to its straightforward setup and the ability to efficiently handle large datasets. Kaggle offers access to high-performance NVIDIA Tesla T4 GPUs, which are ideal for deep learning tasks such as image classification and object detection.

- **Kaggle Configuration:**

- **GPU:** NVIDIA Tesla T4 (15 GB VRAM \* 2, totaling 30 GB VRAM)
- **RAM:** 29 GB (maximum)
- **Disk Usage:** 57 GB (total storage available)
- **Free Usage:** 30 hours of GPU usage per week, with a maximum of 12 hours per session.

In Kaggle, the collaborative environment was beneficial as it allowed us to share notebooks with team members, which facilitated efficient collaboration and model refinement. Kaggle's support for popular libraries like TensorFlow and PyTorch further streamlined the model development process.

## 3. VS Code

Visual Studio Code (VS Code) is a lightweight, extensible code editor that we used as

---

2. <https://www.kaggle.com/>

our local development environment. While it doesn't have the same cloud resources as Colab or Kaggle, VS Code is highly flexible and suitable for developing and debugging code locally. It supports a wide range of programming languages, including Python, and can be easily extended with various plugins and extensions.

In this project, we used VS Code primarily for writing and testing scripts related to data preprocessing, model evaluation, and code related to visualizations and plotting. VS Code's integration with Python and Jupyter Notebooks allowed us to run smaller experiments locally before scaling them to cloud-based environments. It was also used for debugging and optimizing methods, especially for plotting performance metrics, confusion matrices, and training curves.

### 4.2.2 Programming Language: Python

The programming language used for developing our wildfire detection system was Python. Python is renowned for its simplicity, readability, and wide support in the machine learning and data science communities. Its extensive ecosystem of libraries and frameworks makes it an ideal choice for deep learning applications.

Python is particularly favored for its support of both high-level and low-level operations. This versatility allows researchers to prototype quickly while also providing access to more sophisticated techniques when needed. In this project, Python enabled seamless integration with libraries like TensorFlow, Keras, and OpenCV for model development and data manipulation. Its rich ecosystem of libraries and community support made Python the best choice for building and deploying machine learning models.

### 4.2.3 Key Libraries and Frameworks

To support the core functionalities of the system—including API development, deep learning, image processing, and interactive visualization, a selection of essential libraries and tools was used:

- **TensorFlow and Keras:** The main libraries used for building and training segmentation models such as U-Net and FCN. They offer modular APIs and support for GPU acceleration.

- **PyTorch**: Deep learning framework used for training and deploying object detection models, including integration with YOLOv8.
- **Ultralytics (YOLOv8)**: High-level API built on PyTorch for training, inference, and evaluation of YOLO-based object detection models.
- **TensorBoard (optional)**: Used for monitoring training and evaluation metrics via visual dashboards, integrated automatically with YOLOv8.
- **Matplotlib and Seaborn**: Visualization libraries used to plot training histories, confusion matrices, and performance metrics.
- **Pandas**: Used for handling and processing dataset metadata stored in CSV files, such as file paths and labels.
- **NumPy**: Core library for numerical operations, essential for matrix manipulations and preprocessing steps.
- **OpenCV**: Library for image processing and manipulation, commonly used for tasks such as resizing, masking, and augmentation.
- **Albumentations**: Fast and flexible library for applying image augmentations during training to improve model robustness.
- **Pillow (PIL)**: Used for basic image I/O and manipulation tasks.
- **Flask**: Lightweight web framework for building backend APIs.
- **SQLAlchemy**: ORM for database interaction and model definition.
- **Bootstrap**: Frontend framework for responsive UI design.
- **Leaflet**: JavaScript library for rendering interactive maps on the frontend.
- **Mapbox API**: Provides basemaps and geographic data used for spatial visualization.

## 4.3 Results and Experiments

For our wildfire detection system, we experimented with three deep learning models: U-Net, SegFormer b2, and YOLO v8. Each model was trained and evaluated on a carefully prepared dataset of satellite images.

To rigorously evaluate the ability of each model to detect wildfire-affected areas, we train experiments based on *Morphy masks*, which were used as ground-truth annotations for wildfire segmentation.



To optimize the model’s ability to detect fire in satellite images, we utilized a composition of three spectral channels: C7, C6, and C2. This combination was chosen to enhance fire visibility, as these channels allows the model to better capture fire-related features and help it better distinguish fire regions from the background. In this section, we present the results achieved by the models and discuss their performance in detail.

### 4.3.1 Model 1: U-Net

We begin by discussing the U-Net model, as it was one of the key architectures explored for the segmentation task.

- **Metrics Rapport**

The U-Net model was trained and evaluated on the test dataset (15%) using the following key performance metrics: F1 Score, Recall, Precision, and Intersection over Union (IoU).

These metrics were computed exclusively on the test dataset, ensuring the model’s performance was assessed on unseen data.

The results are summarized in Table 4.1.

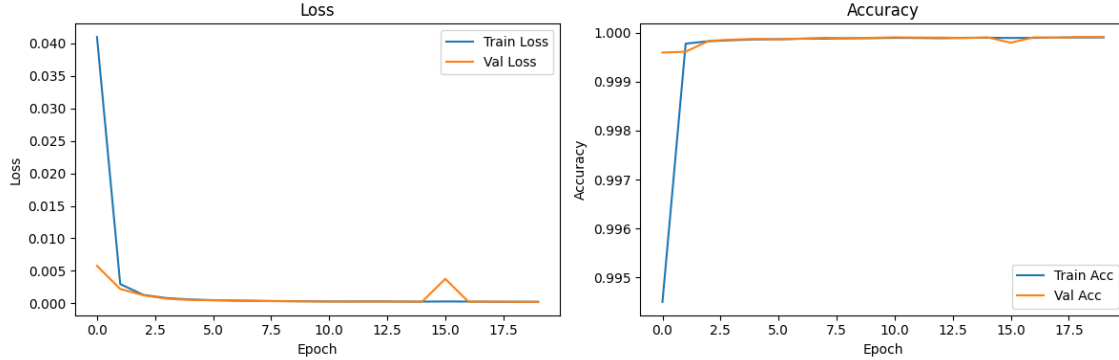
Metric	Value (%)
F1 Score	90.85
Recall	95.36
Precision	86.67
Intersection over Union (IoU)	83.22

**Table 4.1** – Performance Metrics of U-Net on the Test Dataset

These results indicate that the U-Net model demonstrates strong capability in detecting wildfire regions. The high recall (95%) shows that the model successfully identifies the majority of fire pixels, which is critical for minimizing missed detections in a wildfire monitoring system. The F1 score of 90% reflects a good balance between recall and precision, ensuring that the model not only detects fires but also keeps false alarms relatively low. The IoU of 83% further confirms that the segmentation quality is robust and reliable across diverse test images.

### • Accuracy and Loss

To further analyze the model’s learning behavior, we examined its accuracy and loss curves during training.



**Figure 4.1** – Training and Validation Accuracy and Loss Curves for U-Net

As shown in Figure 4.1, both training and validation loss drop sharply in the initial epochs and then converge to near-zero values, indicating that the model has learned effectively and generalizes well to unseen data. The only minor fluctuation observed in the validation loss suggests occasional variation but no consistent overfitting behavior.

Similarly, the training and validation accuracy curves reach approximately 99% within just a few epochs and remain consistently high. This indicates that the U-Net model demonstrates excellent capability in correctly identifying wildfire-affected regions with minimal errors. The overlap between training and validation curves confirms that the model maintains strong generalization without overfitting.

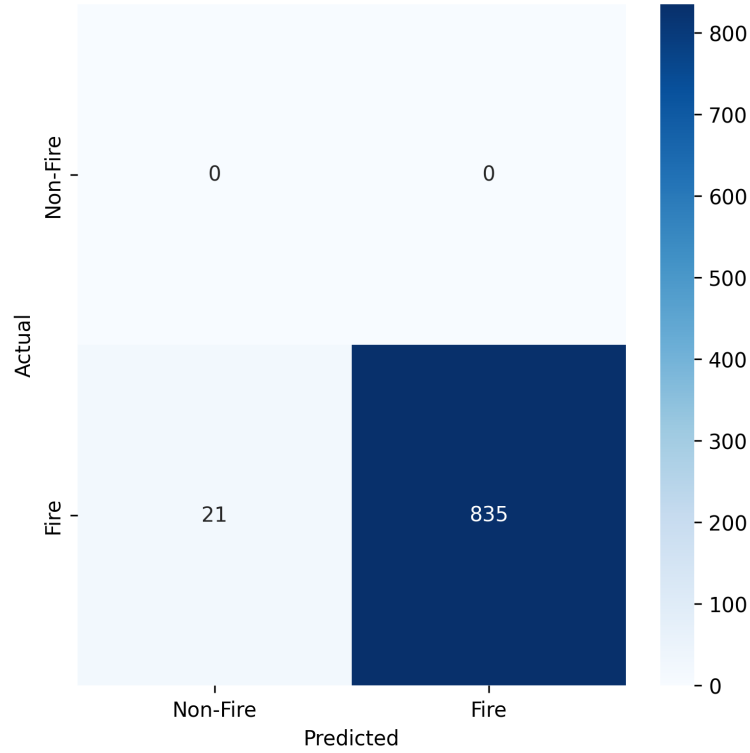
### • Confusion Matrix

In addition to pixel-level metrics, we evaluated the model’s performance at the image level using a specific strategy to compute the confusion matrix:

*If the model predicts 70% or more of the true fire pixels in an image as fire pixels, we consider the image to be classified as a “fire image.” Otherwise, it is treated as a “non-fire image.”*

U-Net Confusion Matrix in Figure 4.2 shows the following results:

- **True Positives (fire images correctly detected):** 835
- **True Negatives (non-fire images correctly detected):** 0



**Figure 4.2** – Confusion Matrix for U-Net on the Test Dataset

- **False Positives (non-fire images misclassified as fire images): 0**
- **False Negatives (fire images misclassified as non-fire images): 21**

**Interpretation:** U-Net demonstrates high sensitivity in detecting fire regions, with a large number of true positives and only a small number of false negatives. Within the context of fire-only imagery, U-Net performs effectively and reliably in detecting wildfire-affected areas.

### 4.3.2 Model 2: SegFormer B2

We now turn our attention to the SegFormer B2 model, a transformer-based architecture known for its high segmentation accuracy and computational efficiency.

#### • Metrics Rapport

SegFormer B2 was evaluated using the same test dataset and performance metrics: F1 Score, Recall, Precision, and Intersection over Union (IoU). These metrics provide a comprehensive view of the model's segmentation capability. The results are summarized in Table 4.2.

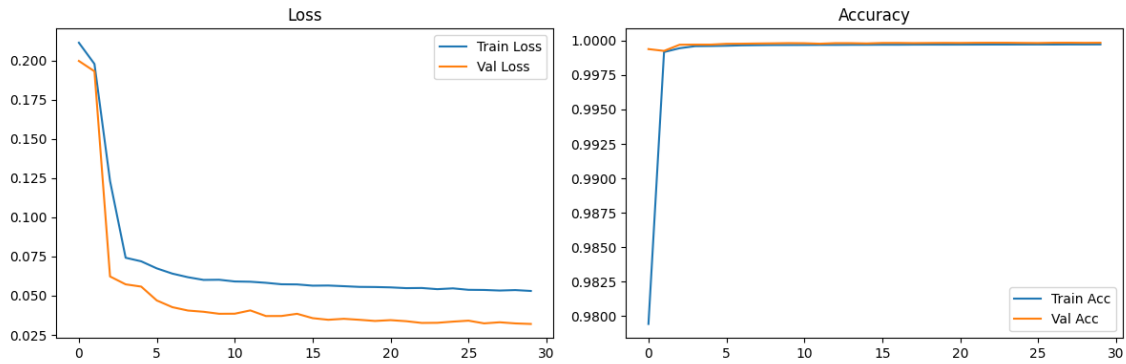
Metric	Value (%)
F1 Score	81.87
Recall	78.02
Precision	86.11
Intersection over Union (IoU)	69.30

**Table 4.2** – Performance Metrics of SegFormer B2 on the Test Dataset

While SegFormer B2 does not reach the same levels of precision and recall as U-Net, it still performs well. The Precision of 86.11% indicates that the model is effective at correctly identifying fire regions, though its lower Recall (78.02%) suggests that it misses some actual fire zones. The F1 Score of 81.87% shows a reasonable balance between these aspects. The IoU of 69.30% implies that the segmentation quality is acceptable, albeit not as refined as U-Net’s.

- **Accuracy and Loss**

To assess training dynamics, we analyzed the loss and accuracy curves during training and validation.



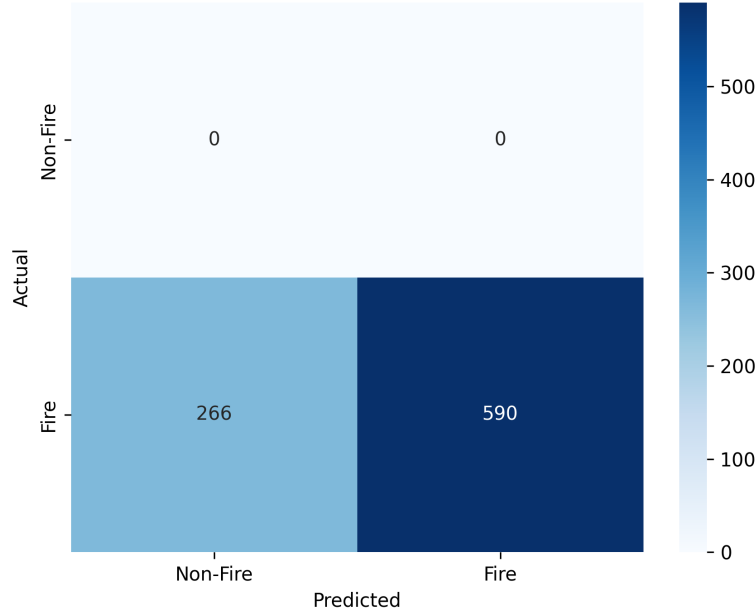
**Figure 4.3** – Training and Validation Accuracy and Loss Curves for SegFormer B2

As shown in Figure 4.3, the loss for both training and validation decreases steadily across epochs, suggesting stable learning. However, the loss plateaus at slightly higher values than those observed for U-Net, reflecting the moderately lower performance metrics.

The accuracy curves reach high levels ( 99.9%) and remain consistent throughout training and validation, indicating the model learns to generalize well without overfitting. Despite the strong accuracy, the gap in metrics like IoU suggests that SegFormer B2 may be more conservative in segmentation boundary precision compared to U-Net.

### • Confusion Matrix

As with U-Net, we applied an image-level evaluation strategy using the same thresholding rule: an image is classified as a “fire image” if the model detects at least 70% of the actual fire pixels.



**Figure 4.4** – Confusion Matrix for SegFormer B2 on the Test Dataset

The confusion matrix for SegFormer B2 yields the following results(Figure4.4):

- **True Positives (fire images correctly detected):** 590 images
- **True Negatives (non-fire images correctly detected):** 0 images
- **False Positives (non-fire images misclassified as fire images):** 0 images
- **False Negatives (fire images misclassified as non-fire images):** 266 images

**Interpretation:** SegFormer B2 exhibits decent performance at the image level. However, the number of false positives and false negatives is higher compared to U-Net. While still reliable, the model may occasionally miss fire events or trigger false alarms. Nonetheless, it remains a viable alternative, especially in resource-constrained or transformer-preferred deployment environments.

### 4.3.3 Model 3: YOLO v8

We now evaluate YOLO v8 developed by Ultralytics. Unlike pixel-wise segmentation models like U-Net or SegFormer, YOLO (You Only Look Once) approaches the problem from a

detection standpoint, identifying bounding boxes around fire regions rather than providing per-pixel classifications.

### • Metrics Rapport

The YOLO v8 model was evaluated using object detection metrics: Precision, Recall, mean Average Precision at IoU thresholds of 0.5 (mAP@0.5) and from 0.5 to 0.95 (mAP<sup>3</sup>@0.5:0.95), in addition to derived values such as the F1-score. The results are summarized in Table 4.3.

Metric	Value (%)
Precision	76
Recall	52
mAP@0.5	60
mAP@0.5:0.95	34
F1 Score (peak at optimal threshold)	58

**Table 4.3** – Performance Metrics of YOLO v8

Although YOLO v8 does not match the pixel-level granularity of SegFormer B2 or U-Net, it demonstrates reasonable potential in bounding box-level detection tasks. The F1-score peaks at 58% at a confidence threshold of 0.3, indicating a moderate trade-off between precision and recall. While the recall remains relatively low at 52%, the precision reaches 76%, suggesting that when the model detects fire, it is usually correct. The mAP metrics—60% at IoU=0.5 and 34% across IoU thresholds—reflect acceptable detection capability, particularly under more lenient overlap criteria.

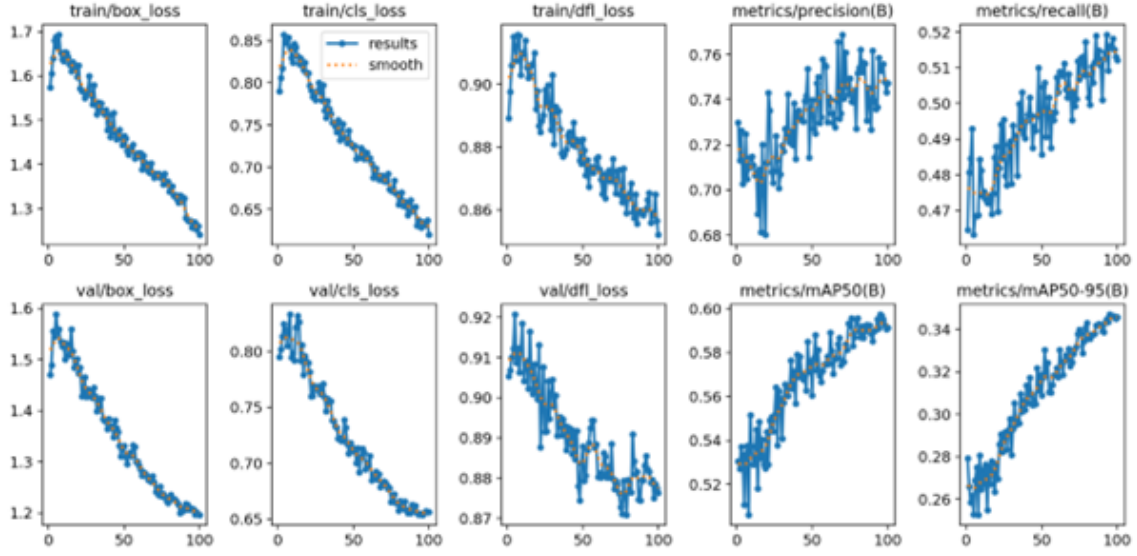
However, compared to U-Net and SegFormer B2, YOLO v8 achieves the lowest overall performance across all evaluated metrics, making it the least effective model for this wildfire detection task.

### • Accuracy and Loss

We tracked three core loss components during training and validation: box loss, classification loss, and Distribution Focal Loss (DFL), which are summed to compute the total loss.

---

3. Mean Average Precision (mAP) is a standard evaluation metric for object detection tasks. It combines precision and recall to measure both the accuracy of predicted object classes and the quality of their corresponding bounding boxes.



**Figure 4.5** – Training and Validation Loss Curves for YOLO v8

As shown in Figure 4.5, all components of the training and validation loss curves exhibit steady decreases:

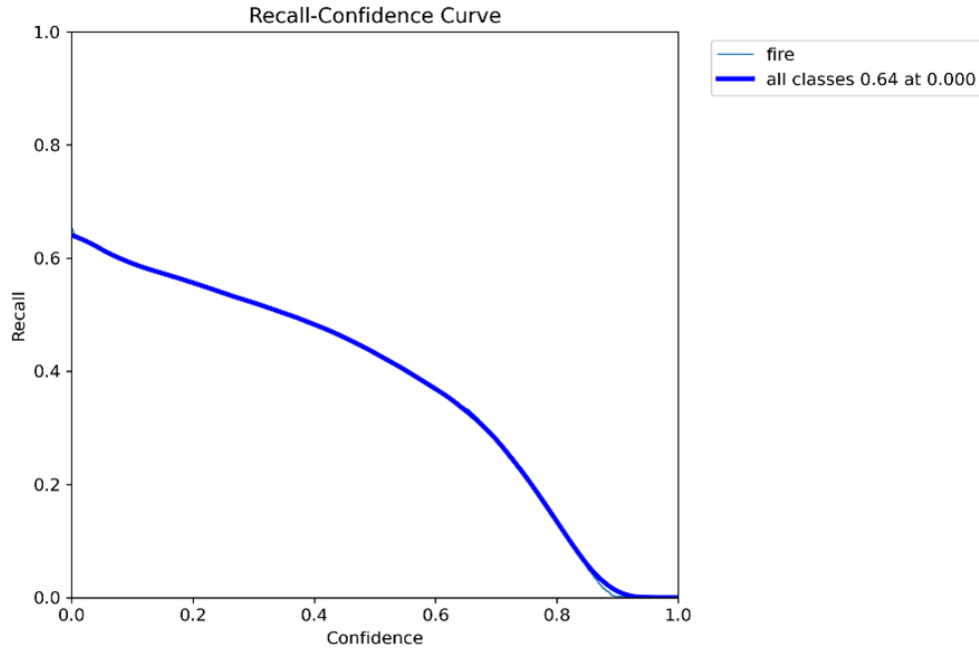
- **Box Loss:** drops from 1.7 to 1.3, indicating improved object localization.
- **Classification Loss:** decreases from 0.85 to 0.65, showing better label accuracy.
- **DFL Loss:** slightly reduces from 0.9 to 0.86, enhancing bounding box precision.

The total loss converges to approximately 2.81 by the end of training. Validation losses follow similar trends, showing no major signs of overfitting, suggesting good generalization on unseen data.

#### • Confidence Curves and Threshold Tuning

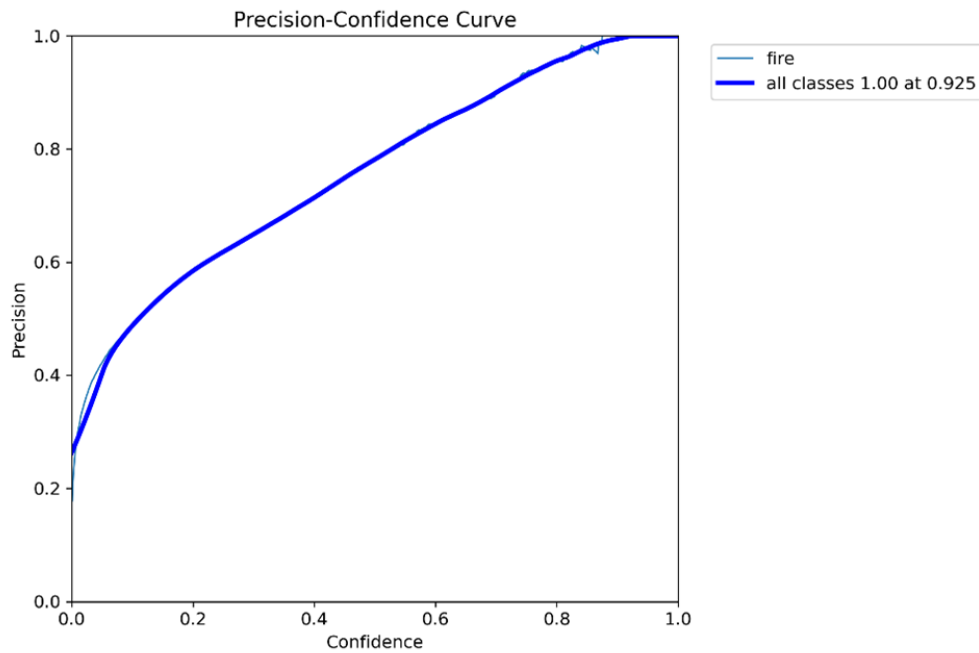
YOLO v8's performance varies based on the prediction confidence threshold. The Recall, Precision, and F1-score confidence curves provide deeper insight into model behavior:

- **Recall:** Starts high (64%) and declines as the model becomes more conservative. (see 4.6)



**Figure 4.6** – Recall, Precision, and F1-Score vs Confidence Threshold for YOLO v8

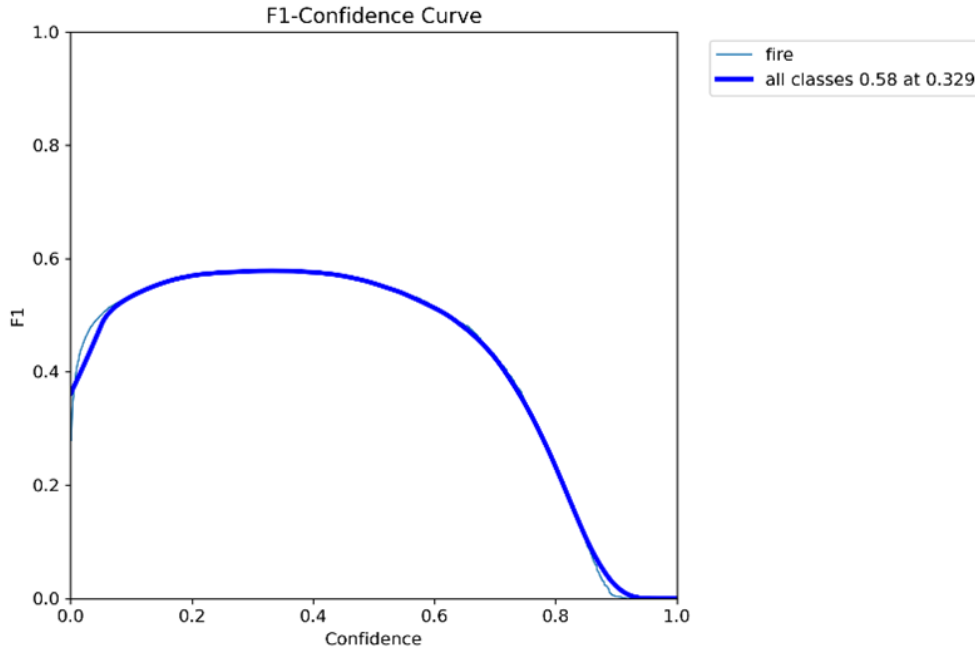
- **Precision:** Rises steadily with higher confidence, reaching 100% near 0.9.(see4.7)



**Figure 4.7** – Recall, Precision, and F1-Score vs Confidence Threshold for YOLO v8

- **F1-Score:** Peaks at 58% around a threshold of 0.3, offering an optimal balance.(see4.8)





**Figure 4.8** – Recall, Precision, and F1-Score vs Confidence Threshold for YOLO v8

This suggests threshold tuning is vital for deployment, depending on whether false positives or false negatives are more tolerable.

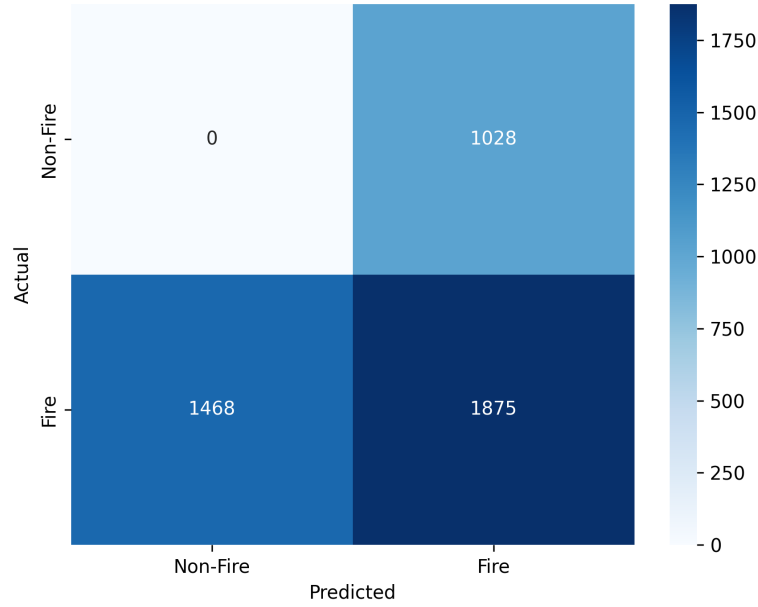
#### • Confusion Matrix

To evaluate the performance of **YOLOv8**, a strategy tailored to its object detection paradigm was adopted. Unlike segmentation models that use a pixel-wise Intersection-over-Union (IoU) threshold (70% in our case) between predicted and ground truth masks, object detection models such as YOLOv8 output *bounding boxes*.

In this evaluation, the confusion matrix represents performance at the **bounding box level**, not the image level. Each predicted bounding box labeled as fire was assessed for correctness:

- A predicted box was counted as a **True Positive (TP)** if it corresponded to an actual fire region.
- A predicted box was considered a **False Positive (FP)** if it incorrectly labeled a non-fire region as fire.
- A **False Negative (FN)** was recorded when a ground truth fire region was not detected by any predicted box.

The confusion matrix for YOLOv8 shows the following results(Figure4.9):



**Figure 4.9** – Confusion Matrix for YOLOv8 at the Bounding Box Level

- **True Positives (TP):** 1875 fire bounding boxes correctly detected.
- **False Positives (FP):** 1028 non-fire regions mistakenly detected as fire.
- **False Negatives (FN):** 1468 fire regions missed (not detected).
- **True Negatives (TN):** Not represented in this confusion matrix, as object detection models do not typically generate exhaustive negative bounding boxes for non-fire regions.

**Interpretation:** This evaluation shows that YOLOv8 has strong fire detection capability but also generates a notable number of false positives and false negatives. These results indicate a trade-off between sensitivity and precision that must be considered depending on the system’s operational context.

## 4.4 Discussion

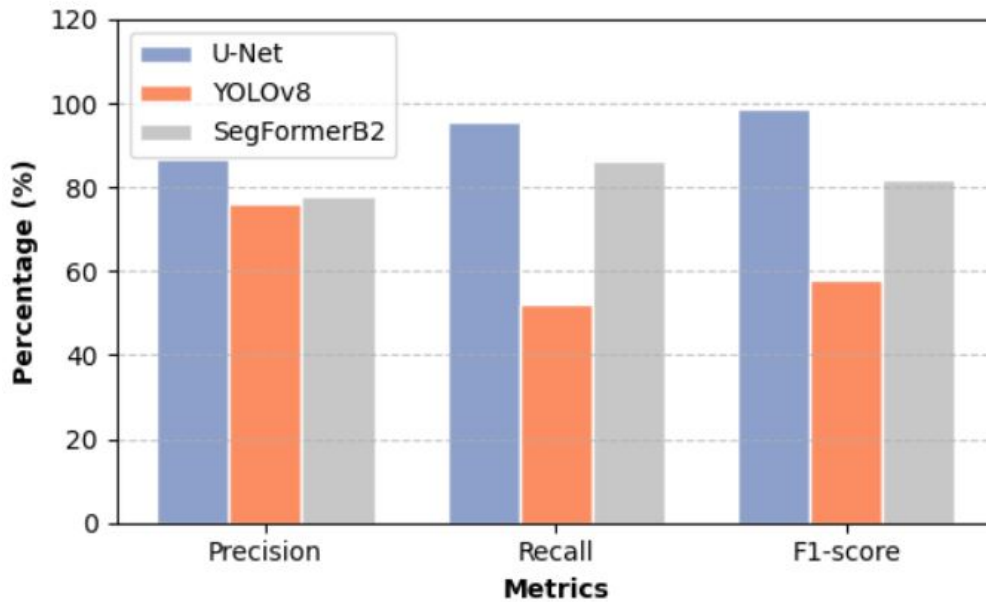
In this section, we perform a comparative analysis of the three models used in our wildfire detection system: **U-Net**, **SegFormer B2**, and **YOLOv8**.

This analysis provides insights into which model and configuration is most suitable for deployment in our wildfire detection system.

To better visualize the performance of the three models, Figure 4.10 presents a bar chart comparing the F-score, Recall, and Precision for each model.

Model	F1 Score (%)	Precision (%)	Recall (%)
U-Net	90	86	95
SegFormer B2	81.87	86.11	78.02
YOLO v8	58	76	52

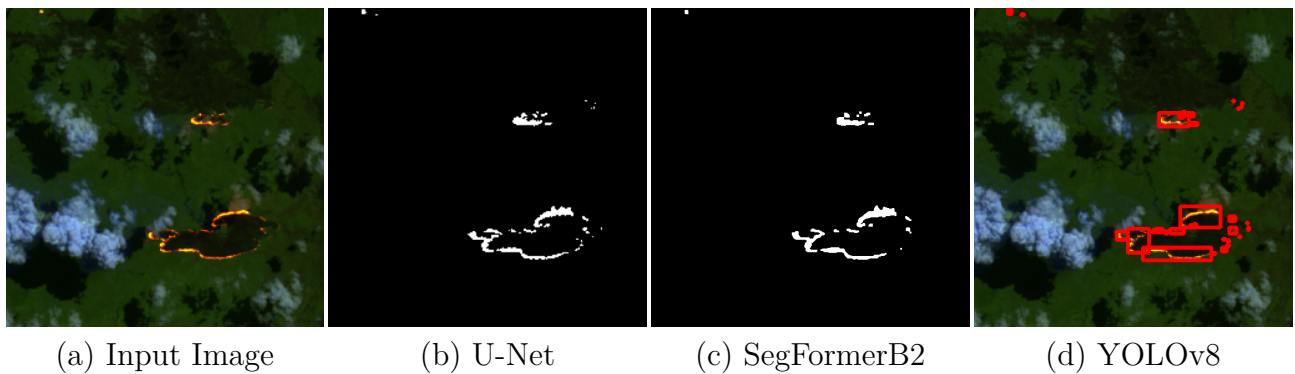
**Table 4.4** – Comparison of Key Performance Metrics Across Models



**Figure 4.10** – Comparative bar chart of the 3 model performance based on key metrics.

Figure 4.11 displays an example satellite image along with the predicted outputs generated by each of the three models: **U-Net**, **SegFormerB2**, and **YOLOv8**.

For the segmentation models (U-Net and SegFormerB2), the figure presents the predicted wildfire masks, while for YOLOv8, the image shows the detected wildfire regions outlined with bounding boxes.



**Figure 4.11** – Comparison of the detected masks from our models: U-Net, SegFormerB2, and YOLOv8.

From both the table and the bar chart, it is evident that the **U-Net** model outperforms the

others, achieving the highest F1 Score (**90%**) and Recall (**95%**). High Recall is particularly critical in wildfire detection, where failing to identify fire zones can have severe consequences.

While **SegFormer B2** demonstrates solid performance, with an F1 Score of 81.87%, a Precision of 86.11%, and a Recall of 78.02%, it still falls short of U-Net’s effectiveness, especially in terms of sensitivity and overall segmentation accuracy.

In contrast, **YOLOv8** showed the lowest performance across all key metrics, with an F1 Score of 58% and a Recall of 52%. Compared to the segmentation models, this indicates that object detection is currently less effective for identifying wildfire regions in satellite imagery. However, its inclusion in this study highlights the potential of object detection methods in this domain.

This study demonstrates that, while segmentation currently yields superior results, object detection remains a promising direction that can be improved through model enhancements, higher-quality data annotation, and task-specific optimizations tailored to wildfire detection.

Based on this comparative evaluation, **U-Net was selected as the final model** for deployment in our wildfire detection system, due to its superior ability to detect fire zones accurately and consistently.

## • Model Comparison with Related Works

In this section, we compare the performance of our models with those from previous studies to highlight the effectiveness of our approach in wildfire detection. The comparison is based on key performance metrics such as Precision, Recall, and F1 Score, which provide insights into the strengths and weaknesses of each model. Our models include U-Net, SegFormer B2, and YOLO v8, trained on satellite imagery from the African region, specifically targeting wildfire detection. We also compare these results to those reported in related work, including Boumaiza and Brahimi (2024), James et al. (2023), and Choutri et al. (2023).

Table 4.5 provides a detailed comparison of these results.

Name of Model	Precision (%)	Recall (%)	F1 Score (%)	Source
<b>Our Models</b>				
<b>U-Net</b>	86	95	90	1 <sup>st</sup> Model
<b>SegFormer B2</b>	78	86	81.87	2 <sup>nd</sup> Model
<b>YOLO v8</b>	76	52	58	3 <sup>rd</sup> Model
<b>Related Works</b>				
<b>U-Net (2024)</b>	95	74	83	Boumaiza & Brahim
<b>MobileNetV2</b>	95	/	95	James et al.
<b>YOLO-NAS</b>	71	/	68	Choutri et al.

**Table 4.5** – Comparison of Model Performance Metrics Between Our Models and Related Works

U-Net (Our Model) achieved the highest F1 score (90%), with a solid precision of 86% and a high recall of 95%. This shows that U-Net is well-suited for wildfire detection, especially when trained on satellite imagery, as it effectively balances precision and recall. Compared to other works, like Boumaiza and Brahim (2024), whose U-Net model achieved an F1 score of 83%, our U-Net model performs better in terms of both recall and overall detection accuracy.

SegFormer B2 (Our Model) performed moderately, with an F1 score of 81.87%. While it didn't outperform U-Net, SegFormer showed decent performance, indicating that more complex architectures may still be viable but are less efficient for this specific task compared to U-Net.

YOLO v8 (Our Model) struggled significantly, with an F1 score of just 58%. This is notably lower than the performance seen in related work using YOLO, such as Choutri et al. (2023), where YOLO-NAS achieved an F1 score of 68%. The main reason for YOLO's poor performance in our study could be the type of data used; YOLO is generally better suited for object detection tasks but may not be ideal for pixel-wise fire detection, especially in satellite imagery, where fine segmentation is required.

James et al. (2023) achieved a remarkable F1 score of 95% using MobileNetV2, even with limited data. Their lightweight CNN model demonstrated that transfer learning can be highly effective for wildfire detection with smaller datasets, showing that classification tasks can sometimes outperform more complex models, especially when the data is appropriately prepared for such architectures.

Overall, the performance of U-Net in our study stands out as the most reliable for wildfire detection in satellite images, confirming that segmentation-based models are better suited for

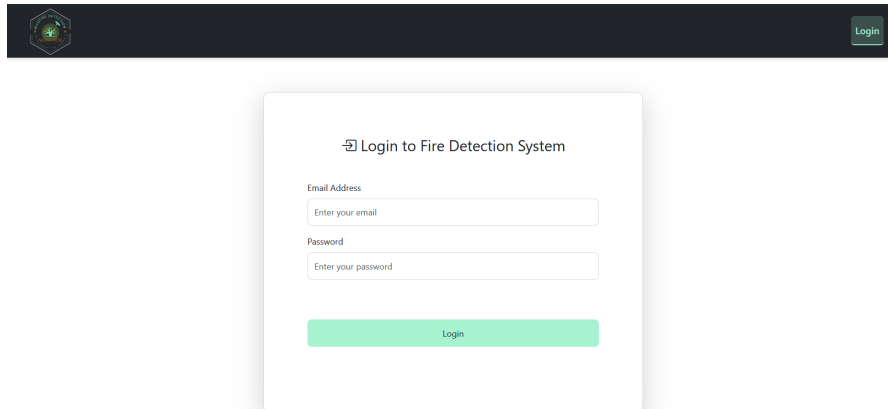
the task. The YOLO v8 model, while effective for object detection, did not perform well in our case due to the pixel-level segmentation required for accurate fire detection. Additionally, MobileNetV2's high performance with limited data highlights the power of transfer learning, which could be a potential area for further exploration in future studies.

## 4.5 Web Application

The Fire Detection System is a web-based application designed to detect fires with satellite imagery. It integrates Flask for the backend, SQLAlchemy for database management, and Bootstrap for a responsive and visually appealing frontend. It supports two user roles: Admin and satations.

The application features a comprehensive set designed pages, detailed in the following sections:

- **Login Page:** Both the admin and fire station can log in to the system with credentials (email, password) to access their corresponding interfaces as shown in Figure 4.12.



**Figure 4.12** – login page

- **Dashboard Page:** As shown in Figure 4.13, the main page for administrators is designed for use in Algeria. Our system currently targets Algeria, with plans to expand this approach to all other wilayas. Administrators must first select a wilaya and then a zone (a specific forest within the same wilaya). Each administrator is responsible for a designated zone. This page enables them to analyze the map to detect existing fires by entering coordinates (start latitude, start longitude, end latitude, and end longitude) along with a zoom level, or by directly searching for a place by name. Users can then view analytical results with relevant information.

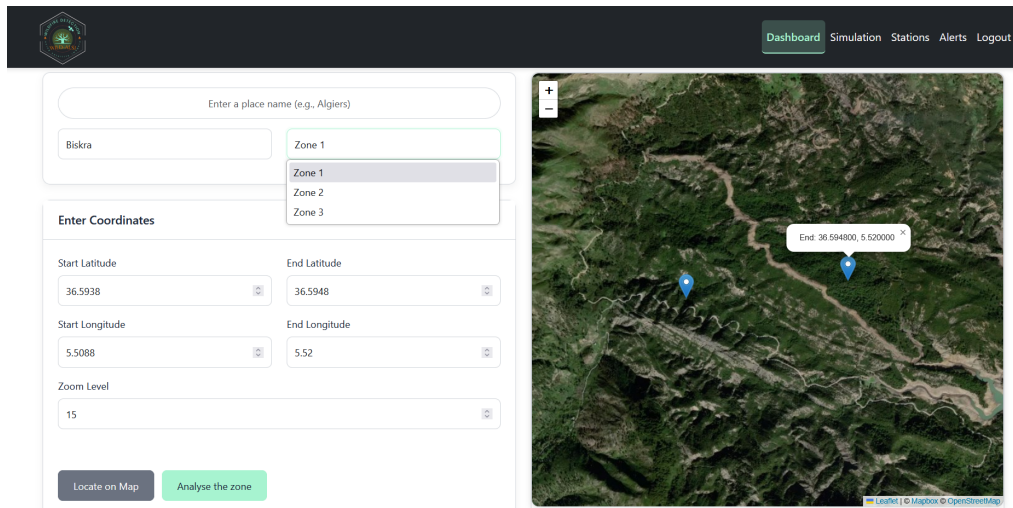


Figure 4.13 – dashboard page

Also, it consists of 2 buttons:

- **Locate on Map button:** updates the map to show the entered coordinate range, and clicking table rows zooms to specific coordinates.
- **Analyze map button:** Fetches satellite images from Mapbox for a grid of coordinates, processes them with our model, and figure 4.14 shows results. Alerts are created in case of fire detections and assigned to stations within the coordinate range.

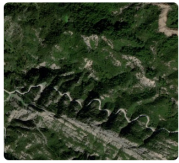


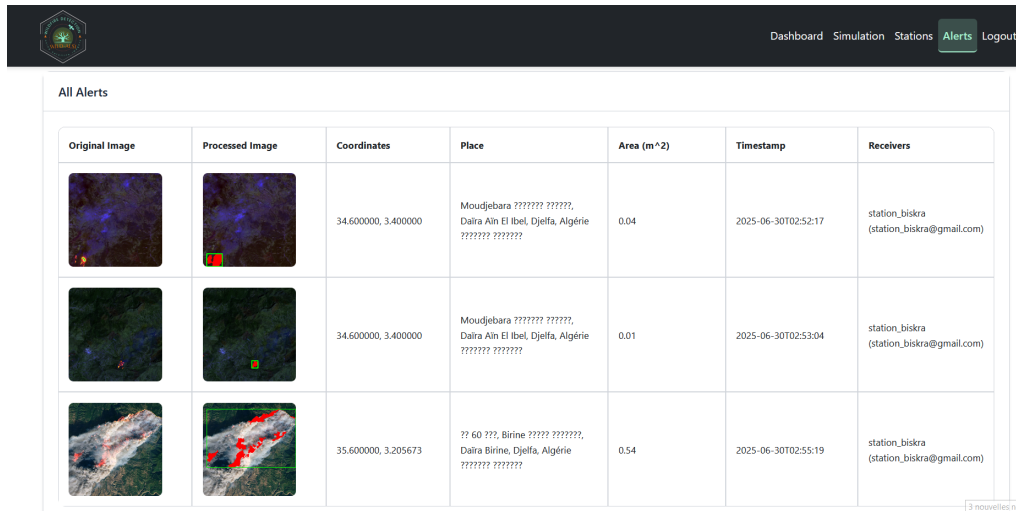
Fire Points	Coordinates	Place	Area (m2)	Timestamp
	36.593800,5.508800	Lat: 36.593800, Lon: 5.508800	0.0	
	36.593800,5.514400	Lat: 36.593800, Lon: 5.514400	0.0	
				

Figure 4.14 – analyse map results

- **Alerts page:** Figure 4.15 displays all alerts (for admins) or assigned alerts (for stations). It consists of a table showing image thumbnail, coordinates, place (geographic name), fire area, timestamp (when the alert was generated), and stations (admins only can see

a list of assigned stations (username and email)).

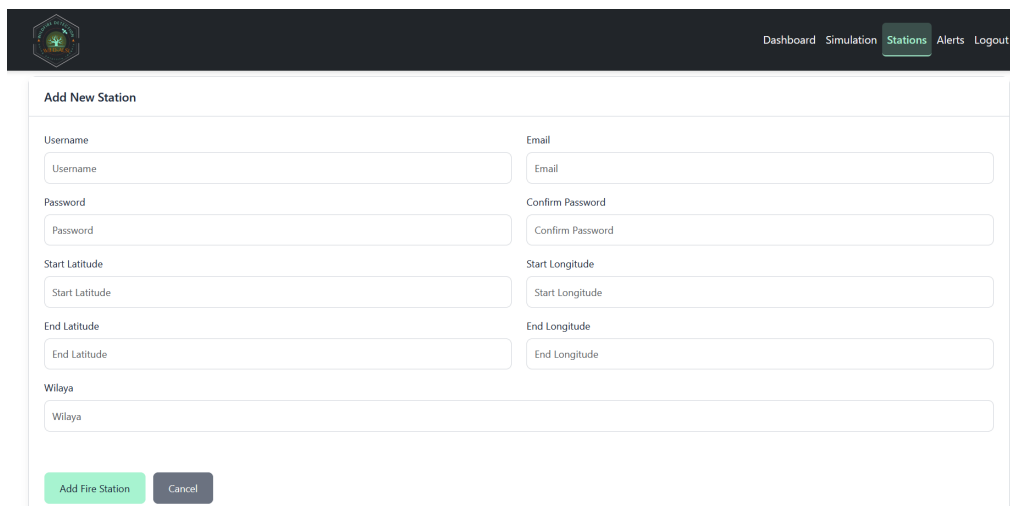
When a fire detection occurs, an alert is immediately sent to the relevant fire stations, where new alerts appear in a distinct color accompanied by a warning sound, enabling the station to respond to the fire and take the necessary actions.



Original Image	Processed Image	Coordinates	Place	Area (m <sup>2</sup> )	Timestamp	Receivers
		34.600000, 3.400000	Moudjebara ?????? ??????, Daira Ain El Ibel, Djelfa, Algérie ?????? ??????	0.04	2025-06-30T02:52:17	station_biskra (station_biskra@gmail.com)
		34.600000, 3.400000	Moudjebara ?????? ??????, Daira Ain El Ibel, Djelfa, Algérie ?????? ??????	0.01	2025-06-30T02:53:04	station_biskra (station_biskra@gmail.com)
		35.600000, 3.205673	?? 60 ???, Birine ????? ??????, Daira Birine, Djelfa, Algérie ?????? ??????	0.54	2025-06-30T02:55:19	station_biskra (station_biskra@gmail.com)

Figure 4.15 – alerts list page

- **Stations page:** As shown in Figure 4.16, the admin can add new stations by entering necessary information like username, email, password, coordinates, and wilaya, also can manage existing station shown in figure 4.17 by editing their information or deleting them.



**Add New Station**

Username

Email

Password

Confirm Password

Start Latitude

Start Longitude

End Latitude

End Longitude

Wilaya

Figure 4.16 – add station



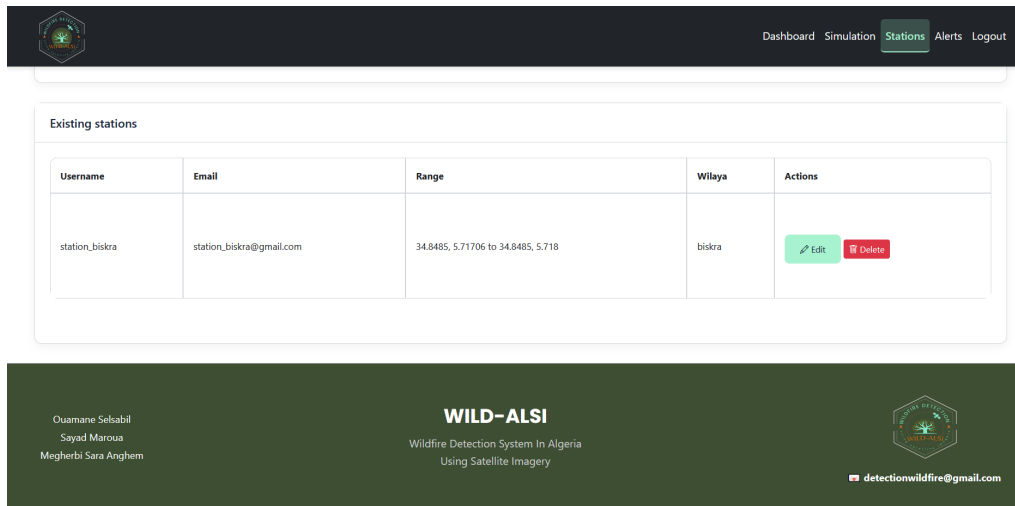


Figure 4.17 – existant station

- **Simulation page:** Allows admins to upload images for fire detection, process them, and split them into sub-images as shown in 4.18.

If a sub-image have fire, we send the alert with their information (coordinates, place, area affected by fire, timestamp, receivers) as shown in Figure 4.20 , and we send it to the corresponding receiver based on the coordinates of the fire, if a sub-image doesn't contain fire nothing happens as shown in the Figure 4.19.

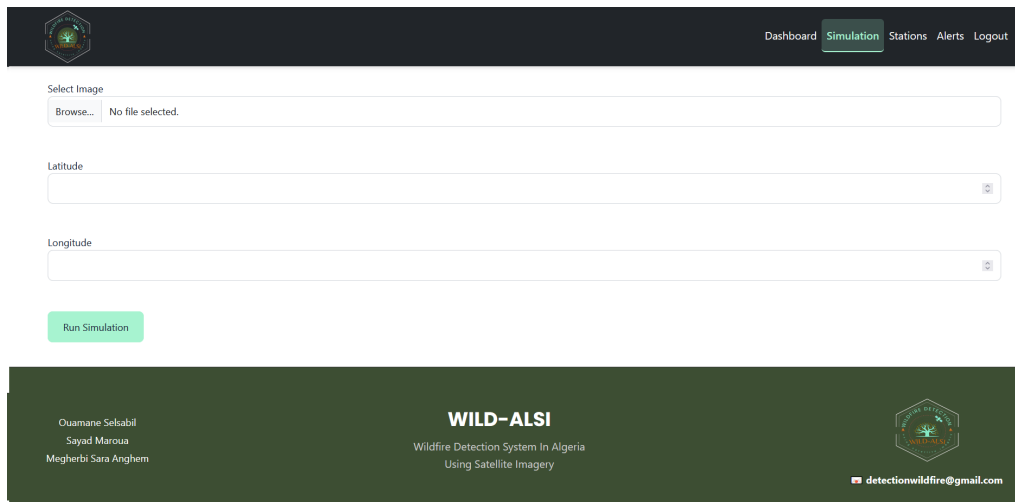


Figure 4.18 – Simulation page


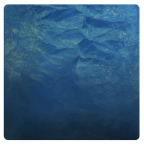


<div>  <div>Dashboard <b>Simulation</b> Stations Alerts Logout</div> </div>						
Title	Fire Points	Coordinates	Place	Area (hectares)	Timestamp	Receivers
1		36.100000, 5.200000	Autoroute Est-Ouest, Mezouag <span>مزاوق</span> , Daira <span>دائرة</span> Ain Arnat, Sétif, Algérie <span>الجزائر</span>	0.0		None
2		36.100000, 5.202854	Autoroute Est-Ouest, Mezouag <span>مزاوق</span> , Daira <span>دائرة</span> Ain Arnat, Sétif, Algérie <span>الجزائر</span>	0.0		None
						

Figure 4.19 – sub-images when no fire detected




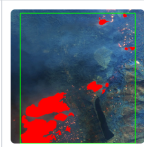
<div>  <div>Dashboard <b>Simulation</b> Stations Alerts Logout</div> </div>						
6		36.098616, 5.200000	Ain Arnat, Sétif, Algérie <span>الجزائر</span>	0.01	2025-06-30T04:16:12.818853	None
7		36.098616, 5.202854	Autoroute Est-Ouest, Mezouag <span>مزاوق</span> , Daira <span>دائرة</span> Ain Arnat, Sétif, Algérie <span>الجزائر</span>	0.8	2025-06-30T04:16:17.005526	None
8		36.098616, 5.205709	Autoroute Est-Ouest, Mezouag <span>مزاوق</span> , Daira <span>دائرة</span> Ain Arnat, Sétif, Algérie <span>الجزائر</span>	0.85	2025-06-30T04:16:22.382204	None

Figure 4.20 – sub-images when fire is detected

## 4.6 Conclusion

In this chapter, we presented the development and evaluation of our wildfire detection system. Based on the evaluation results, we selected U-Net as the final model due to its superior performance over the other models. After choosing U-Net, we conducted a comparative analysis with related work to highlight the strengths and limitations of our approach. We then integrated the selected model into a web interface using the Flask framework, which enabled easy access to the model for wildfire monitoring. Finally, we explored the system's interface, which allows users to interact with the model and efficiently visualize wildfire detection results.



# Conclusion and perspective

Wildfires are among the most destructive natural disasters, causing massive damage to ecosystems, agriculture, property, and even human lives. In recent years, their frequency and intensity have increased, making early detection and rapid response more critical than ever, especially in vulnerable regions like Algeria. Traditional wildfire monitoring methods in the country often rely on manual observation, which is limited in speed and accuracy.

To address this challenge, the objective of our project was to develop a wildfire detection system based on satellite imagery and deep learning techniques. We focused on integrating three advanced models, U-Net, SegFormer B2, and YOLOv8, to analyze satellite images and detect active fire regions automatically.

Among the three models tested, U-Net and SegFormer produced the most promising results. Their ability to perform semantic segmentation allowed us to accurately localize wildfire areas in satellite images. These results demonstrate the strong potential of segmentation-based deep learning models for wildfire detection tasks and have played a key role in shaping the foundation of our integrated detection and monitoring application.

On the other hand, YOLOv8, which is primarily an object detection model, did not perform well in our specific context. It struggled to effectively detect fires due to the diffuse, irregular, and often small-scale nature of wildfires in satellite imagery. While YOLO is a powerful tool in many vision tasks, it may not be the best choice for detecting fire areas from satellite images where boundary precision is critical.

The integration of our most successful models into a functional system is a significant step toward real-world deployment. Our web-based application is capable of analyzing satellite images, detecting fires, and supporting early warning mechanisms that could assist fire response teams in Algeria.

## Future Work

- **Evaluation of Alternative Models**

Future work will explore other advanced deep learning models or hybrid approaches to improve wildfire detection accuracy under varying environmental conditions.

- **Utilizing Algerian Satellite Data**

Incorporating imagery from *AlSat* (Algeria's satellite program) will enhance detection accuracy

with more localized, high-resolution data, supporting real-time monitoring.

- **Incorporating Additional Factors**

It would be beneficial to consider additional factors beyond just satellite imagery. These factors could include weather conditions (temperature, humidity, wind), topography, and vegetation types to improve detection reliability and fire behavior analysis.

- **Fire Spread Prediction and Coverage Analysis**

Currently, our system focuses on detecting the presence of fires, but it would be valuable to predict how wildfires will spread over time based on various factors. By incorporating fire spread models, such as Cellular Automata or LSTM-based approaches, we could predict the potential coverage area and speed of fire spread.

- **Operational Integration**

Collaboration with Algerian civil protection authorities will enable real-world testing and integration of the system into national wildfire monitoring and emergency response frameworks.

In conclusion, this project demonstrates how combining satellite remote sensing with deep learning can offer efficient, and accurate solution for wildfire detection. The results are encouraging and show real promise for practical deployment in Algeria and beyond.

# Bibliography

- [1] James B. Campbell and Randolph H. Wynne. *Introduction to Remote Sensing*. 5th. New York: Guilford Press, 2011. ISBN: 9781609181765. URL: <https://www.guilford.com/p/campbell12>.
- [2] Canada Centre for Remote Sensing. *Remote Sensing Tutorial*. <https://www.nrcan.gc.ca/maps-tools-and-publications/satellite-imagery-and-air-photos/remote-sensing-tutorials/9309>. Accessed: 2025-05-09. Natural Resources Canada, 2011.
- [3] Testbook. *Remote Sensing: Definition, Components, Types, Principles and Uses*. <https://testbook.com/physics/remote-sensing>. Accessed: 2025-05-09. Testbook Edu Solutions Pvt. Ltd., 2023. URL: <https://testbook.com/physics/remote-sensing> (visited on 05/09/2025).
- [4] V. Shivakumar, Samir Pokhrel, and Anant Parekh. *Block-1 Introduction to Remote Sensing*. IGNOU, 2017. ISBN: 9788126659661. URL: <http://hdl.handle.net/123456789/39605>.
- [5] Emilio Chuvieco and Alfredo Huete. *Fundamentals of Satellite Remote Sensing*. Taylor & Francis, 2009. ISBN: 9780415310842.
- [6] Professional Aerial Photographers Association. *A Brief History of Aerial Photography*. [https://papa.clubexpress.com/content.aspx?page\\_id=22&club\\_id=808138&module\\_id=158950](https://papa.clubexpress.com/content.aspx?page_id=22&club_id=808138&module_id=158950). Accessed: 2025-05-29. Professional Aerial Photographers Association, n.d. URL: [https://papa.clubexpress.com/content.aspx?page\\_id=22&club\\_id=808138&module\\_id=158950](https://papa.clubexpress.com/content.aspx?page_id=22&club_id=808138&module_id=158950).
- [7] Shubham Yadav. *History of Remote Sensing*. Accessed: 2025-05-09. 2023. URL: <https://geographicbook.com/history-of-remote-sensing/>.
- [8] European Space Agency. *ERS-1 (European Remote-Sensing Satellite-1)*. Image Credit: ESA. Accessed: 2025-05-09. eoPortal. 2023. URL: <https://www.eoportal.org/satellite-missions/ers-1>.
- [9] Wim H. Bakker et al. *Principles of Remote Sensing: An Introductory Textbook*. Ed. by Lucas L. F. Janssen and Gerrit C. Huurneman. Enschede, The Netherlands: International Institute for Geo-Information Science and Earth Observation (ITC), 2001.

- [10] M. Herold, M.E. Gardner, and D.A. Roberts. “Spectral resolution requirements for mapping urban areas”. In: *IEEE Transactions on Geoscience and Remote Sensing* 41.9 (2003), pp. 1907–1919. DOI: [10.1109/TGRS.2003.815238](https://doi.org/10.1109/TGRS.2003.815238).
- [11] Almi Ramadhi. *Satellite Data Sensor: Passive Sensors Vs Active Sensors*. Accessed: 2025-05-13. 2021. URL: <https://medium.com/@almiramadhi/satellite-data-sensor-d0b0d0db7a8c>.
- [12] Bachir Kaddar. *Basics Principles of Remote Sensing and GIS*. Course manual intended for Master students in Ecosysteme Steppique et Saharien, Agro-écologie, Biodiversité et écologie végétale, and Agriculture de précision. Algeria, 2022.
- [13] Geoimage. *The Advantages of Satellite Imagery Compared to Other Alternatives*. Accessed: 2025-05-10. 2023. URL: <https://geoimage.com.au/blog/advantages-satellite-imagery-compared-other-alternatives> (visited on 05/10/2025).
- [14] Satpalda. *What is Remote Sensing and How Does it Work?* Accessed: 2025-05-10. 2023. URL: <https://satpalda.com/what-is-remote-sensing-and-how-does-it-work/> (visited on 05/10/2025).
- [15] Spatial Post. *Active vs. Passive Sensors in Remote Sensing*. 2023. URL: <https://www.spatialpost.com/active-vs-passive-sensors-remote-sensing>.
- [16] Satpalda. *What is Aerial Remote Sensing?* Accessed: [Insert Last Accessed Date]. n.d. URL: <https://satpalda.co/what-is-aerial-remote-sensing/>.
- [17] Altigator. *Drones, UAV, UAS, RPA ou RPAS ? Quelle est la différence ?* Accessed: 2025-05-09. URL: <https://altigator.com/fr/drones-uav-uas-rpa-ou-rpas/>.
- [18] AAU Geodesy. *Ground-Based Sensing Platforms*. Accessed: 2025-05-09. URL: <https://aaugeodesy.com/ground-based-sensing-platforms/>.
- [19] Limko. *Le capteur MODIS (Moderate Resolution Imaging Spectroradiometer)*. Accessed: 2025-05-09. Oct. 2018. URL: <https://www.limko.cm/gis-geomatics/2018/10/le-capteur-modis-moderate-resolution-imaging-spectroradiometer/>.
- [20] *What is Satellite Imagery?* <https://www.geoimage.com.au/blog/what-satellite-imagery>. Accessed: 2025-05-01. Oct. 2021.
- [21] NOAA. *Why Don't Satellites Fall Out of the Sky?* <https://www.nesdis.noaa.gov/news/why-dont-satellites-fall-out-of-the-sky>. Accessed: 2025-06-01. 2023.
- [22] European Space Agency (ESA). *Types of Orbits*. Online; accessed June 1, 2025. European Space Agency. 2023. URL: [https://www.esa.int/Enabling\\_Support/Space\\_Transportation/Types\\_of\\_orbits](https://www.esa.int/Enabling_Support/Space_Transportation/Types_of_orbits) (visited on 06/01/2025).
- [23] SOAR Earth. *From Satellite Data to Maps*. Blog post about processing satellite imagery into maps. SOAR Earth. 2023. URL: <https://about.soar.earth/blog-pages/from-satellite-data-to-maps> (visited on 06/02/2024).

- [24] Thierry Toutin. “Review article: Geometric processing of remote sensing images: models, algorithms and methods”. In: *International Journal of Remote Sensing* 25.10 (2004). Comprehensive review of geometric correction techniques for satellite imagery., pp. 1893–1924. DOI: [10.1080/0143116031000101611](https://doi.org/10.1080/0143116031000101611). URL: <https://www.sciencedirect.com/science/article/abs/pii/S0924271603000583>.
- [25] U.S. Geological Survey. *Landsat Missions*. Accessed: 2023-10-15. 2023. URL: <https://www.usgs.gov/landsat-missions>.
- [26] National Aeronautics and Space Administration (NASA). *MODIS Website*. Accessed: 2023-10-15. 2023. URL: <https://modis.gsfc.nasa.gov/>.
- [27] Peng Liu et al. “Burned area detection and mapping using time series Sentinel-2 multispectral images”. In: *Remote Sensing of Environment* 296 (2023), p. 113753. ISSN: 0034-4257. DOI: <https://doi.org/10.1016/j.rse.2023.113753>. URL: <https://www.sciencedirect.com/science/article/pii/S0034425723003048>.
- [28] European Space Agency (ESA). *Sentinel-1 Mission*. Accessed: 2023-10-15. 2023. URL: <https://sentiwiki.copernicus.eu/web/s1-mission>.
- [29] Canadian Space Agency (CSA). *RADARSAT-2*. Accessed: 2023-10-15. 2023. URL: <https://www.asc-csa.gc.ca/eng/satellites/radarsat2/>.
- [30] Japan Aerospace Exploration Agency (JAXA). *ALOS-2 Satellite*. Accessed: 2023-10-15. 2023. URL: <https://global.jaxa.jp/projects/sat/alos2/>.
- [31] U.S. Geological Survey. *Landsat-8*. Accessed: 2023-10-15. 2023. URL: <https://www.usgs.gov/landsat-missions/landsat-8>.
- [32] National Oceanic and Atmospheric Administration. *VIIRS (Visible Infrared Imaging Radiometer Suite)*. Accessed: 2023-10-15. 2023. URL: <https://www.star.nesdis.noaa.gov/jpss/VIIRS.php>.
- [33] Japan Aerospace Exploration Agency (JAXA). *Global Change Observation Mission - Water "SHIZUKU" (GCOM-W)*. Accessed: 2023-10-15. 2023. URL: [https://global.jaxa.jp/projects/sat/gcom\\_w/](https://global.jaxa.jp/projects/sat/gcom_w/).
- [34] National Aeronautics and Space Administration (NASA). *Global Precipitation Measurement (GPM) Mission*. Accessed: 2023-10-15. 2023. URL: <https://gpm.nasa.gov/>.
- [35] National Aeronautics and Space Administration (NASA). *Soil Moisture Active Passive (SMAP) Mission*. Accessed: 2023-10-15. 2023. URL: <https://smap.jpl.nasa.gov/>.
- [36] European Space Imaging. *WorldView-3 Satellite Imagery*. Commercial satellite specifications and applications. European Space Imaging, 2024. URL: <https://www.euspaceimaging.com/about/worldview-3/> (visited on 07/10/2024).

- [37] Swift Geospatial. “The Basics of Spectral Bands: NIR, SWIR, and RGB”. In: *Swift Geospatial Solutions* (Mar. 2025). URL: <https://swiftgeospatial.solutions/2025/03/12/the-basics-of-spectral-bands-nir-swir-and-rgb/>.
- [38] constellr. “constellr Captures Tokyo’s Thermal Signature in Historic First Satellite Image”. In: *constellr* (Mar. 2025). Press release. URL: <https://www.constellr.com/article/constellr-captures-tokyos-thermal-signature-in-historic-first-satellite-image>.
- [39] World Economic Forum. “How Hyperspectral Satellite Imaging Can Help Fight Climate Change”. In: *World Economic Forum* (May 2024). Accessed:05-05-2025. URL: <https://www.weforum.org/stories/2024/05/hyperspectral-satellite-imaging-climate-change/>.
- [40] NASA Earthdata. *Synthetic Aperture Radar (SAR) Basics*. Accessed: July 10, 2024. National Aeronautics and Space Administration (NASA), 2023. URL: <https://www.earthdata.nasa.gov/learn/earth-observation-data-basics/sar> (visited on 07/10/2024).
- [41] Sidike Paheding et al. “Advancing horizons in remote sensing: a comprehensive survey of deep learning models and applications in image classification and beyond”. In: *Neural Computing and Applications* 36.27 (2024), pp. 16727–16767.
- [42] Catherine Nakalembe et al. “A review of satellite-based global agricultural monitoring systems available for Africa”. In: *Global Food Security* 29 (2021), p. 100543.
- [43] Hafez Ahmad. “Applications of remote sensing in oceanographic research”. In: *Int. J. Oceanogr. Aquac* 3 (2019), p. 000159. DOI: [10.23880/ijoac-16000159](https://doi.org/10.23880/ijoac-16000159).
- [44] Jun Chen. “Improving Urban Planning by Integrated Utilization of Remote Sensing and GIS’s”. In: *International Archives of Photogrammetry and Remote Sensing* 29 (1993), pp. 598–598.
- [45] *Climate Change Indicators: Wildfires*. Accessed: 2025-05-10. 2024. URL: <https://www.epa.gov/climate-indicators/climate-change-indicators-wildfires#ref17>.
- [46] Zach Taras. *Largest Wildfire in World History*. Accessed: 2025-05-10. Mar. 2025. URL: <https://science.howstuffworks.com/nature/natural-disasters/largest-wildfire-in-world-history.htm>.
- [47] Paul Przyborski. *Global Temperatures - World of Change*. Accessed: 2025-05-10. URL: <https://earthobservatory.nasa.gov/world-of-change/global-temperatures>.



- [48] Ouahiba Meddour-Sahar et al. “Wildfire Management Policies in Algeria: Present and Future Needs”. In: *Proceedings of the Fourth International Symposium on Fire Economics, Planning, and Policy: Climate Change and Wildfires*. Ed. by Armando González-Cabán. General Technical Report PSW-GTR-245. U.S. Department of Agriculture, Forest Service, Pacific Southwest Research Station, 2013, pp. 382–395. URL: [https://www.fs.usda.gov/psw/publications/documents/psw\\_gtr245/psw\\_gtr245\\_382.pdf](https://www.fs.usda.gov/psw/publications/documents/psw_gtr245/psw_gtr245_382.pdf).
- [49] Global Wildfire Information System (GWIS), European Commission, Joint Research Centre. *Country profile charts for Algeria*. <https://gwis.jrc.ec.europa.eu/apps/country.profile/charts/ba>. Accessed: 2025-05-13.
- [50] Global Wildfire Information System (GWIS), European Commission, Joint Research Centre. *Country profile overview for Algeria*. [https://gwis.jrc.ec.europa.eu/apps/country.profile/overviewaoi/AOI/UN\\_AFR](https://gwis.jrc.ec.europa.eu/apps/country.profile/overviewaoi/AOI/UN_AFR). Accessed: 2025-05-13.
- [51] Algerian Press Service. *Feux de forêt: plus 100.000 hectares ravagés dans 21 wilayas durant l’été 2021*. Accessed: 2025-05-21. May 2022. URL: <https://www.aps.dz/economie/139524-feux-de-foret-plus-100-000-hectares-ravages-dans-21-wilayas-durant-1-ete-2021>.
- [52] Global Forest Watch. *Algeria - Dashboard: Fires*. Accessed: 2025-05-13. URL: <https://www.globalforestwatch.org/dashboards/country/DZA/?category=fires>.
- [53] International Federation of Red Cross and Red Crescent Societies (IFRC). *Emergency Plan of Action (EPoA): Algeria Forest Wildfires*. 2021.
- [54] Omdena. *Developing a Forest Fire Detection and Monitoring System for Algeria*. <https://www.omdena.com/projects/developing-a-forest-fire-detection-and-monitoring-system-for-algeria>. Accessed: May 2025. 2023.
- [55] UP42. *How to Track Forest Fires with Satellite Imagery and the Normalized Burn Ratio*. Blog post. Accessed: 2025-05-22. 2023. URL: <https://up42.com/blog/track-forest-fires-satellite-imagery-normalized-burn-ratio>.
- [56] Gabriel Henrique de Almeida Pereira et al. “Active fire detection in Landsat-8 imagery: A large-scale dataset and a deep-learning study”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 178 (2021), pp. 171–186. ISSN: 0924-2716. DOI: <https://doi.org/10.1016/j.isprsjprs.2021.06.002>. URL: <https://www.sciencedirect.com/science/article/pii/S092427162100160X>.
- [57] Abdelghania Aba. *Wildfire Prediction Dataset*. <https://www.kaggle.com/datasets/abdelghaniaaba/wildfire-prediction-dataset>. 2023.
- [58] Louis Giglio, Ivan Csiszar, and Wilfrid Schroeder. “Global Limitations of Satellite-Based Fire Detection: Challenges and Perspectives”. In: *Remote Sensing* 16.1 (2024), p. 42. DOI: [10.3390/rs16010042](https://doi.org/10.3390/rs16010042). URL: <https://www.mdpi.com/2072-4292/16/1/42>.

- [59] Sabrina Boumaiza and Nour El Houda Brahimi. “Wildfire Segmentation in Satellite Images Using Deep Learning Techniques”. Available upon request or internal university archive. Master’s thesis. Biskra, Algeria: University of Biskra, 2023.
- [60] George L. James et al. “An Efficient Wildfire Detection System for AI-Embedded Applications Using Satellite Imagery”. In: *Fire* 6.4 (2023). ISSN: 2571-6255. URL: <https://www.mdpi.com/2571-6255/6/4/169>.
- [61] Kabir Yahuza et al. “Recent Advancements in Detection and Quantification of Malaria Using Artificial Intelligence”. In: *UMYU Journal of Microbiology Research (UJMR)* 9 (Sept. 2024), pp. 1–17. DOI: [10.47430/ujmr.2492.001](https://doi.org/10.47430/ujmr.2492.001).
- [62] A. Vali, S. Comai, and M. Matteucci. *Remote Sensing and Deep Learning*. In Encyclopedia. Available at: <https://encyclopedia.pub/entry/1704>. Aug. 2020.
- [63] Ron Karjian. *History and Evolution of Machine Learning: A Timeline*. Accessed: June 8, 2025. Available at: <https://www.techtarget.com/whatis/feature/History-and-evolution-of-machine-learning-A-timeline>. June 2024.
- [64] Lei Ma et al. “Deep Learning in Remote Sensing Applications: A Meta-Analysis and Review”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 152 (2019), pp. 166–177. ISSN: 0924-2716. DOI: [10.1016/j.isprsjprs.2019.04.015](https://doi.org/10.1016/j.isprsjprs.2019.04.015).
- [65] Qiangqiang Yuan et al. “Deep Learning in Environmental Remote Sensing: Achievements and Challenges”. In: *Remote Sensing of Environment* 241 (2020), p. 111716. ISSN: 0034-4257. DOI: [10.1016/j.rse.2020.111716](https://doi.org/10.1016/j.rse.2020.111716).
- [66] Xiaoyu Zhang et al. “Deep Learning Library Testing: Definition, Methods and Challenges”. In: *ACM Computing Surveys* 57.7 (Mar. 2025), pp. 1–37. DOI: [10.1145/3716497](https://doi.org/10.1145/3716497).
- [67] V7 Labs. *Deep Learning Guide: Everything You Need to Know*. Accessed: 2024-06-04. 2023. URL: <https://www.v7labs.com/blog/deep-learning-guide>.
- [68] DataCamp Team. *Deep Learning Tutorial - A Guide for Beginners*. Accessed: 2024-06-04. 2023. URL: <https://www.datacamp.com/tutorial/tutorial-deep-learning-tutorial>.
- [69] Built In. *Deep Learning: What it is, Why it Matters, and How it Works*. Accessed: 2025-06-05. Available at: <https://builtin.com/machine-learning/deep-learning>. 2023.
- [70] Tim P. Vogels, Kanaka Rajan, and Larry F. Abbott. “Neural Network Dynamics”. In: *Annual Review of Neuroscience* 28.1 (2005), pp. 357–376.
- [71] Phil Picton. *What is a Neural Network?* Springer, 1994.

- [72] Codingal. *What is a Neural Network? - Coding for Kids*. Accessed: 2025-06-05. Available at: <https://www.codingal.com/coding-for-kids/blog/what-is-a-neural-network/>. 2024.
- [73] R. Sasikala and R. Radhakrishnan. *Machine Learning and Deep Learning Architectures*. Accessed: 2024-06-05. Available at: <https://developer.ibm.com/articles/cc-machine-learning-deep-learning-architectures/>. IBM Developer. 2020.
- [74] GeeksforGeeks. *Feedforward Neural Network*. Accessed: 2025-06-05. Available at: <https://www.geeksforgeeks.org/feedforward-neural-network/>. GeeksforGeeks. n.d.
- [75] Robin M. Schmidt. “Recurrent Neural Networks (RNNs): A Gentle Introduction and Overview”. In: *arXiv preprint arXiv:1912.05911* (2019).
- [76] Data Aspirant. *How Recurrent Neural Network (RNN) Works?* Accessed: 2025-06-05. Available at: <https://dataaspirant.com/how-recurrent-neural-network-rnn-works/>. Data Aspirant. n.d.
- [77] Data Science Dojo. *Transformer Models: Types and Their Uses*. Accessed: 2025-06-07. Data Science Dojo. n.d. URL: <https://datasciencedojo.com/blog/transformer-models-types-their-uses/>.
- [78] Great Learning. *Understanding Transformer Architecture*. Accessed: 2025-06-05. Available at: <https://www.mygreatlearning.com/blog/understanding-transformer-architecture/>. Great Learning. n.d.
- [79] Zewen Li et al. “A survey of convolutional neural networks: analysis, applications, and prospects”. In: *IEEE transactions on neural networks and learning systems* 33.12 (2021), pp. 6999–7019.
- [80] Pinecone. *Convolutional Neural Networks for Image Search*. Accessed: 2025-06-05. Available at: <https://www.pinecone.io/learn/series/image-search/cnn/>. Pinecone. n.d.
- [81] Lynn Miller, Charlotte Pelletier, and Geoffrey I. Webb. “Deep Learning for Satellite Image Time-Series Analysis: A review”. In: *IEEE Geoscience and Remote Sensing Magazine* 12.3 (2024), pp. 81–124. DOI: [10.1109/MGRS.2024.3393010](https://doi.org/10.1109/MGRS.2024.3393010).
- [82] Parsa Moghadam, Peter Nowak, and Dieter Möller. “A Deep Learning Approach for Wildfire Detection Using Satellite Images”. In: *Environmental Modelling & Software* 167 (2023). Accessed: 2025-06-03, p. 105723. DOI: [10.1016/j.envsoft.2023.105723](https://doi.org/10.1016/j.envsoft.2023.105723).
- [83] Affan Bachri and Muhammad Alif Haikal Bin Ahmad. “Comparison Analysis of Satellite Images for Wildfire Detection Using Convolutional Neural Network (CNN)”. In: *Evolution of Information, Communication and Computing Systems (EICCS)*. Accessed: 2025-06-06. Available at: <https://publisher.uthm.edu.my/bookseries/index.php/eiccs>. UTHM Publisher, 2024, pp. 102–113.

- [84] Zhong Huang et al. “A Novel Wildfire Detection Method Using Multispectral Satellite Imagery and Deep Learning”. In: *Remote Sensing* 15.19 (2023). Accessed: 2025-06-06, p. 4855. DOI: [10.3390/rs15194855](https://doi.org/10.3390/rs15194855).
- [85] Mounia Aarich, Awatif Rouijel, and Aouatif Amine. “Deep Learning Approaches for Forest Fires Detection and Prediction Using Satellite Images”. In: *Procedia Computer Science* 251 (2024), pp. 758–763. ISSN: 1877-0509. DOI: [10.1016/j.procs.2024.11.181](https://doi.org/10.1016/j.procs.2024.11.181).
- [86] EasyODM. *Object Detection with Convolutional Neural Networks (CNNs)*. Accessed: 2025-06-07. EasyODM. n.d. URL: <https://easyodm.tech/object-detection-with-cnns/>.
- [87] Juan Du. “Understanding of object detection based on CNN family and YOLO”. In: *Journal of Physics: Conference Series*. Vol. 1004. IOP Publishing. 2018, p. 012029.
- [88] Ameera Al-Karkhi. *Understanding YOLOv8 Architecture: A Breakthrough in Real-time Artifact Detection*. Accessed: 2025-06-07. Available at: <https://toronto.ieee.ca/newsletter/understanding-yolov8-architecture-a-breakthrough-in-real-time-artifact-detection/>. IEEE Toronto Section. July 2023.
- [89] Baris Kayalibay, Grady Jensen, and Patrick van der Smagt. “CNN-Based Segmentation of Medical Imaging Data”. In: *arXiv preprint arXiv:1701.03056* (2017).
- [90] Équipe Blent. *U-Net : Le réseau de neurones populaire en Computer Vision*. Accessed: 2025-06-08. Available at: <https://blent.ai/blog/a/unet-computer-vision>. Blent.ai. Mar. 2022.
- [91] Benlahmar. *Architecture U-Net: Une explication détaillée*. Accessed: 2025-06-08. Available at: <https://datasciencetoday.net/index.php/en-us/deep-learning/228-unet>. DataScienceToday. 2018.
- [92] Enze Xie et al. “SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers”. In: *arXiv preprint arXiv:2105.15203* (2021). Accepted by NeurIPS 2021.
- [93] Le Yang, Xing Wang, and Jingsheng Zhai. “Waterline Extraction for Artificial Coast With Vision Transformers”. In: *Frontiers in Environmental Science* 10 (Feb. 2022). DOI: [10.3389/fenvs.2022.799250](https://doi.org/10.3389/fenvs.2022.799250).
- [94] Ahmed A. Elngar et al. “Image Classification Based on CNN: A Survey”. In: *Journal of Cybersecurity and Information Management* 6.1 (2021), pp. 18–50.
- [95] Lense.fr. *Les incendies de Californie vus du ciel*. 2020. URL: <https://www.lense.fr/les-incendies-de-californie-vus-du-ciel/>.
- [96] Petru P. *What is EfficientNet?* Accessed: 2025-06-02. Available at: <https://blog.roboflow.com/what-is-efficientnet/>. 2023.

- [97] U.S. Geological Survey. *Landsat 8 (L8) Data Users Handbook*. Tech. rep. Version 5.0. Accessed June 2025. Sioux Falls, SD: U.S. Department of the Interior, U.S. Geological Survey, Nov. 2019. URL: <https://www.usgs.gov/media/files/landsat-8-data-users-handbook>.
- [98] H. Singh, L. M. Ang, and S. Srivastava. “Active wildfire detection via satellite imagery and machine learning: an empirical investigation of Australian wildfires”. In: *Natural Hazards* 121 (2025), pp. 9777–9800. DOI: [10.1007/s11069-025-07163-w](https://doi.org/10.1007/s11069-025-07163-w). URL: <https://doi.org/10.1007/s11069-025-07163-w>.
- [99] University of Alaska Fairbanks, Geographic Information Network of Alaska (GINA). *0.49 micron quick guide*. Accessed June 2025. 2022. URL: <https://gina.alaska.edu/wp-content/uploads/2022/06/0.49-micron-quick-guide.pdf>.
- [100] KDnuggets. *The Importance of Preprocessing in Machine Learning*. <https://www.kdnuggets.com/2023/02/importance-preprocessing-machine-learning.html>. Accessed: [01-06-2025]. 2023.
- [101] Ultralytics. *Learning Rate - Ultralytics Glossary*. 2023. URL: <https://www.ultralytics.com/glossary/learning-rate>.
- [102] K. Y. Leung. *Hyperparameter Tuning for Deep Learning*. Course notes for AA548. 2025. URL: [https://faculty.washington.edu/kymleung/\\_pages/courses/2025/aa548/html/dl/Hyperparameter\\_tuning.html](https://faculty.washington.edu/kymleung/_pages/courses/2025/aa548/html/dl/Hyperparameter_tuning.html) (visited on 06/03/2025).
- [103] Coursera. *What Does Batch Size Mean in Deep Learning?* Coursera. n.d. URL: <https://www.coursera.org/articles/what-does-batch-size-mean-in-deep-learning> (visited on 03/15/2024).
- [104] SabrePC. *Epochs, Batch Size, Iterations - Understanding the Differences*. SabrePC. n.d. URL: <https://www.sabrepc.com/blog/Deep-Learning-and-AI/Epochs-Batch-Size-Iterations> (visited on 03/15/2024).
- [105] Simplilearn. *What is Epoch in Machine Learning?* Simplilearn. n.d. URL: <https://www.simplilearn.com/tutorials/machine-learning-tutorial/what-is-epoch-in-machine-learning> (visited on 06/10/2024).
- [106] Milvus. *How does the number of training epochs during finetuning affect the quality of a sentence transformer model versus the risk of overfitting?* Milvus. n.d. URL: <https://milvus.io/ai-quick-reference/how-does-the-number-of-training-epochs-during-finetuning-affect-the-quality-of-a-sentence-transformer-model-versus-the-risk-of-overfitting> (visited on 06/10/2024).
- [107] IBM. *What Is a Loss Function?* Explanation of loss functions in machine learning. IBM. n.d. URL: <https://www.ibm.com/think/topics/loss-function> (visited on 06/01/2025).

- [108] Ammar Arora. *Understanding Focal Loss*. Technical blog post explaining Focal Loss for object detection. June 29, 2020. URL: <https://amaarora.github.io/posts/2020-06-29-FocalLoss.html> (visited on 06/01/2025).
- [109] Tsung-Yi Lin et al. *Focal Loss*. Method description with implementation details. Papers With Code. 2017. URL: <https://paperswithcode.com/method/focal-loss> (visited on 06/01/2025).
- [110] SoftwareMill. *Instance Segmentation Loss Functions*. Technical overview of loss functions for instance segmentation. SoftwareMill. 2023. URL: <https://softwaremill.com/instance-segmentation-loss-functions/> (visited on 05/10/2025).
- [111] Arize AI. *Understanding Binary Cross-Entropy/Log Loss: A Visual Explanation*. Online; accessed [today's date]. Arize AI. 2023. URL: <https://arize.com/blog-course/binary-cross-entropy-log-loss/> (visited on 11/01/2023).
- [112] GeeksforGeeks. *Binary Cross Entropy/Log Loss for Binary Classification*. Online; accessed [today's date]. GeeksforGeeks. 2023. URL: <https://www.geeksforgeeks.org/binary-cross-entropy-log-loss-for-binary-classification/> (visited on 11/01/2023).
- [113] Built In. *Adam Optimization Algorithm for Machine Learning*. Online; accessed [today's date]. Built In. 2023. URL: <https://builtin.com/machine-learning/adam-optimization> (visited on 11/01/2023).
- [114] V7 Labs. *Intersection over Union (IoU) in Object Detection & Segmentation*. V7 Labs. 2021. URL: <https://www.v7labs.com/blog/intersection-over-union-guide> (visited on 05/27/2025).
- [115] Adrian Rosebrock. *Intersection over Union (IoU) for Object Detection*. PyImageSearch. 2016. URL: <https://pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/> (visited on 05/25/2025).
- [116] Evidently AI. *Confusion Matrix in Machine Learning: A Complete Guide with Examples*. Evidently AI. 2021. URL: <https://www.evidentlyai.com/classification-metrics/confusion-matrix> (visited on 03/15/2024).
- [117] Plutora. *Machine Learning Model Deployment Guide*. Accessed: 2025-06-10. 2025. URL: <https://www.plutora.com/blog/machine-learning-model-deployment-guide>.