

DOT/FAA/TC-26/17

Federal Aviation Administration
William J. Hughes Technical Center
Aviation Research Division
Atlantic City International Airport
New Jersey 08405

Using Artificial Intelligence and Machine Learning for Standardized Fire Test Evaluation

July 2026

Final report



U.S. Department of Transportation
Federal Aviation Administration

NOTICE

This document is disseminated under the sponsorship of the U.S. Department of Transportation in the interest of information exchange. The U.S. Government assumes no liability for the contents or use thereof. The U.S. Government does not endorse products or manufacturers. Trade or manufacturers' names appear herein solely because they are considered essential to the objective of this report. The findings and conclusions in this report are those of the author(s) and do not necessarily represent the views of the funding agency. This document does not constitute FAA policy. Consult the FAA sponsoring organization listed on the Technical Documentation page as to its use.

This report is available at the Federal Aviation Administration William J. Hughes Technical Center's Full-Text Technical Reports page: actlibrary.tc.faa.gov in Adobe Acrobat portable document format (PDF).

Form DOT F 1700.7 (8-72)

Reproduction of completed page authorized

1. Report No. DOT/FAA/TC-26/17		2. Government Accession No.		3. Recipient's Catalog No.	
4. Title and Subtitle Using Artificial Intelligence and Machine Learning for Standardized Fire Test Evaluation				5. Report Date July 2026	
				6. Performing Organization Code ANG-E21	
7. Author(s) Ahmed Ben Sidhom(1), Giuseppina Carannante(1), and Nidhal C. Bouaynaya(1), Richard N. Walters(2)				8. Performing Organization Report No.	
9. Performing Organization Name and Address (1) Department of Electrical and Computer Engineering, Rowan University, Glassboro, NJ (2) Federal Aviation Administration, William J. Hughes Technical Center, Atlantic City International Airport, NJ 08405 USA				10. Work Unit No. (TRAIS)	
				11. Contract or Grant No.	
12. Sponsoring Agency Name and Address Fire Safety Branch ANG-E211				13. Type of Report and Period Covered	
				14. Sponsoring Agency Code ANG-E2	
15. Supplementary Notes This research report is a result of the collaborative work performed through the FAA Micro-credentialing Program.					
16. Abstract Fire safety remains a paramount concern for the Federal Aviation Administration (FAA), particularly in the context of passenger aircraft materials and cabins. The FAA utilizes standardized bench-scale fire tests, including the Vertical Flame Propagation (VFP) apparatus, to assess material flammability, flame spread, and burning behavior under controlled laboratory conditions. These tests provide critical data on flame propagation, burn-through, and post-flame characteristics essential for certification and rulemaking to ensure passenger safety by minimizing fire hazards. This study advances FAA's Fire Safety Branch objectives by integrating deep learning and computer vision, specifically, object detection algorithms such as Faster Region- Convolution Neural Network (Faster R-CNN), to analyze video records of FAA regulatory flame tests. These algorithms automate the detection and quantification of fire features from video frames, including ignition timing, burn length, flame intensity, and char area, eliminating subjectivity inherent in traditional operator observations. The application of image preprocessing and data augmentation enhances detection accuracy, while transfer learning and deeper network architectures improve model robustness. Performance evaluations on videos from the FAA's VFP tests demonstrate that deep neural networks can reliably identify fire presence, quantify burning characteristics, and extract precise temporal and spatial fire metrics. This automated and quantitative approach has the potential to standardize material flammability assessments, improve repeatability, and facilitate regulatory compliance, ultimately contributing to enhanced aircraft fire safety.					
17. Key Words VFP, fire detection, fire quantifying, object detection, deep neural network, Faster- RCNN			18. Distribution Statement This document is available to the U.S. public through the National Technical Information Service (NTIS), Springfield, Virginia 22161. This document is also available from the Federal Aviation Administration William J. Hughes Technical Center at actlibrary.tc.faa.gov .		
19. Security Classif. (of this report) Unclassified		20. Security Classif. (of this page) Unclassified		21. No. of Pages 66	
				22. Price	

Contents

1	State-of-the-art fire detection.....	1
1.1	Introduction.....	1
1.2	Fire and flammability testing.....	1
1.2.1	Chemistry and physics of fire	1
1.2.2	Classification of fire.....	6
1.2.3	Flammability tests.....	6
1.3	Fire detection technologies	9
1.3.1	Traditional techniques.....	9
1.3.2	Computer-vision based techniques	11
1.4	Conclusion	18
2	Faster R-CNN for fire detection	18
2.1	Introduction.....	18
2.2	Introduction to deep learning.....	19
2.2.1	Understanding how deep learning works.....	19
2.2.2	Hyperparameter tuning	21
2.2.3	Evaluation metrics	25
2.3	Object detection	26
2.3.1	Measuring object detector performance.....	27
2.4	Faster R-CNN architecture	28
2.5	Conclusion	31
3	Approach	31
3.1	Introduction.....	31
3.2	Data preparation.....	32
3.2.1	Data collection and discovery	32
3.2.2	Data structuring and preprocessing.....	32
3.3	Building the model.....	34

3.4	Information extraction	36
3.4.1	Object detection algorithm.....	36
3.4.2	Image processing	39
3.5	Conclusion	40
4	Results and discussion	40
4.1	Introduction.....	40
4.2	Environment.....	41
4.2.1	Hardware environment: Google Colaboratory.....	41
4.2.2	Software environment: Detectron2	41
4.3	Results.....	42
4.3.1	Object detection results	42
4.3.2	Feature extraction results	47
4.4	Discussion	49
4.5	Conclusion	50
5	References	52

Figures

Figure 1: Fundamental elements of fire triangle.....	2
Figure 2: Elements of fire tetrahedron	2
Figure 3: Heat conduction through solid medium	4
Figure 4: Heat transfer through convection	5
Figure 5: Heat through radiation.....	5
Figure 6: FAA Vertical Flame Propagation test	9
Figure 7: Examples of fire detection using image classification	11
Figure 8: Examples of fire detection using segmentation.....	12
Figure 9: Advanced image processing for fire detection	13
Figure 10: Example of faster-RCNN	14
Figure 11: Example of YOLO v3	15
Figure 12: Fire detection using object detection showing detected region of interest.....	17
Figure 13: Analogy between biological neuron (a) and artificial neuron (b)	19
Figure 14: Examples of shallow and deep neural networks	20
Figure 15: Loss function measures quality of network's output.....	22
Figure 16: Loss score used as feedback signal to adjust weights	23
Figure 17: Convolutional neural network architecture	24
Figure 18: Images broken into local patterns (edges, textures)	25
Figure 19: Example of detecting a stop sign.....	27
Figure 20: Comparison of computing IOU for bounding boxes.....	27
Figure 21: R-CNN architecture example	28
Figure 22: Fast R-CNN architecture example	30
Figure 23: Faster R-CNN example	30
Figure 24: Manual annotation using the LabelImg tool	32
Figure 25: Data splitting example.....	33
Figure 26: Preprocessing data and accounting for interferences	34
Figure 27: Data augmentation.....	34
Figure 28: Resnet 50 FPN architecture example	35
Figure 29: Residual connection around a processing block.....	36
Figure 30: Time of burning extraction.....	37
Figure 31: Images representing the burn length extraction	37
Figure 32: Bounding boxes showing the number of droplets extraction	38
Figure 33: Burn scale extraction from images	38
Figure 34: Flame intensity extraction by converting images to HSV.....	39

Figure 35: Charring area extraction from end-of-test frames	40
Figure 36: Extraction of results from additional inputs defining a region.....	40
Figure 37: Training of R50_FPN.....	41
Figure 38: Detectron2 model zoo	42
Figure 39: Optimization results on raw data.....	43
Figure 40: Optimization results on rotated data.....	44
Figure 41: Brightness optimization results	45
Figure 42: Total loss curve for the blur filter.....	46
Figure 43: Examples of inference from other data	46
Figure 44: 1000 steps loss vs 2000 steps loss	47
Figure 45: Object detection results from algorithm.....	48
Figure 46: Light intensity extraction from flaming sample	49
Figure 47: Post-burn charring area extraction	49

Tables

Table 1: CNN based vs manually selected based architecture	16
Table 2: Performance of the object detection models.....	16
Table 3: Comparison results of various fire detection approaches	18
Table 4: Tabulated results for the material class identified by the algorithm.....	47

Acronyms

Acronym	Definition
CNN	Convolution Neural Network
FAA	Federal Aviation Administration
Faster R-CNN	Faster Region-Convolution Neural Network
FPN	Feature Pyramid Network
IoU	Intersection of Union
LDS	Linear Dynamical Systems
LEL	Lower explosive limit
mAP	Mean Average Precision
R-CNN	Region-based convolutional neural networks
R-FCN	Region-based fully convolutional network
ReLU	Rectified linear unit
ROI	Region of interest
RPN	Region proposal network
RTCA	Radio Technical Commission for Aeronautics
SSD	Singe Shot Detector
UEL	Upper explosive limit
VFP	Vertical Flame Propagation
VLAD	Vector of Locally Aggregated Descriptors
YOLO	You Only Look Once

Executive summary

Fire safety is of great concern to the Federal Aviation Administration (FAA). Fire detection and feature extraction using object detection algorithms is an important area of research and development aimed at improving early fire detection systems. Traditional fire detection methods often rely on smoke or heat sensors, which can be limited in their effectiveness, especially in complex environments or situations where smoke may not be present. In this study, algorithms used for fire detection were developed to evaluate the burning behavior of materials in a controlled environment. Video records of standardized regulatory test methods were evaluated to extract the parameters used to assess material flammability results for rule-making purposes.

Object detection algorithms, such as YOLO (you only look once) or Faster R-CNN (region-based convolutional neural networks), offer a promising approach to fire detection by leveraging the capabilities of deep learning and computer vision. These algorithms can automatically identify and localize objects of interest within images or video frames, making them well-suited for fire detection and analysis.

In order to use object detection algorithms for fire detection, the algorithm must be trained using a dataset of fire images or videos. This training process involves feeding the algorithm with labeled data, where the presence of fire is marked as the target object. The algorithm learns to recognize fire patterns and characteristics by analyzing various visual features, such as color, texture, shape, and motion.

Once the object detection algorithm is trained, it can be applied to real-time video streams or images to detect and locate fires while quantifying metrics of interest. The algorithm analyzes each frame or image, searching for fire objects based on the learned features. When a fire is detected, the algorithm outputs the coordinates or bounding box around the fire, providing information about its location within the scene.

Feature extraction plays a crucial role in this process. By extracting relevant features from the fire objects, such as flame color, size, intensity, or motion patterns, additional information about the fire can be obtained. These features can be used for further analysis, classification, or even for estimating the severity of the fire and evaluating the criteria for pass/fail fire tests used for certification of aircraft materials.

Despite their potential and their effectiveness in fire detection, object detection methods do have some limitations. Challenges such as varying lighting conditions, occlusions, or complex

backgrounds can affect their performance. Therefore, ongoing research and development in this field aim to address these challenges and further improve the accuracy and reliability of fire detection systems based on object detection algorithms.

In summary, fire detection and feature extraction using object detection algorithms leverage deep learning and computer vision techniques to automatically detect fires and extract relevant information. By combining object detection with feature analysis, these algorithms can improve the repeatability and reproducibility of tests dependent on operator observations and have the potential to enhance fire safety and emergency response in various applications and environments.

The following report is organized in four sections:

- The first section contains a state-of-the-art study of the detection procedure that details the structure of the fire. Then the vertical flame propagation (VFP) test technique is detailed and highlighted its importance for early detection and features extraction.
- The second section presents artificial intelligence algorithms and explains deep neural networks, in particular, as a powerful architecture for intelligent task automation while focusing on the object detection algorithms and the Faster R-CNN architecture.
- The third section is dedicated to the working strategy development, where the methods used for image processing, customized model development and parameter optimization are detailed.
- The final section is for model evaluation and performance testing using chosen parameters, a comparative study to validate the performance of this strategy.

1 State-of-the-art fire detection

1.1 Introduction

The growth, spread, and development of fire are all impacted by a variety of circumstances. The physical makeup and condition of the fuel, the amount of oxygen available, and the transfer of heat are all crucial factors in how a fire develops. While each fire is unique, they all exhibit a few common characteristics that, when recognized by the investigator, can be used to establish the origin and causation of a fire. Small-scale standardized tests have been developed to characterize the burning behavior of materials. The Federal Aviation Administration (FAA) utilizes bench scale fire tests to assess the flammability and flame spread of materials used in passenger aircraft (Marker, 2019). These tests have well defined ignition criteria and controlled environmental conditions to minimize variables that can influence the burning behavior. However, these tests are not quantitative and are dependent on operator observations of burning behavior to determine pass/fail criteria.

1.2 Fire and flammability testing

Fire is a complicated chemical process, and investigators must be familiar with the fundamental chemistry and physics involved in order to derive meaningful values and develop conclusions based on principles rather than operator observations. An investigator may be unable to appropriately assess the cause, origin, and spread of a fire if they are not able to explain the technical aspects of material fire performance and fire behavior. Tests are used to aid in the characterization and assessment of material fire performance. The FAA has developed several bench-scale fire tests to simulate fire scenarios that caused a fire event. These tests are then used to evaluate materials for specific applications and prevent future incidents under similar conditions. Again, many of these tests are dependent on operator observations and opinions, and results can vary based on this.

1.2.1 Chemistry and physics of fire

Three fundamental components make up the diffusion flame process (fire): fuel, oxygen, and heat. For more than a century, the science of fire protection has acknowledged these fundamental elements (Drysdale, 1986).

1.2.1.1 Life cycle of fire

The six components that make up a fire's life cycle are heat, fuel, oxygen, proportioning, mixing, and continuation of ignition (NFPA, 1991). The diffuse flame combustion process cannot begin

or continue without each of these components. The fire triangle, shown in Figure 1, symbolizes the initial three elements: heat input, fuel, and oxygen.

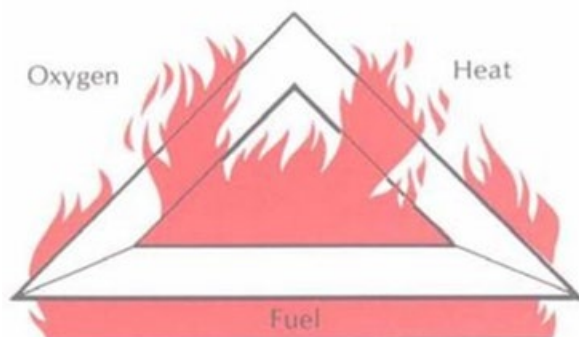


Figure 1: Fundamental elements of fire triangle

The combustion reaction can be depicted more accurately by a four-sided solid geometric form called a tetrahedron. The four sides of the tetrahedron shown in Figure 2 represent heat, fuel, oxygen, and uninhibited chain reactions. A chain reaction occurs when there is an exchange of electrons creating an energy transfer and adding heat to the system. These chain reactions are a result of uncontrolled flow of electrons and do not require oxygen to proceed. If any one of these elements of the tetrahedron is interrupted, the fire will go out.

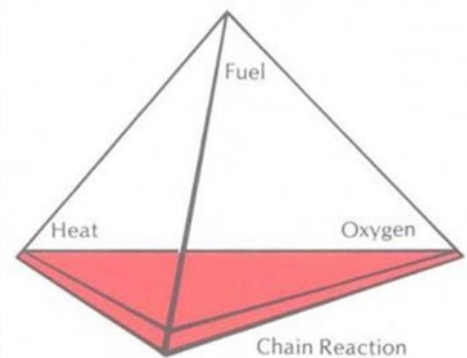


Figure 2: Elements of fire tetrahedron

1.2.1.2 Input heat

Materials that are solid or liquid cannot burn. These materials need to be heated enough to create vapor in order for combustion to occur. These fumes are the fuel and what ignites the flames. The flashpoint is the lowest temperature at which a solid or liquid material emits enough vapor to ignite under controlled laboratory circumstances. The flame point, or the temperature at which the fuel will continue to release enough vapor (fuel generation rate) to support an ongoing flame, is located a few degrees above the flashpoint. The ignition temperature, also known as the auto-ignition temperature, is the temperature at which the vapors will ignite. Piloted ignition is the

term used when the heat source is an open flame or spark. Radiant heaters with piloted flames and Bunsen burners are used as the heat and ignition sources in FAA regulatory testing. In order for a material to ignite and sustain burning in these tests, chemical bonds need to break to generate the volatile fuels. Commercial passenger aircraft materials are held to higher flammability standards and typically do not sustain burning without additional heat added to the system.

1.2.1.3 Fuel

At room temperature, the fuel may initially take the form of a gas, liquid, or solid. Liquid and solid fuels must be heated sufficiently to form vapors, as was previously discussed. Flammable materials can ignite easily in air under normal ambient temperature and pressure conditions. Combustible materials can be made to burn in air under standard pressures but require elevated temperatures to generate the fuel to ignite. Extra effort is taken when evaluating materials for use in aircraft. Aircraft materials must be able to resist small ignition sources, not propagate a fire if ignited, and provide thermal barrier fires for post-crash fuel spill scenarios. Elevated temperatures and flaming ignition sources are used to simulate more extreme scenarios where better material performance is required.

1.2.1.4 Oxygen

The atmosphere, which contains around 20.9 percent oxygen, is typically the main supply of oxygen for combustion. While charring or smoldering can happen with as little as 1-2 percent oxygen, flaming combustion typically requires a concentration of at least 15 to 16 percent to continue. Pyrolysis is the process of anaerobically decomposing with only use of heat to change a chemical into multiple other volatile compounds. The reaction of the pyrolyzed compounds with oxygen is what provides the heat to propagate the reaction and ignite surrounding materials. Although the atmosphere is typically the main source of oxygen, some substances onboard aircraft, referred to as "oxidizers," can either be the main source or a secondary source of oxygen in a fire (O'Connor & Hagen, 1999). Materials and systems have been put into place to prevent fires on aircraft for the oxidizers that are part of the aircraft design. Additional oxidizing substances that are not part of an aircraft design can be brought onboard as luggage and cargo, where systems are in place but are not tailored to oxidizers.

1.2.1.5 Mixing and proportioning

Continuous mixing and proportioning are two responses that are necessary for the ignition and spread of fire. The right ratios of fuel vapors and oxygen are required to get an ignition. Such mixtures of oxygen and fuel vapors are considered to be within the flammable or explosive limits. Limits for explosive or flammable materials are indicated as a percentage of the air's fuel

vapor concentration. The Lower Explosive Limit (LEL) is the lowest concentration that will ignite, while the Upper Explosive Limit (UEL) is the highest. A combination that has less fuel vapor than is required for ignition is too lean, whereas a mixture with a concentration of fuel vapor that is too high is too rich. Materials will not ignite if the fuel to oxygen ration is too lean or too rich. Once a material is ignited and burning, the thermal feedback into the material has to be sufficient enough to propagate the reaction. If the fuel generation rate is not sufficiently high, the fire will go out.

1.2.1.6 Ignition continuity

The heat feedback from the fire to the fuel is known as ignition continuity. Direct flame contact, convection, radiation, and conduction are all methods of transferring heat. The direct transfer of heat through a solid body, shown in Figure 3, is known as conduction. For instance, on a hot stove, heat is transferred from the burner to the contents of the pot. Metals are good conductors of heat whereas wood typically isn't. Materials that do not conduct heat well are insulators.

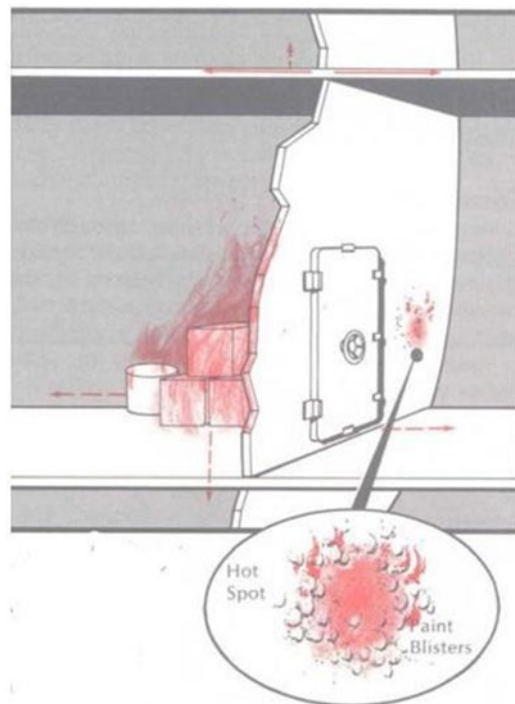


Figure 3: Heat conduction through solid medium

Convection, shown schematically in Figure 4, is the heat transfer process brought on by the transfer of heat between a solid surface and the surrounding fluid (Drysdale, 1986). Flow created by the variations in the liquid and gas densities along with thermal buoyancy is considered natural convection. If the fluid is directed in a continuous stream, it is considered forced convection.

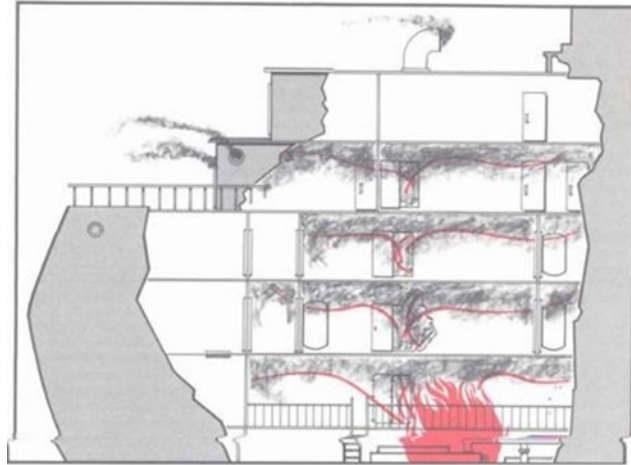


Figure 4: Heat transfer through convection

Radiation, shown in Figure 5, is another recognized method of heat transfer along with conduction and convection. Heat is transferred by infrared radiation (heat waves, such as those from the sun), which is typically invisible to the unaided eye, from a source over an intervening distance through a process known as radiation; no physical object is involved. The way that heat leaves a fire is in straight lines, similar to light. It is transmitted, reflected, or absorbed when it comes into contact with a body. The body that is absorbing the radiant energy becomes hotter. As an illustration, the burning rate of a material determines the radiant heat received by a neighboring material and can raise its temperature, possibly to the point where it also ignites and burns (NFPA, 1991). Radiation, conduction and convection are three components of heat transfer that all contribute to the overall flammability of materials.

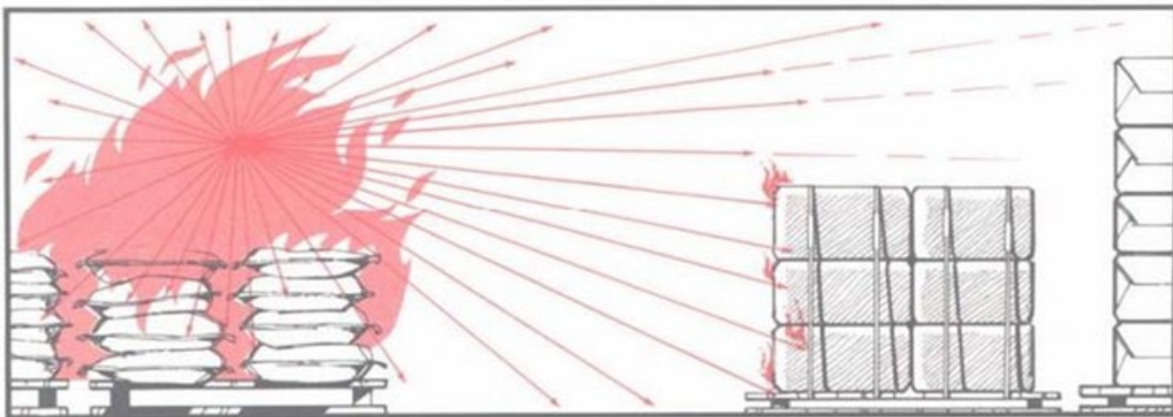


Figure 5: Heat through radiation

1.2.2 Classification of fire

Fires are classified by the types of materials that are burning:

- Class A fires involve ordinary combustible materials, such as wood, cloth, paper, rubber, and many plastics.
- Class B fires involve flammable/combustible liquids, greases, and gases.
- Class C fires involve energized electrical equipment.
- Class D fires involve combustible metals, such as magnesium, titanium, zirconium, sodium, and potassium.
- Class E fires involve fire caused by electrical equipment, such as machinery or appliances.
- Class K fires involve cooking oils fats and grease.

All fires produce combustion products. Combustion products fall into four categories: heat, gases, flame, and smoke.

- Heat is defined as a form of energy characterized by vibration of molecules and capable of initiating and supporting chemical changes and changes of state.
- Gases are substances that have no shape or volume of their own and will expand to take the shape and volume of the space they occupy. Fire gases include complete combustion products such as carbon dioxide and water along with incomplete combustion products such as linear and cyclical hydrocarbons, carbon monoxide, hydrogen cyanide, ammonia, hydrogen chloride, and acrolein.
- Flame is the luminous portion of burning gases or vapors caused by the reaction with oxygen.
- Smoke is the airborne particulate products of incomplete combustion, suspended in gases, vapors, solid or liquid aerosols. Soot, black particles of carbon, are also contained in smoke (DeHaan, 1983).

1.2.3 Flammability tests

Flammability tests are essential for ensuring safety in aircraft, industries and applications. These tests are designed to evaluate the ability of a material to ignite, sustain combustion, and

propagate flames. The necessity for flammability tests can be attributed to several important reasons identified below.

Fire safety

Flammability tests are crucial for assessing the fire hazard potential of materials. They help identify materials that are more likely to ignite, burn rapidly, or contribute to the spread of fire. Heat release rate is the best indicator of the fire hazard of a material (Babrauskas & Grayson, 1992). By understanding the flammability characteristics of materials, appropriate safety measures can be implemented to prevent fires and protect lives and property.

Compliance with regulations and standards

Many industries are subject to specific regulations and standards related to flammability. For example, large surface area materials in passenger aircraft, including seats, wall panels, stowage bins, seat-back tray, and other components, need to meet flammability requirements outlined by domestic and international regulations. Flammability tests provide the necessary data to determine compliance and ensure that materials meet the established safety criteria, preventing loss of life.

Risk assessment and mitigation

Flammability tests play a key role in assessing the fire risk associated with different materials. By understanding the flammability properties, such as ignition time, flame spread rate, heat release rate, and smoke production, appropriate risk mitigation strategies can be developed. This includes designing fire-resistant structures, implementing fire suppression systems, and selecting suitable materials for specific applications.

Product development and innovation

Flammability tests are essential during the product development phase. By conducting these tests, manufacturers can evaluate the fire performance of new materials, components, or products during development. This information enables them to make informed decisions regarding material selection, design modifications, or the inclusion of additional fire protection features. Flammability tests also drive innovation in the development of new fire-resistant materials and technologies.

Insurance and liability considerations

Flammability information plays a significant role in insurance and liability assessments. Insurance providers may require flammability test data to determine coverage and premium rates for businesses and industries. Similarly, manufacturers and suppliers may need to demonstrate

compliance with flammability standards to mitigate potential liability claims arising from fire incidents.

Public safety and consumer confidence

Flammability tests contribute to public safety by ensuring that products and materials available to consumers meet established safety standards. Consumers have the right to expect that the products they use are not excessively combustible and pose minimal fire risk. Flammability testing helps build consumer confidence, as it demonstrates a commitment to safety and quality. The FAA continuously implements new safety regulations through rules and advisory circulars. Sometimes in-service aircraft can be retrofit during routine maintenance schedules over the course of months or years. Other times, when the threat is more severe, entire fleets can be grounded until the hazards are eliminated.

In summary, flammability tests are necessary to ensure fire safety, comply with regulations and standards, assess risk, drive product development and innovation, consider insurance and liability factors, and instill consumer confidence. By evaluating the flammability characteristics of materials, these tests enable the implementation of appropriate safety measures and help prevent fire incidents, thereby protecting lives, property, and the environment (Khandual, 2016).

1.2.3.1 FAA Flammability Tests

The specific tests and requirements may vary depending on the type of material and its application within an aircraft. The FAA regulations aim to ensure that materials used in aircraft interiors meet strict flammability standards, thereby enhancing the overall safety of air travel.

Vertical Flame Propagation Test

Fire test data from Radio Technical Commission for Aeronautics (RTCA) box testing and a vertical flame propagation (VFP) apparatus provided by researchers at the Federal Aviation Administration (FAA) was used for this study. Several other Bunsen burner type tests in the FAA Aircraft Materials Fire Test Handbook (Marker, 2019) are applicable to the technology discussed here but no video data was available at the time of the study. Data from the VFP is the main focus of the analyses presented here. The VFP apparatus shown in Figure 6 assesses the flame spread properties of the materials used in passenger aircraft cabins. Samples of a component or construction are exposed for a predetermined amount of time to a standard flame. Flame propagation, burn-through duration, and after-flame time are among the required metrics.

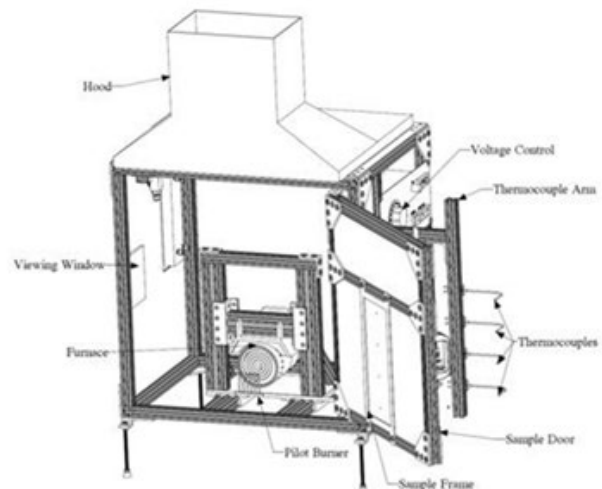
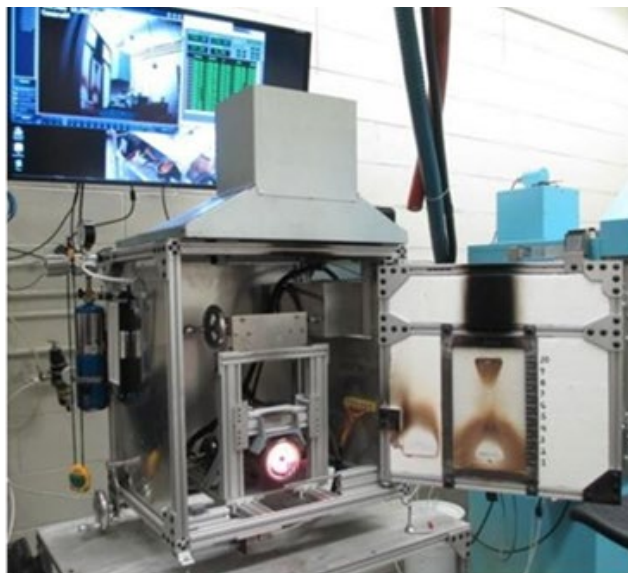


Figure 6: FAA Vertical Flame Propagation test

Material response to heat and flame is tested and reported in controlled laboratory settings. A test sample is exposed for a predetermined amount of time while being held vertically over a controlled flame. The flame source is eliminated after being exposed. The duration of the specimen's flame as well as the afterglow's persistence after the flame source has been extinguished are measured. After applying a specific tearing force, the test specimen's observable damage, or char length, is measured. There are other instances of melting and leaking that are noted (Marker, 2019).

1.3 Fire detection technologies

Over the past ten years, numerous novel fire detection methods and concepts have been developed thanks to advancements in sensor, microelectronics, and information technologies as well as a better knowledge of fire physics. Previous work at the FAA examined the applicability of these sensors for detecting flames and quantifying the results with limited success (Anaya & Rehn).

1.3.1 Traditional techniques

Traditional techniques are based fundamentally on physical sensors or basic naive image processing.

1.3.1.1 Heat detectors

One of the newest and most promising heat detection devices for fire prevention applications is the distributed fiber optic temperature sensor. Given that variations in these physical parameters

change the refractive indices and geometric characteristics of the optical fiber, they can also be used to detect changes in temperature, strain, and tensile force applied to the fiber optic cable. This causes changes in the intensity, phase, or polarization of the light wave propagating within the optical fiber. RFID tags can also be used to detect elevated temperatures (Karp, 2025).

1.3.1.2 Smoke detectors

As a fire grows and develops, smoke is produced far earlier than other fire characteristics. The likelihood of successful fire suppression, evacuation, and survivability can be increased by quick detection of smoke at very low levels. The interaction of the particles with a light or electromagnetic beam can detect smoke. Key characteristics for smoke detection include the mass concentration, volume fraction, and size distribution of the smoke. Due to the structural and chemical differences in the smokes produced by smoldering and flame combustion, a smoke detector must be able to detect both types of smoke. Other gases and particulates in the cabin air can give false positives for smoke detection. Nuisance/false alarms are a major concern for the FAA due to aircraft being taken out of service to inspect and evaluate the causes of the alarms.

1.3.1.3 Flame detectors

The radiation emitted in the burning zone can be used to identify the fire because the fire is a source of radiation in and of itself. Based on the measurement of the flame radiation range, there are now two types of flame detectors: infrared detectors and ultraviolet detectors. Ultraviolet (UV) detectors identify fires when any ultraviolet radiation created by fiery combustion is detected, while infrared detectors (IR) detect fires when they detect the characteristic flame flicker caused by fire. Because flame detectors do not rely on smoke or heat from the fire, they can be utilized to protect broad areas and have a quick reaction. Unlike smoke detectors, which require a ceiling to function, they can also be used outdoors. However, false alarms may be generated by radiation from other sources such as welding, sunlight, and tungsten lamps.

1.3.1.4 Gas sensors

Almost any stable gaseous species formed before or during combustion may currently be measured. Numerous interactions, such as catalytic, electrochemical, mechanic-chemical, and optical processes, can detect chemical species. Many of the gases produced during combustion are, however, also produced naturally or in non-dangerous combustion processes. Additionally, the sensor requires a considerable power source in order to identify particular gas signatures, which limits its ability to function as an affordable fire detector. Also, chemical cells have a lifespan and service, and maintenance costs make them impractical for use in aircraft.

1.3.2 Computer-vision based techniques

In computer vision, there are different techniques to accomplish the tasks in this study.

Basic image processing

Traditional image processing techniques utilized RGB channel comparison to detect different objects such as fire in frames or videos (Demirel & Celik, 2009; Petra, Bakar, DeSilva, & Umar, 2017; Zaidi, Lokman, Daud, Achmad, & Khor, 2015). These traditional methods are not free of errors and are not fully reliable (Yuan, Liu, & Zhang, 2015). For instance, RGB value comparison methods that usually consider a threshold value to detect fire may detect sunset and sunrise as a false positive outcome.

Image classification

In this case, the objective is to give an image one or more labels. It can be either single-label classification, in which case an image can only belong to one category and not the others, or multi-label classification, in which case all of an image's categories are tagged. For instance, when we perform a keyword search in the Google Photos app, a very large multilabel classification model: one with over 20,000 different classes, trained on millions of images, is being accessed in the background.

The following example is a state-of-the-art work done using image classification for fire detection: *Fire Image Detection Based on Convolutional Neural Network*, see Figure 7 (Li & Zhao, 2020).

In this work, a model that can identify the presence of fire in image data was presented. A deep convolutional neural network (CNN) was trained for the image classification task using a transfer learning technique on VGG16 and RESNET50. Very good accuracy of 97% on the validation data and the precision and recall are given by 98.99% and 98.50%, respectively.



Figure 7: Examples of fire detection using image classification

Image segmentation

This technique aims to "segment" or "partition" an image into various sections, often each area denoting a particular category. For instance, Zoom or Google Meet uses an image segmentation model to distinguish our face from what is behind it with pixel-level accuracy when it displays a bespoke background behind users during a video conversation.

The following example is a state-of-the-art work done using image segmentation for fire detection: *Aerial Imagery Pile burn detection using deep learning: the FLAME dataset* (Shamsoshoara, et al., 2021).

Solutions to two machine learning problems were highlighted in this study:

1. Binary classification of video frames based on the presence and absence of fire flames:
An artificial neural network (Xception architecture) method was developed that achieved a 76% classification accuracy.
2. Fire detection using segmentation methods to precisely determine fire borders was also used (see Figure 8). A deep learning method was designed, based on the U-Net up-sampling and down-sampling approach to extract a fire mask from the video frames.

The FLAME dataset and method approached a precision of 92% and recall of 84%.

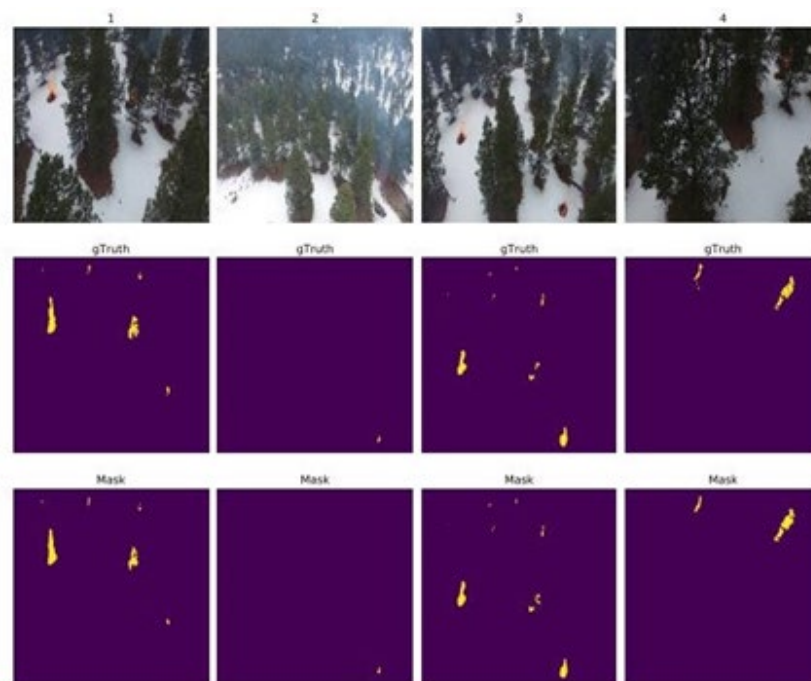


Figure 8: Examples of fire detection using segmentation

Advanced image processing

The following example is a state-of-the-art work done using advanced image processing and video sequences for fire detection: *Vision-Based Fire Detection Using Video Sequences* (Chang, 2002).

In this work they aimed to develop an automatic system for vision-based fire detection using video sequences. Their system included four major steps, namely image preprocessing (background subtraction, fire color filtering, morphological processing, connected component labeling), foreground region analysis, fire dynamic behavior analysis, and fire flow energy analysis. Overall, this system could achieve the detection rates of over 91% in either indoor or outdoor environments. Figure 9 shows an example of the pyramid Lucas-Kanade optical flows method: (a) image containing the candidate fire region after image preprocessing; (b) region contour of (a); (c) estimated motion vectors.

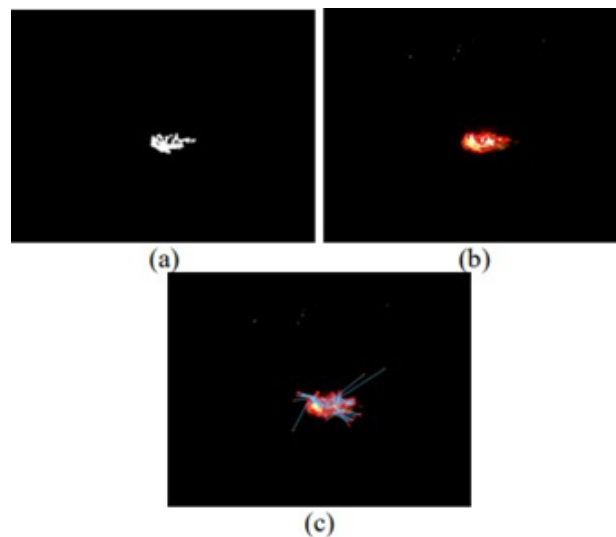


Figure 9: Advanced image processing for fire detection

Object detection

The objective of this technique is to identify things of interest in an image by drawing rectangles around them (known as bounding boxes), then classifying each rectangle. An object-detection model could be used by a self-driving car to track other vehicles, pedestrians, and signs in front of its cameras, for example.

State-of-the-art object detection methods can be categorized into two main types: One-stage vs. two-stage object detectors (Zhao, 2019).

1. State-of-the-art two-stage object detection algorithms

The two-stage architecture involves object region proposal with conventional Computer Vision methods or deep networks, followed by object classification based on features extracted from the proposed region with bounding-box regression.

Two-stage methods achieve the highest detection accuracy but are typically slower, see example of faster-RCNN in Figure 10 (Li & Zhao, 2020).

- RCNN and SPPNet (2014)
- Fast Region- Convolution Neural Network (RCNN) and Faster RCNN (2015)
- Mask R-CNN (2017)
- Pyramid Networks/ Feature Pyramid Network (FPN) (2017)
- G-RCNN (2021)

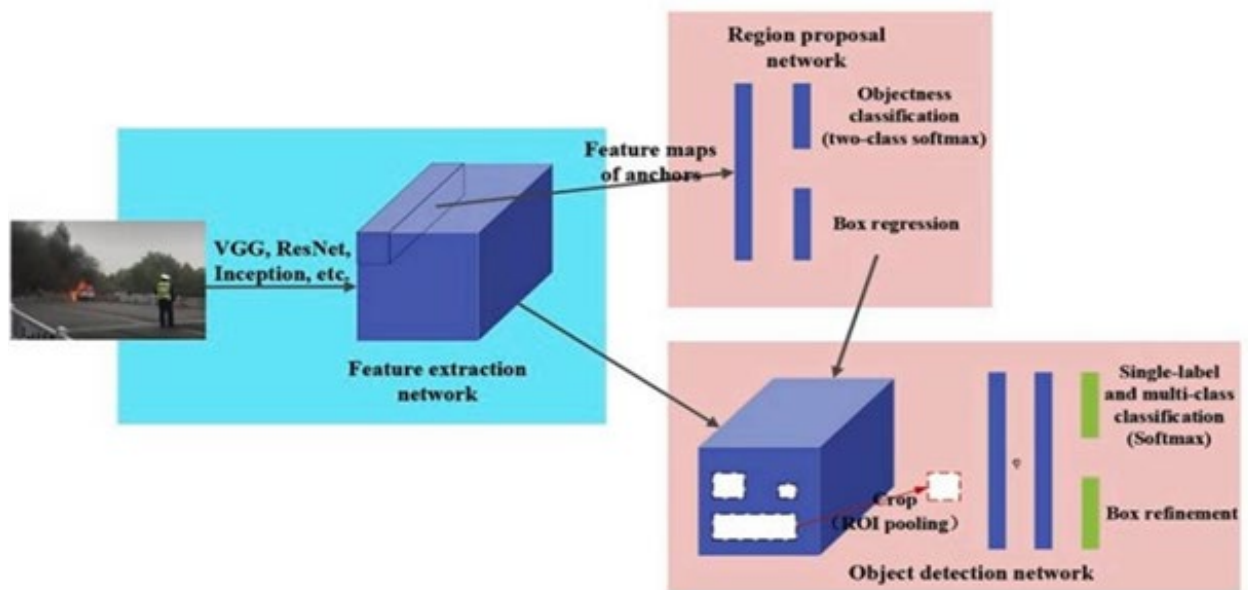


Figure 10: Example of faster-RCNN

2. State-of-the-art one-stage object detection algorithms

One-stage detectors predict bounding boxes over the images without the region proposal step. This process consumes less time and can therefore be used in real-time applications, see YOLO example in Figure 11 (Li & Zhao, 2020).

- YOLO (You Only Look Once) (2016)
- SSD (Single Shot Detector) (2016)
- RetinaNet (2017)
- YOLOv3 (2018)
- YOLOv4 (2020)
- YOLOR (2021)
- YOLOv7 (2022)

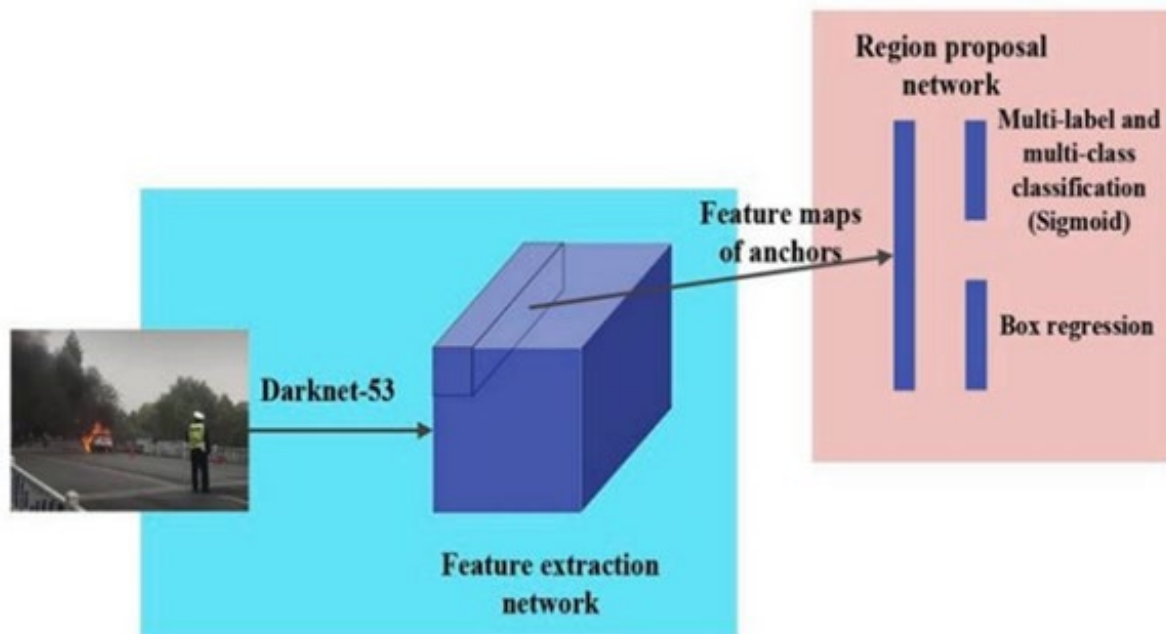


Figure 11: Example of YOLO v3

The following examples are the state-of-the-art work done using object detection for fire detection: *Image fire detection algorithms based on convolutional neural networks* (Li & Zhao, 2020). This study concluded that when image fire detection is based on an algorithmic analysis of images, there is a lower accuracy, delayed detection, and a large amount of computation in common detection algorithms, including manually and machine automatically extracting image features. Therefore, novel image fire detection algorithms based on the advanced object detection CNN models of Faster-RCNN, Region-based fully convolutional network (R-FCN), SSD, and YOLO v3 are proposed in this paper. The first part of this article doesn't provide the Mean Average Precision (mAP) but does provide a comparison of the proposed and current algorithms

applied on classification dataset (Foggia, Saggese, & Vento, 2015), which reveals that the accuracy of classification of the bounding boxes of fire detection algorithms based on CNNs is higher than other algorithms.

As seen in Table 1 (Koirala, 2020), algorithms 1–6 extract features by manually selection, and algorithms 7 and 8 extract features by machine automatically learning based on image classification CNNs. Algorithms 9–12 are proposed in this paper. The results indicate that the algorithms based on CNNs are better than the traditional algorithm extracting fire features manually, as seen in Table 2 (Li & Zhao, 2020).

Table 1: CNN based vs manually selected based architecture

No	Algorithm	Accuracy
1	Chen (Chen, Wu, & Chiou, 2004)	87.10
2	Celik (Celik & Demirel, 2009)	83.87
3	Raffiee (Rafiee, Dianat, Jamshidi, Tavakoli, & Abbaspour, 2011)	87.10
4	Habibuglu (Habiboglu, Günay, & Cetin, 2011)	90.32
5	De Lascio (Di Lascio, et al., 2014)	92.86
6	Foggia (Foggia, Saggese, & Vento, 2015)	93.55
7	Muhammed (Alexnet) (Muhammad, Ahmad, Mehmood, Rho, & Baik, 2018)	94.39
8	Muhammed (Googlenet) (Muhammad, Ahmad, & Baik, 2017)	94.43
9	Faster R-CNN	99.43
10	R-FCN	99.20
11	SSD	98.93
12	YOLO v3	99.62

Table 2: Performance of the object detection models

Algorithm	mAP
Faster-RCNN	82.3
R-FCN	80.9
SSD	77.8
YOLO v3	84.5

The four proposed algorithms were applied on another dataset for object detection developed in this paper. They all achieve high average precision, which indicates that using the object detection CNNs to detect fire in images is entirely feasible. The differences between average precision of the algorithms based on YOLO v3 and Faster-RCNN were not significant.

Fire detection from images using Faster R-CNN and multidimensional texture analysis

In this paper, an approach for fire detection from images combining the merits of deep learning and spatial texture analysis was presented (Barmpoutis, Dimitropoulos, Kaza, & Nikos, 2019). The two main steps of the proposed approach are:

1. Candidate region detection using a Faster R-CNN network trained for fire detection
2. Validation of detected fire regions using analysis of spatial characteristics through linear dynamical systems (LDS).

In order to discriminate fire-colored objects and actual fire, they used Vector of Locally Aggregated Descriptors (VLAD) encoding that improves performance and significantly decreases detection errors. The approach was applied to both fire images and fire-colored objects. Experimental results show that the proposed approach retains high true positive rates, while simultaneously significantly reducing false positives due to fire-colored objects. Figure 12 shows images of fire detection with detected regions of interest (Koirala, 2020) and Table 3 shows the comparison results of the various fire detection approaches.



Figure 12: Fire detection using object detection showing detected region of interest

Table 3: Comparison results of various fire detection approaches

Method	F-Score
LDS (Martin distance)	81.5
h-LDS (Martin distance)	91.1
Grassmannian VLD encoding	93.8
SSD	92.3
YOLO v3	96.4
Faster R-CNN (Alexnet)	94.1
Faster R-CNN (Alexnet)+VLAD	95.5
Faster R-CNN(VGG16)	96.7
Faster R-CNN (VGG16)+VLAD	99.6
Faster R-CNN(Resnet101)	97.9
Faster R-CNN (Resnet101)+VLAD	99.7

Discussion

Research on previous articles and an extensive literature review was performed to find the model that was able to extract information after a precise detection of fire. This problem was addressed by implementing an object detection model (Faster R-CNN) because image classification will not be able to locate the fire and labeling in segmentation is also time consuming.

1.4 Conclusion

Deep learning techniques offer significant benefits by extracting valuable information from fire imagery and enhancing the analysis of fires in terms of behavior, spread, and intensity. This integrated approach improves fire safety by enabling the development of better fire-resistant materials, the implementation of effective fire detection systems, and the formulation of efficient fire mitigation strategies.

2 Faster R-CNN for fire detection

2.1 Introduction

In this section, the fundamental concepts of deep learning and object detection, as well as a comprehensive understanding of the Faster R-CNN is discussed.

2.2 Introduction to deep learning

Deep learning (also called hierarchical learning, deep structured learning) are types of machine learning algorithms that have enabled huge advancements in the field of artificial intelligence. In a machine learning task, the program analyzes a set of data in order to extract rules, with which it can find conclusions for new data. Deep learning is based on artificial neurons, which by definition are analogous to a biological neuron in terms of construction and function. A biological neuron receives multiple signals from other neighboring neurons and sums them, if the resulting signal is above the threshold, the neuron is activated and sends a signal to its neighbors. Following the same logic, an artificial neuron receives weighed inputs, sums them along with a bias, if the result is above the threshold of the activation function, the neuron is activated and forwards the value as an output. Figure 13 shows the analogy between a biological and artificial neuron (Muhammad, Ahmad, Mehmood, Rho, & Baik, 2018).

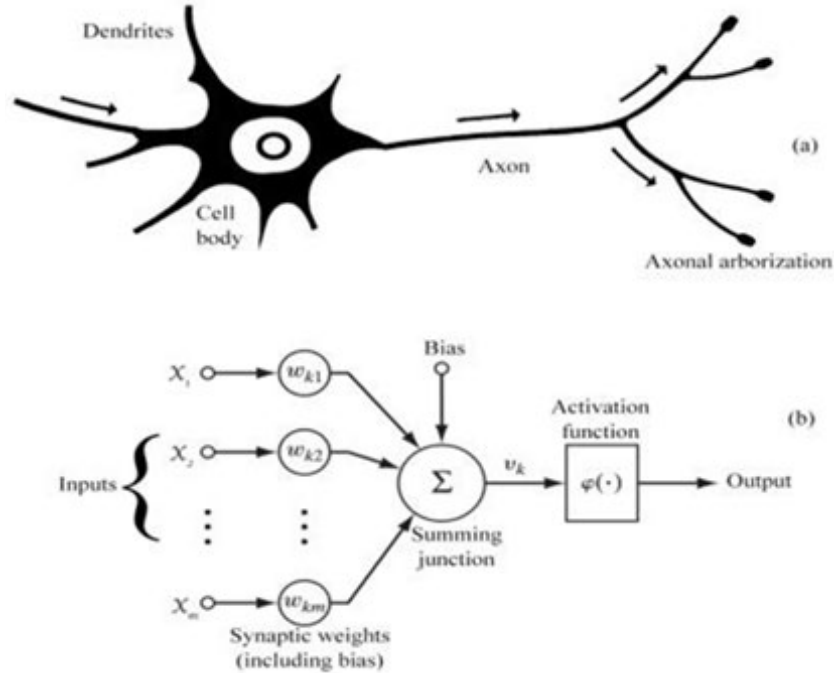


Figure 13: Analogy between biological neuron (a) and artificial neuron (b)

2.2.1 Understanding how deep learning works

The neuron is the building block of every deep learning method. In order to harvest its capacity, it must be used in multiple stacks that are called layers. We define this as a neural network. A network of neurons is composed of:

- An input layer, the first layer of every neural network and holds the data that is to be processed.
- A hidden layer (at least), responsible for performing mathematical operations on the values provided by the previous layer. A network with one or two middle layers is called a shallow network. A network with at least 3 middle layers is a deep neural network. Figure 14 shows examples of the shallow and deep neural networks.
- An output layer, displaying the result of all previous operations. The number of output neurons is equal to the number of desired classes of results.

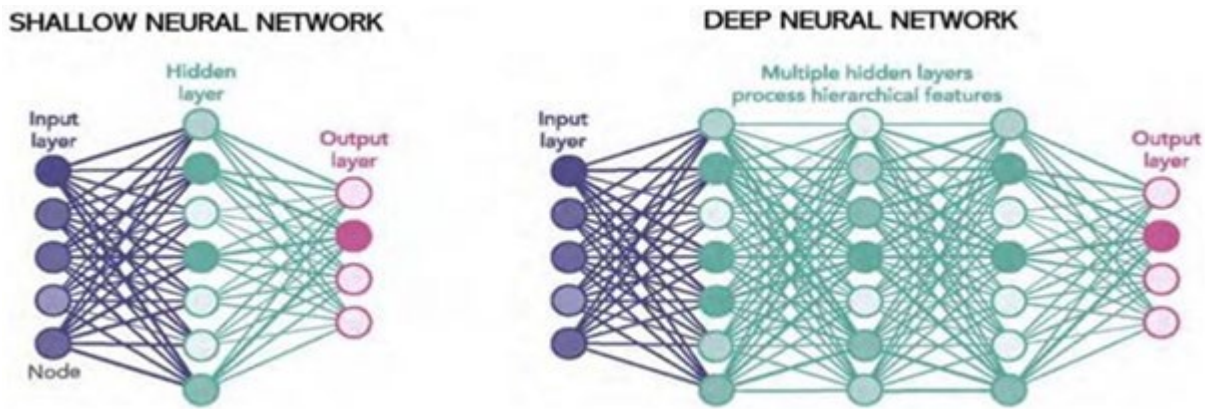


Figure 14: Examples of shallow and deep neural networks

At each neuron, the input signals are multiplied by weights. The neuron calculates the scalar product of input signals and corresponding weights and adds a bias to the result. Let $X = x_1, x_2, \dots, x_n$ an input vector, $W = w_1, w_2, \dots, w_n$ a weights vector, and b a bias, the neuron calculates the scalar product as follows:

$$S = \sum_{i=1}^n x_i w_i + b = x_1 w_1 + x_2 w_2 + \dots + x_n w_n + b \quad 1$$

The result is subjected to an activation function, which helps standardize the value. An activation function is a nonlinear mathematical operation that decides whether the neuron gets activated or not.

Some commonly used functions are:

- Sigmoid: The sigmoid, also called logistic function, is defined as:

$$S(x) = \frac{1}{1 + e^{-x}} \quad 2$$

- ReLU: The rectified linear unit (LeCun, Bengio, & Hinton, 2015) is defined as $f(x) = \max(0, x)$:

$$f(x) = 0 \text{ if } x < 0, x \geq 0 \quad 3$$

- Softmax: Softmax calculates the probability distribution of an event over n events, probabilities are used to determine the class of the input (Gold & Rangarajan, 2022). The Softmax formula is as follows:

$$f(x) = e^{x_i} / \sum_k e^{x_k} \quad 4$$

After training the network on data (called training data), a model is obtained containing the weights of neural connections. A portion of the training data is used to validate the model by comparing the output of the network to the correct data labels. This set is called the validation data. To test the accuracy of the model, another set of data, called testing data, is used. The new set contains data samples that the model has not seen before and serves to assess the performance of the model. The conventional train/test data split is 80%-20% (Barmpoutis, Dimitropoulos, Kaza, & Nikos, 2019). A deep learning model is not an immediate success. It is an iterative process that must be improved by changing its parameters, called hyperparameters.

2.2.2 Hyperparameter tuning

The learning process of a neural network can be controlled and optimized by configuring hyperparameters to optimal values. A hyperparameter is a variable that defines the structure and performance of a network. The following are the most common hyperparameters for deep learning models.

Number of hidden layers

Hidden layers define the depth of a neural network, they are important for feature learning (Barmpoutis, Dimitropoulos, Kaza, & Nikos, 2019). In general purpose neural networks, the more hidden layers there are, the better the performance gets. Yet, a potential problem of designing very deep networks is the saturation of performance, which consumes computation resources and execution time.

Activation function

Activation functions are used to determine the output of the network's layers. They are also used to add a nonlinearity aspect to the network, as the model tends to adapt better to different sets of data using nonlinear functions rather than linear functions. The choice of an activation function affects the output of the network. Sigmoid and hyperbolic tangent (tanh) are better suitable for binary classification. Softmax is more used for multiclass classification, and the rectified linear

unit (ReLU) helps for cases of multiclass classification with interactive effects where the effect of a variable depends on other variables (Elliot, 1992).

Loss function

The loss function calculates the error between the actual class of a sample and the predicted value by the model. The mean values of loss function over the entire sample database are called the cost function. The loss function is extremely important for evaluating and optimizing deep neural networks. Figure 15 shows how loss function measures the quality of the network's output and Figure 16 shows how loss score is used as a feedback signal to adjust the weights.

Optimization algorithm

This hyperparameter tries to find a set of intrinsic parameters that perform well with performance metrics (mean quadratic error, log error, etc.), using gradient descent calculus. An optimization algorithm is an iterative process that uses parameters to make predictions on data samples, compare the predicted result to the actual data class, calculate the prediction error, and update the parameters.

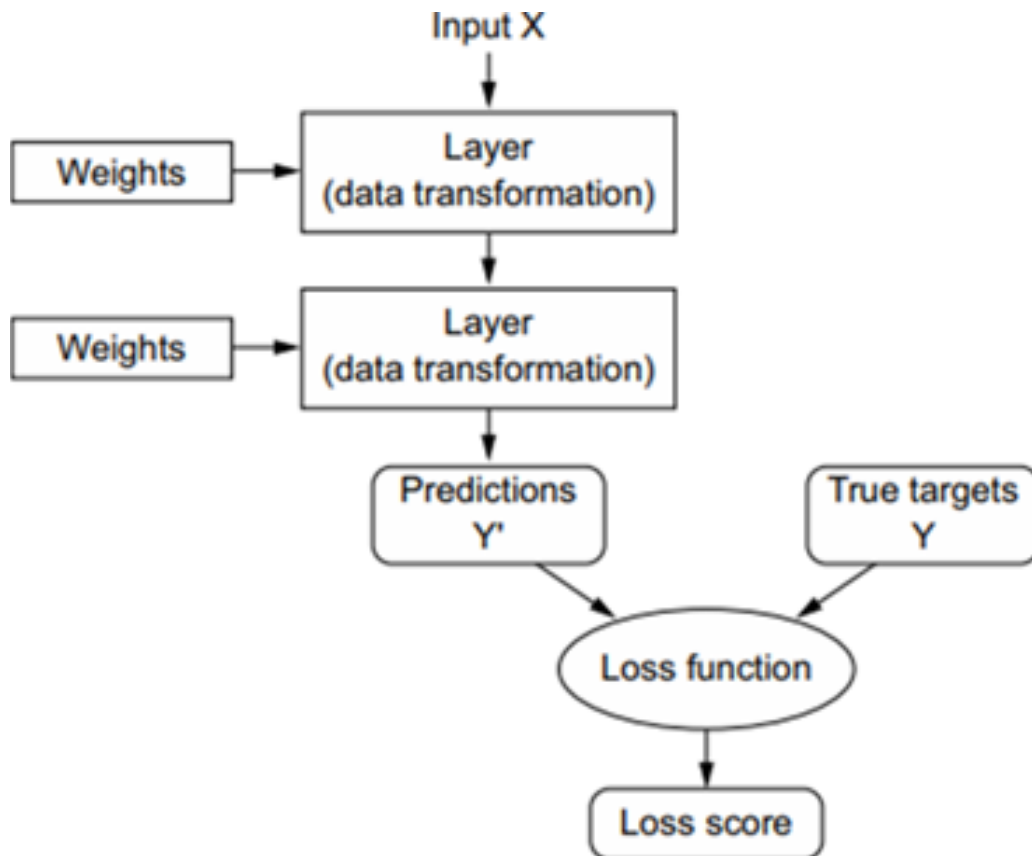


Figure 15: Loss function measures quality of network's output

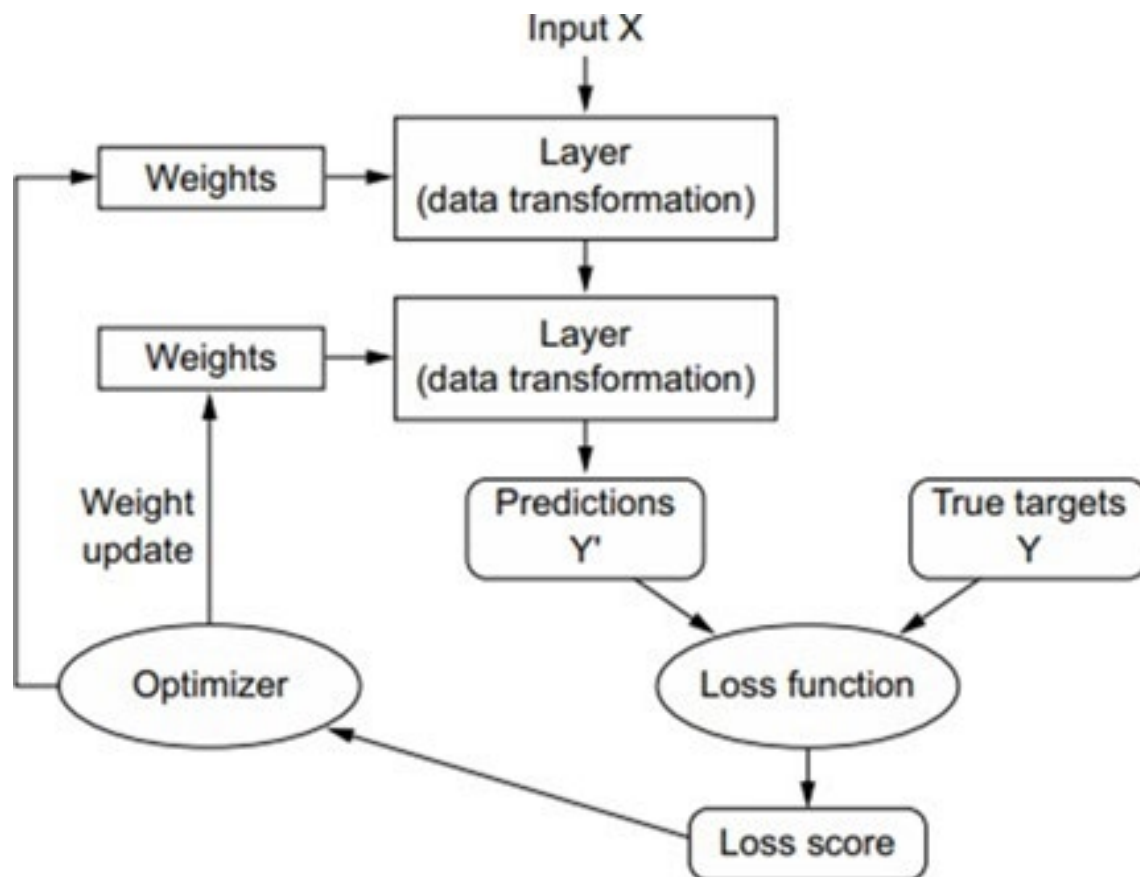


Figure 16: Loss score used as feedback signal to adjust weights

Learning rate

The learning rate defines the speed of convergence of the optimization algorithm towards the optimal minimum of the loss function. A high number gets the algorithm to converge faster but is less precise. A low number means a slower convergence rate but provides a better guarantee for global minima searching.

Number of epochs

The number of epochs is a hyperparameter that defines the number of times that the learning algorithm circles around the input data.

Batch size

The batch size is the number of samples that are processed every time the model is updated. The batch size should be between one and the number of samples in the database.

Convolution Neural Network (CNN)

The oldest and greatest success story of deep learning is computer vision. Google Photos, Google image search, YouTube, video filters in camera apps, OCR software, and many other services let users engage with deep vision models on a daily basis. Additionally, these models serve as the foundation for cutting-edge research in robotics, AI-assisted medical diagnosis, self-operating retail checkout systems, and even autonomous farming. The issue space that sparked the initial growth of deep learning between 2011 and 2015 was computer vision. Convolutional neural networks, a subset of deep learning models, began to perform extremely well in image classification contests around that same time frame (Chollet, 2019).

A CNN is composed of two parts:

- A convolution network is a succession of filters that are applied to the input image, resulting in new images called feature maps, with the goal to extract features from input images. At the end of the convolutional network, feature maps are concatenated to obtain a feature vector.
- A classification network comprises fully connected layers to classify images, based on the feature vector provided by the previous network (Alzubaidi L. , Zhang, Humaidi, & Al-Dujaili, 2021). Refer to Figure 17 for the CNN architecture.

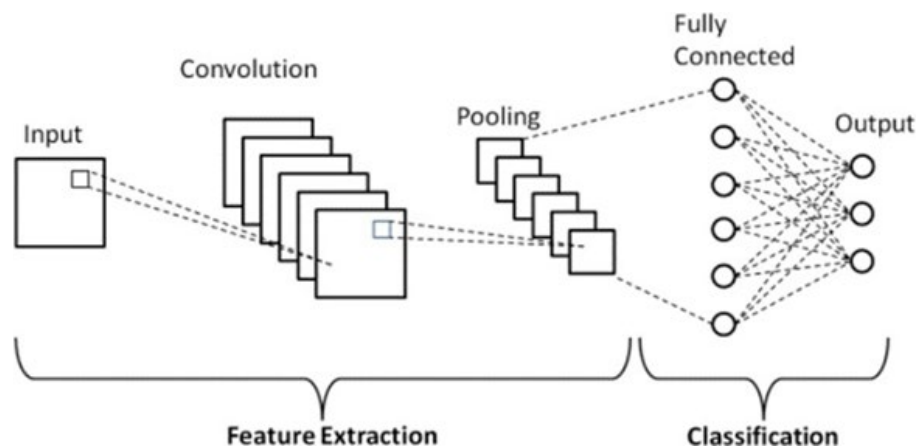


Figure 17: Convolutional neural network architecture

CNNs are structured as follows:

- The input layer is where images are formatted to tensors and used as an input to the CNN.
- The convolution layer detects features and locates them on the image using convolutional filters. The user chooses a filter that will be applied to the image. The filter slides on the

image matrix following a parameter called stride (number of steps along one direction) and performs a convolution calculus. The output is a feature map that contains extracted information.

- A dense layer learns global patterns in its input feature space, whereas a convolution layer learns local patterns. This is the key distinction between a dense layer and a convolution layer.
- The pooling layer is responsible for reducing the dimensions of feature maps while retaining their relevance. Max pooling and average pooling are the most used techniques. Max pooling retains the maximum value of each window of the feature map and average pooling calculates the mean value of each window. This layer helps reduce model parameters and computational resources.
- The fully connected layer is responsible for the classification, where each neuron is connected to all previous and following neurons. The number of output neurons is equal to the number of classes, each neuron outputs the class probability of the input image (Elliot, 1992). Refer to Figure 18 for images broken into local patterns.

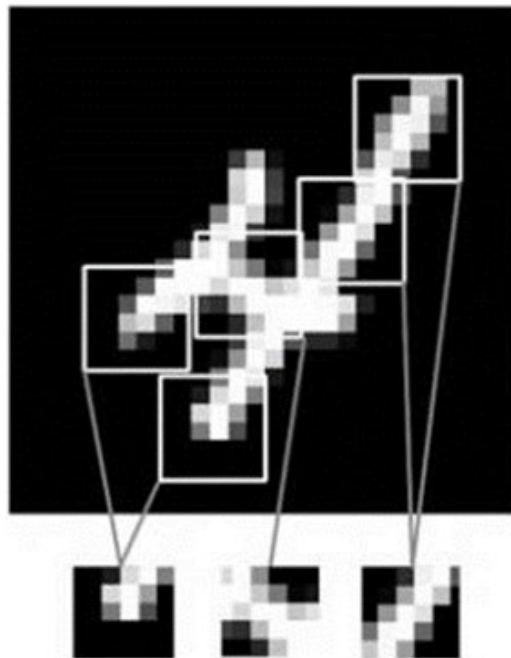


Figure 18: Images broken into local patterns (edges, textures)

2.2.3 Evaluation metrics

To assess the performance of a neural network, we use a set of indicators:

Confusion matrix

The confusion matrix is an important metric that describes the complete performance of the model. For the example of a binary classification task, it provides 4 values:

- True Positives (TP): positive prediction, positive class
- True Negative (TN): negative prediction, negative class
- False Positive (FP): positive prediction, negative class
- False Negative (FN): negative prediction, positive class (Visa, Ramsay, Ralescu, & van der Knapp, 2011)

Accuracy

Accuracy is the rate of correct predictions of all performed predictions.

$$Accuracy = TP + TN / TP + TN + FP + FN \quad 5$$

Precision (Positive Predictive Value)

Precision is the rate of correct positive prediction of all positive predictions.

$$Precision = TP / TP + FP \quad 6$$

Recall (Sensitivity)

Recall, also called true prediction rate (TPR), is the rate of correct positive predictions of all correct predictions.

$$Recall = TP / TP + FN \quad 7$$

F1 Score

F1 score is the harmonic mean between precision and recall values, ranging between 0 and 1.

$$F1 \text{ score} = 2 * 1 / 1 + 1 \text{precision recall} \quad 8$$

2.3 Object detection

Object detection, regardless of whether performed via deep learning or other computer vision techniques, has three primary goals we wish to obtain from a given input image:

- A list of bounding boxes, or the (x, y)-coordinates for each object in an image.
- A class label associated with each bounding box.
- The probability/confidence score associated with each bounding box and class label.

2.3.1 Measuring object detector performance

In Figure 19, the anticipated bounding box is depicted in red, whereas the ground-truth bounding box is depicted in green (Zhao, 2019). Our objective is to determine where these bounding boxes intersect. The area of overlap between the bounding boxes must be divided by the area of union to get the intersection of union (IoU).

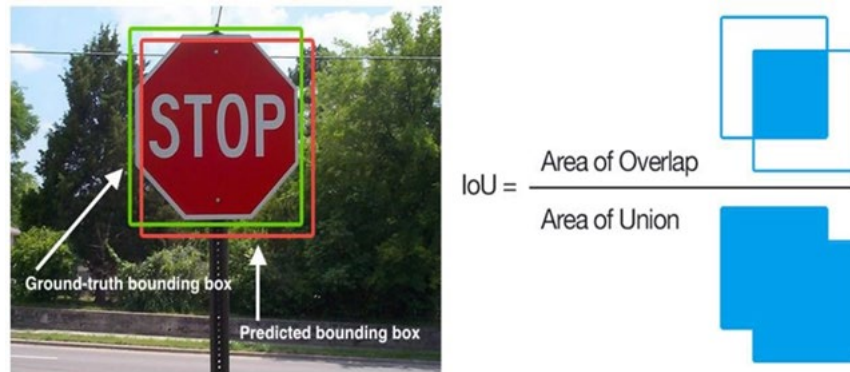


Figure 19: Example of detecting a stop sign

In order to apply IoU to evaluate an arbitrary object detector, we need:

- Ground-truth bounding boxes (hand labeled bounding boxes from the testing set that specifies where in an image the object of interest is).
- Predicted bounding boxes from our model.

Figure 20 shows the comparison of computing IOU for bounding boxes (Li & Zhao, 2020). In order to compute recall along with precision, the ground-truth class labels and predicted class labels are also needed.



Figure 20: Comparison of computing IOU for bounding boxes

Because coordinates will not match exactly, an evaluation metric that rewards predicted bounding boxes for heavily overlapping with the ground-truth need to be defined.

Mean Average Precision (mAP)

The IoU of all the data points for a specific class is calculated in order to get the average precision for that class. When the IoU is divided by all of the class labels for that particular class, the average precision is obtained. The term "mean average precision" refers to the result of computing the average IoU for all n classes, followed by taking the average of these n averages.

2.4 Faster R-CNN architecture

Figure 21 shows an example of the R-CNN architecture (Muhammad, Ahmad, Mehmood, Rho, & Baik, 2018). To better understand how the Faster R-CNN object detection algorithm works, the history of the R-CNN algorithm, including its original incarnation and how it has evolved needs to be reviewed.

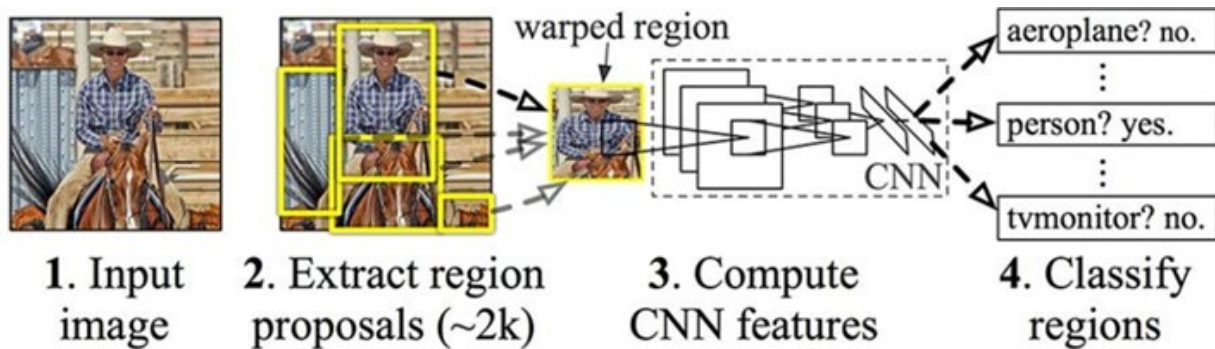


Figure 21: R-CNN architecture example

The original R-CNN architecture consisted of:

- Accepting an input image
- Extracting $\approx 2,000$ proposals via the selective search algorithm
- Extracting features for each proposal region of interest (ROI) using a pre-trained CNN (such as one trained on ImageNet)
- Classifying the features for each ROI using a class specific linear SVM

The algorithm is given a starting image in the first stage. The region proposal algorithm is then used in a selective search. Sliding windows and picture pyramids are replaced with the Selective Search algorithm, which intelligently examines the input image at numerous scales and locations. As a result, there are significantly fewer proposal ROIs that need to be given to the network for categorization. Selective search can therefore be compared to an intelligent sliding window and image pyramid technique. Once the proposed locations are obtained, each one is cropped from the input image separately and feature extraction used to apply transfer learning. Feature

extraction is used to enable a downstream classifier to learn more discriminating patterns from these CNN characteristics rather than getting the final predictions from the CNN.

The fourth and last phase entails training a number of SVMs for each class using these extracted features. The original R-CNN method has the drawback of still being exceedingly slow. Additionally, a deep neural network to learn how to locate is not actually being used. The question of whether it is possible to achieve end-to-end deep learning-based object detection arises because it's leaving localization to the selective search algorithm and only classifying the ROI once it has been deemed "interesting" and "worth examining" by the region proposal algorithm.

While selective search is still used to generate suggestions in the Fast R-CNN architecture, ROI pooling is now used to generate the final class label and bounding box by removing a fixed-size window from the feature map. In this new method, we run CNN on the entire input image and then use our network to extract a feature map from it. In order to generate the output label for the ROI, ROI pooling first extracts a fixed-size window from the feature map and then passes it into a group of fully- connected layers. Despite being end-to-end trainable, the network's performance at inference time deteriorated greatly due to its reliance on the selective search (or comparable) region proposal method. To make the R-CNN architecture even faster, the region proposal directly into the R-CNN needed to be incorporated.

The newest member of the R-CNN family, Faster R-CNN, incorporates a region proposal network (RPN), which eliminates the requirement for the Selective Search algorithm by including region proposals directly into the design. Where a probable object might be in a picture is determined using the first component, the RPN. Currently, all it knows is that there might be an object at a specific spot in the photograph but does not know what it is. Figure 22 shows the Fast R-CNN architecture example and Figure 23 shows the Faster R-CNN example (Muhammad, Ahmad, Mehmood, Rho, & Baik, 2018).

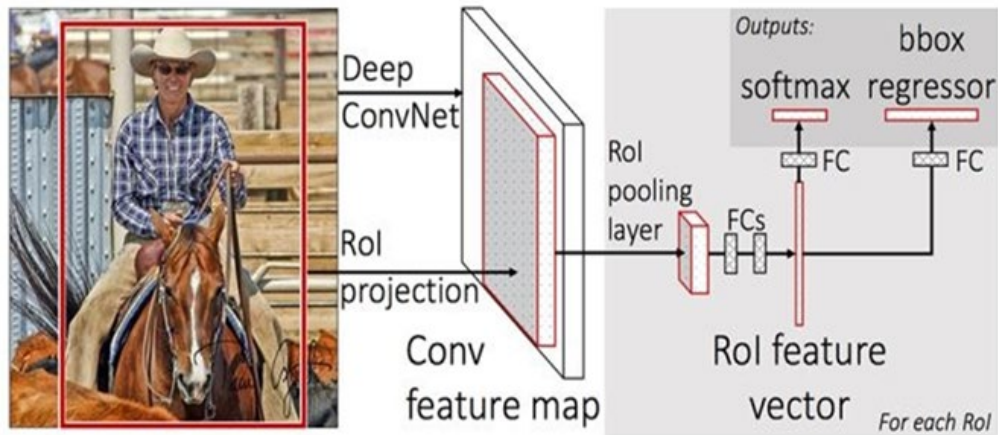


Figure 22: Fast R-CNN architecture example

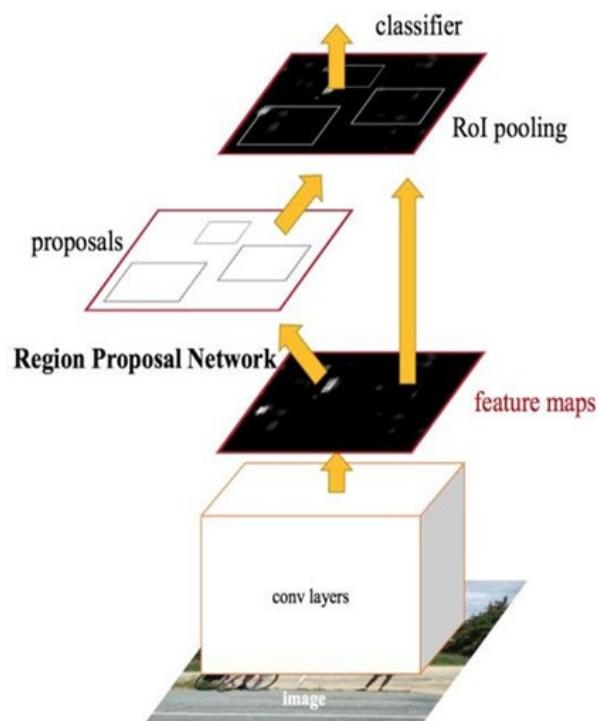


Figure 23: Faster R-CNN example

The network's region of interest (ROI) Pooling module and the features that were recovered in the previous stage constitute the foundation for the suggested bounding box ROIs. To extract

fixed-size windows of features, ROI pooling is utilized. These windows are then transferred into two fully connected layers to produce the final localizations.

Evaluating faster R-CNN:

- `total_loss`: This is a weighted sum of the following individual losses calculated during the iteration. By default, the weights are all one.
- `loss_cls`: Classification loss in the ROI head. Measures the loss for box classification; how good the model is at labelling a predicted box with the correct class.
- `loss_box_reg`: Localization loss in the ROI head. Measures the loss for box localization (predicted location vs true location).
- `loss_rpn_cls`: Classification loss in the region proposal network. Measures the "objectness" loss; how good the RPN is at labelling the anchor boxes as foreground or background.
- `loss_rpn_loc`: Localization loss in the region proposal network. Measures the loss for localization of the predicted regions in the RPN (Rosebrock, 2017).

2.5 Conclusion

In conclusion, deep learning and object detection techniques have emerged as the state-of-the-art approaches in various fields, including computer vision, image processing, and pattern recognition. Deep learning models, particularly convolutional neural networks (CNNs), have revolutionized the field by exhibiting remarkable capabilities in detecting and localizing objects within images and videos.

3 Approach

3.1 Introduction

In this section, an in-depth analysis based on the use of Faster R-CNN for this approach in fire detection and quantifying is presented. This section focuses on fire test videos from the FAA VFP apparatus. The videos were of aircraft materials in several different configurations. The videos from the tests were not optimized for extracting test parameters from the video record. Bad angles, focus and image obscuration made the analysis more difficult but not impossible.

3.2 Data preparation

Data preparation is a crucial step in machine learning. It allows a more efficient analysis of the situation by lessening precision errors and facilitating the learning process and includes early cleaning and transformation of raw data samples before processing and using them for the model. The process of data preparation encompasses operations such as format standardization, data enrichment, and outlier removal.

3.2.1 Data collection and discovery

The data in this study consisted of 23 videos and after frame extraction with FPS=2 (the best frame rate to monitor fire) and yielded 3811 images in which 1242 images contained fire that was annotated manually by drawing the bounding boxes and labeling them for each frame using the LabelImg tool. Figure 24 shows manual annotation using the LabelImg tool.

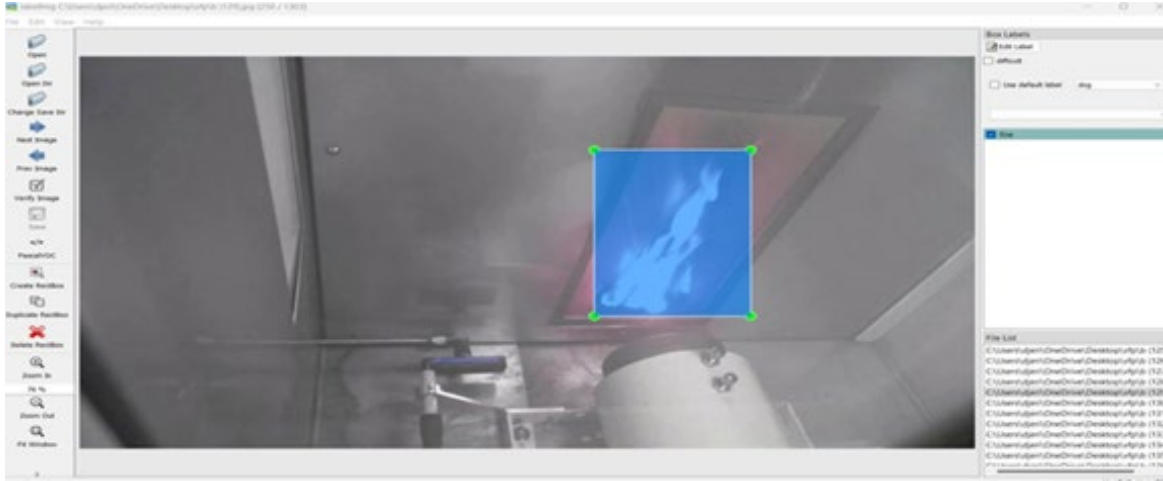


Figure 24: Manual annotation using the LabelImg tool

3.2.2 Data structuring and preprocessing

The RoboFlow was used as the platform to split, preprocess and export the data. Once the data is annotated, RoboFlow offers a range of preprocessing capabilities to augment and enhance the data. These preprocessing techniques include resizing, cropping, rotation, and more, which are essential for optimizing the training process and improving model accuracy.

Data resizing and splitting

The first step after finishing labeling our data was resizing it to the same size to make it simpler and faster for the model to process. For an accurate evaluation for our results, we proceed to splitting it to training, validation, and testing datasets, shown graphically in Figure 25.



Figure 25: Data splitting example

- Training DATASET: The data from which the model learns and extract the features of their labels.
- Validation DATASET: The data used to monitor the model and modify its hyperparameters.
- Testing DATASET: The data used for inference.

Data augmentation

- Blur: Helps to solve the problem of bad quality images.
- Brightness: Helps to solve the problem of heterogeneous data and reflection.
- Rotation: Helps to solve the problem of different camera settings.

Combining those filters managed to augment the training dataset to more than 2600 samples which helped to solve the problem of limited data (see Figure 26 for preprocessing data and accounting for interferences).



Figure 26: Preprocessing data and accounting for interferences

3.3 Building the model

The architecture of the backbone we want to use and experiment with was chosen first, for example, `faster_rcnn_R_50_FPN`: Resnet feature pyramid network backbone architecture with 50 layers. If the wrong architecture was chosen, the model can be forced to use metrics that aren't ideal and no amount of training data will be able to fix that. In contrast, a strong model architecture will quicken learning and let the model utilize the given training data effectively. This minimizes the need for massive datasets. A good model architecture is one that makes it simpler to converge on a good point in the search space or otherwise minimizes the size of the search space. The goal of model design, like that of feature engineering and data curation, is to simplify the problem such that gradient descent can solve it. Figure 27 shows the data augmentation and Figure 28 is an example of the Resnet 50 FPN architecture.

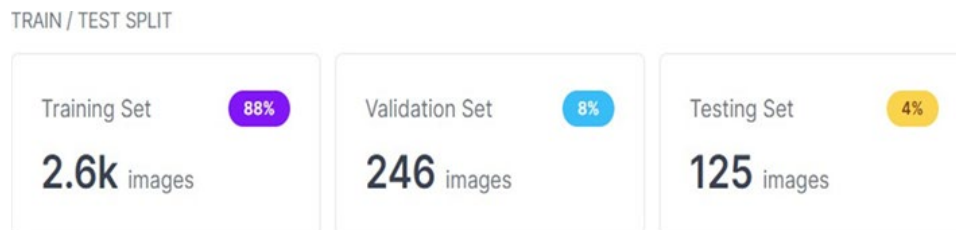


Figure 27: Data augmentation

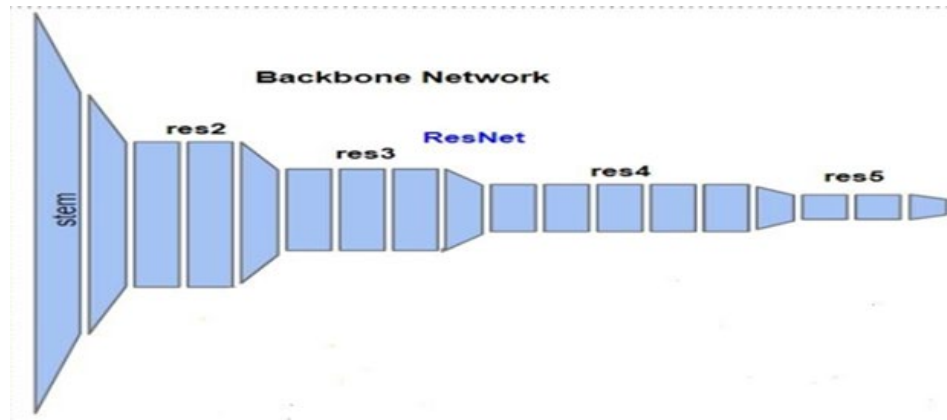


Figure 28: Resnet 50 FPN architecture example

Most convnets often feature pyramid-like structures (feature hierarchies). The number of filters grows with layer depth, while the size of the feature maps shrinks accordingly. Deeper hierarchies are beneficial in and of themselves because they promote feature reuse and, by extension, abstraction. Generally speaking, a deep stack of thin layers outperforms a shallow stack of thick layers. However, the issue of vanishing gradients places a limit on how many layers can be stacked. Remaining connections are our first fundamental model architecture pattern (Alzubaidi L. , et al., 2021).

Residual connections

In order to allow error gradient information from early layers to pass noiselessly through a deep network, the residual connection functions as an information shortcut around noisy or destructive blocks (such as blocks that contain ReLU activations or dropout layers), see Figure 29. The ResNet family of models debuted this technique in 2015.

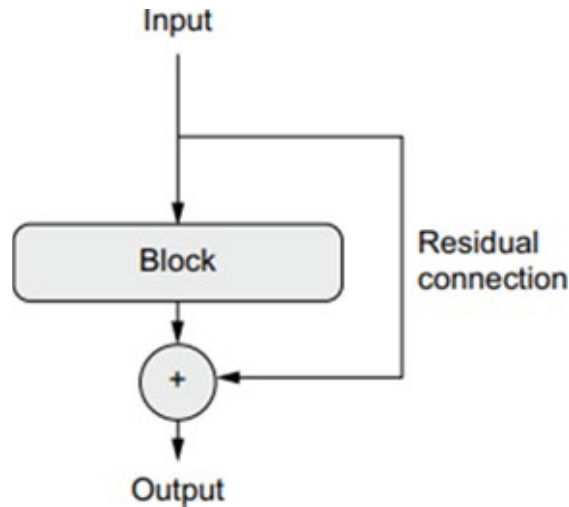


Figure 29: Residual connection around a processing block

The default trainer data set is imported from the Detectron engine module and the hyperparameters including the number of workers, batch size, and classifications are set. The model is then trained with pre-trained weights as its foundation. The scale of the dataset and the difficulty of the task can affect the max iterations parameter. Each time a deep learning model is trained, its final checkpoint is saved and loaded quickly to make predictions and draw conclusions.

3.4 Information extraction

To extract the measurement, the object detection algorithm (the bounding boxes coordinates, their number and their label) and some image processing techniques (changing channels, thresholding and cropping) were used.

3.4.1 Object detection algorithm

Time of Burning: The time of detection of the first frame with the bounding box of fire (see example in Figure 30).

Let L: list of the nbr of Bboxes in each frame in the video and FPS= 30: The frame rate of the camera that took the video

$L=[0,0,0,0,0,0,1,1,1,2,2,2,2,2,3,3,3,3,3,3,3,3,3,3,1,1,1,1,1,1,0,0,0,0,0,0,0]$

$\text{Time} = 7 * (1/\text{FPS}) = 7/30 = 0.2333\text{s}$ where 7 is the position of frame with the first fire detection.



Figure 30: Time of burning extraction

Burn Length: The number of frames with bounding boxes multiplies by the frame rate FPS (see example in Figure 31).

Let L: list of the nbr of Bboxes in each frame in the video and FPS= 30

$L=[0,0,0,0,0,0,1,1,1,1,2,2,2,2,2,3,3,3,3,3,3,3,3,3,3,1,1,1,1,1,1,0,0,0,0,0,0,0]$

$\text{Length} = 300 * (1/\text{FPS}) = 300/30 = 10\text{s}$ where 300 is the number of frames with fire detection.

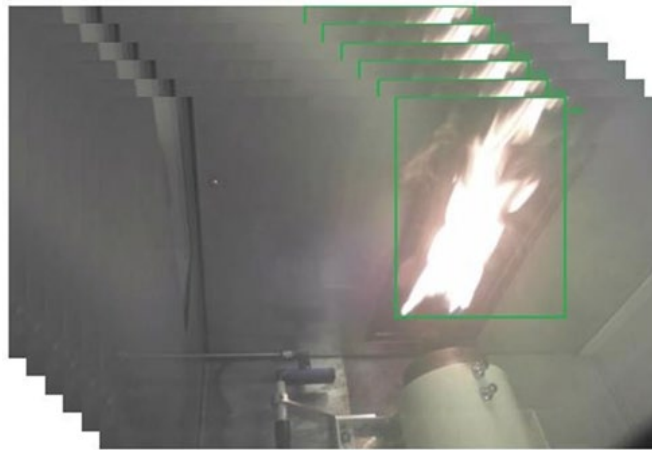


Figure 31: Images representing the burn length extraction

Number of Droplets: The number of bounding boxes in the frame which have the most boxes in the video, or number of droplets, is considered (see example in Figure 32).

Let L: list of the nbr of Bboxes in each frame in the video and FPS = 30

$L = [0,0,0,0,0,0,1,1,1,1,2,2,2,2,2,3,3,3,3,3,3,4,3,3,3,1,1,1,1,1,1,0,0,0,0,0,0,0]$

$\text{Nbr} = \max(L) = 4$

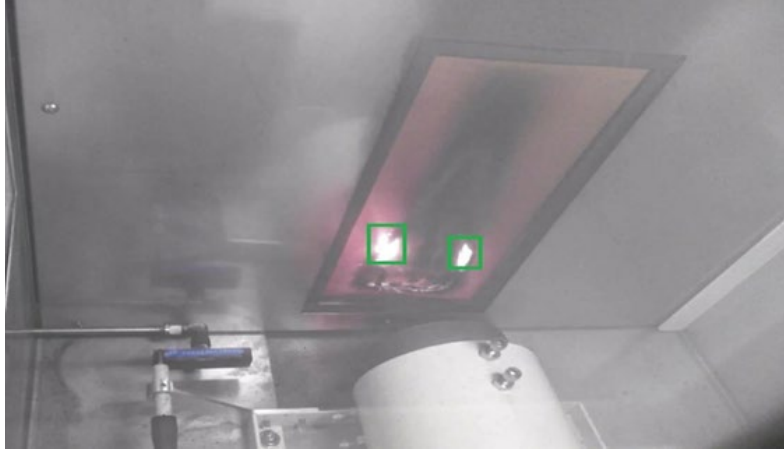


Figure 32: Bounding boxes showing the number of droplets extraction

Burn Scale: The surface of the biggest bounding box with fire calculated by multiplying the width and height of that bounding box (see example in Figure 33).

$$\text{Bbox_Surface} = H * W$$

L1: list of Bbox_Surface in single frame scale = max(L1)

L2: list of scales for all the frame in a video scale = max(L2)

$$\text{Burn_Scale} = \text{scale}/\text{Frame_Surface}$$

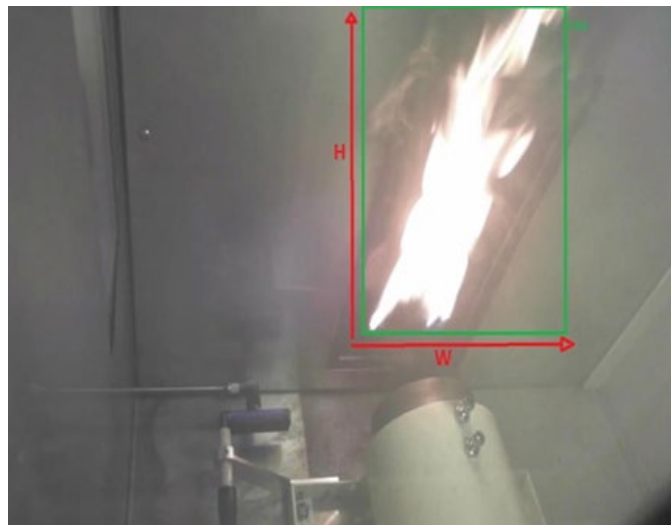


Figure 33: Burn scale extraction from images

3.4.2 Image processing

Flame intensity

Converts frames from RGB to HSV and extracts the frame with the max value in the video (see Figure 34). The max value is the flame intensity (Chang, 2002). The color value refers to the relative lightness or darkness of a color and is perceived based on the quantity of light reflected off of a surface and absorbed by the human eye.

Charring area

The last frames are converted from RGB to greyscale and crop the frame. A threshold on the cropped frame and the number of black pixels is then applied to the charring area (Singh, 2019). Because the fire propagation direction can't be controlled, the charring area changes from video to video, so the operator needs to enter additional inputs other than the raw video. Figure 35 shows charring area extraction from end-of-test frames and Figure 36 shows Extraction of results from additional inputs defining a region.



Figure 34: Flame intensity extraction by converting images to HSV



Figure 35: Charring area extraction from end-of-test frames

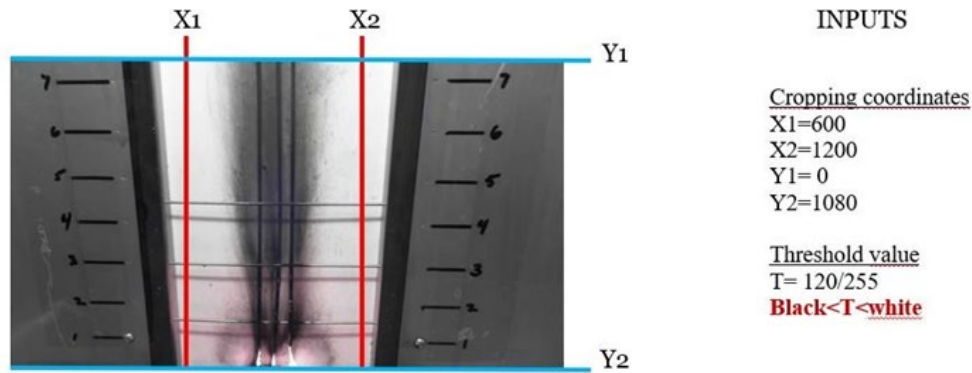


Figure 36: Extraction of results from additional inputs defining a region

3.5 Conclusion

In this section, a method for processing images based on cropping, filtering, and augmentation was proposed. Therefore, allowing a better edge definition and noise removal, and proposed a customized object detection model and image processing methods for the purpose of fire detection and quantifying. In the following section, the effect of preprocessing on the accuracy of classification was experimentally evaluated.

4 Results and discussion

4.1 Introduction

This section details the model deployment and performance assessment. The effect of preprocessing techniques on the efficiency of the CNN backbone model, along with the analysis and interpretation of the results were examined.

4.2 Environment

Colab and Detectron2 were chosen as the best environment for their extensive resources and reliability.

4.2.1 Hardware environment: Google Colaboratory

The object detection model and feature extraction algorithms were developed using Python 3.7 and trained on the Google Colab platform, an online Python editor for machine learning and education with a dedicated K80 GPU (16GB RAM). Python provided a diverse set of libraries for machine learning, which were utilized for this work.

4.2.2 Software environment: Detectron2

The Detectron object identification platform has grown to become one of Facebook AI Research's (FAIR) most frequently used open-source initiatives since its launch in 2018. Detectron2 represents detected objects optimally for further analysis and good understanding, and is useful for detection and localization of objects quickly. The Detectron2 has higher training speeds up to 60 images per second compared to all the other open-source frameworks. The mmdetection based on PyTorch and tensorpack from tensorflow V1 is an example (Figure 37).

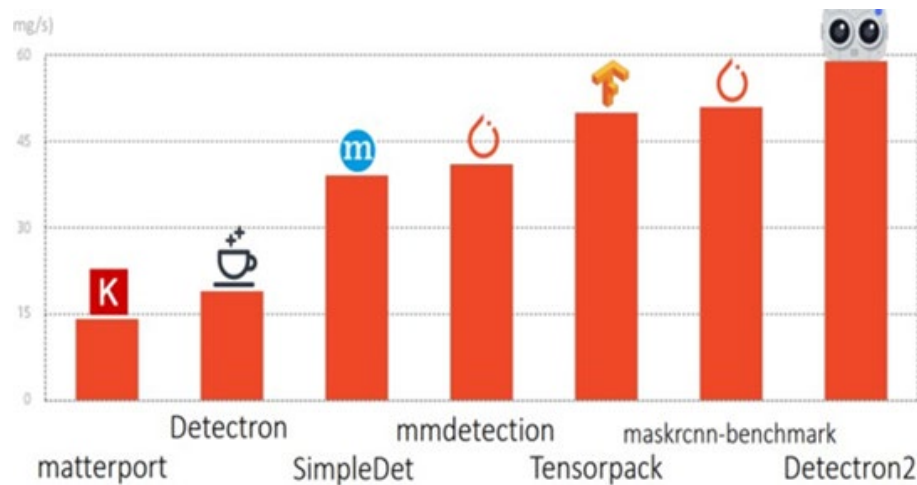


Figure 37: Training of R50_FPN

From its model zoo (Figure 38), Detectron2 gives many favors like:

- Different settings for users' interface
- Standard baselines for researchers
- Efficient models for production

Faster R-CNN:

Name	lr sched	train time (s/iter)	inference time (s/im)	train mem (GB)	box AP
R50-C4	1x	0.593	0.110	4.8	35.7
R50-DC5	1x	0.380	0.068	5.0	37.3
R50-FPN	1x	0.210	0.055	3.0	37.9
R50-C4	3x	0.589	0.110	4.8	38.4
R50-DC5	3x	0.378	0.073	5.0	39.0
R50-FPN	3x	0.209	0.047	3.0	40.2
R101-C4	3x	0.656	0.149	5.9	41.1
R101-DC5	3x	0.452	0.082	6.1	40.6
R101-FPN	3x	0.286	0.063	4.1	42.0
X101-FPN	3x	0.638	0.120	6.7	43.0

Figure 38: Detectron2 model zoo

4.3 Results

The performance of the object detection model played an essential role in the overall performance of this work and led to the final decision of the approach for fire feature extraction.

4.3.1 Object detection results

The fire detection model was optimized by exploring different architecture options and selecting the one that yielded the best results. The choice of architecture plays a crucial role in determining the model's ability to extract relevant features and accurately detect fires in VFP videos. Once the optimal architecture was determined, image processing techniques were applied to further refine the results. Image processing techniques, such as normalization, resizing, and noise reduction, were employed to enhance the quality and clarity of the input images. By standardizing the images and reducing noise, the model's ability to extract relevant features and accurately detect fires is improved.

In addition to image processing, data augmentation techniques to augment and diversify the training dataset was incorporated. By generating additional training samples through techniques such as rotation, scaling, flipping, and filtering, it introduces variability and increased the model's ability to generalize to different fire scenarios. Data augmentation enabled the model to learn from a wider range of fire instances, thereby improving its robustness and reducing the risk of overfitting.

Starting point

After analyzing the literature review and experimenting with some models using various Hyperparameters, Faster R-CNN_R50_FPN using batch size = 128, a learning rate = 0.00025, iteration number = 1000, and transfer learning from the known COCO-dataset was chosen. Applying these values to the raw data yielded a good starting point with a mAP = 93.88. See Figure 39 for optimization results on raw data.

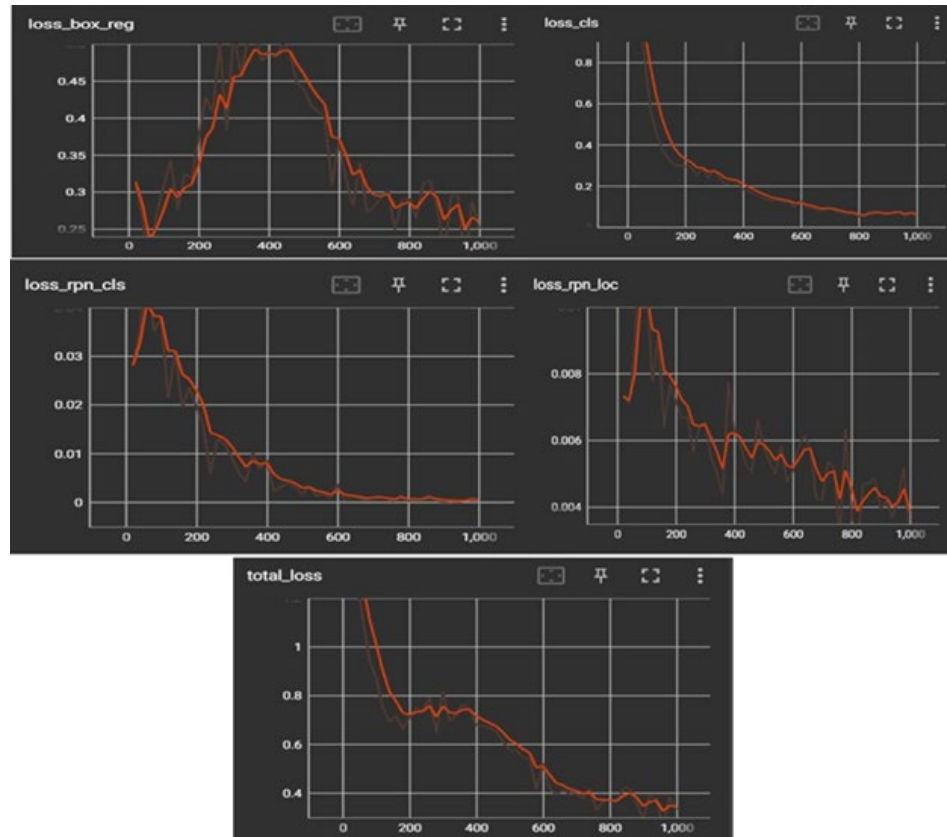


Figure 39: Optimization results on raw data

Image processing and augmentation

Rotation: Using the same model, hyperparameters and transfer learning but adding random rotation and augmenting, the data *3 yielded a mAP = 85.04.

This resulted in the average precision decreasing. After analyzing the graphs, it was concluded that most of the flammability test videos are similar and the location of the fire is approximately the same. The model focuses on the location of the bounding boxes (coordinates) as a feature as the effect of the rotation on the `loss_box_reg` graph increased. See Figure 40 for optimization results on rotated data.

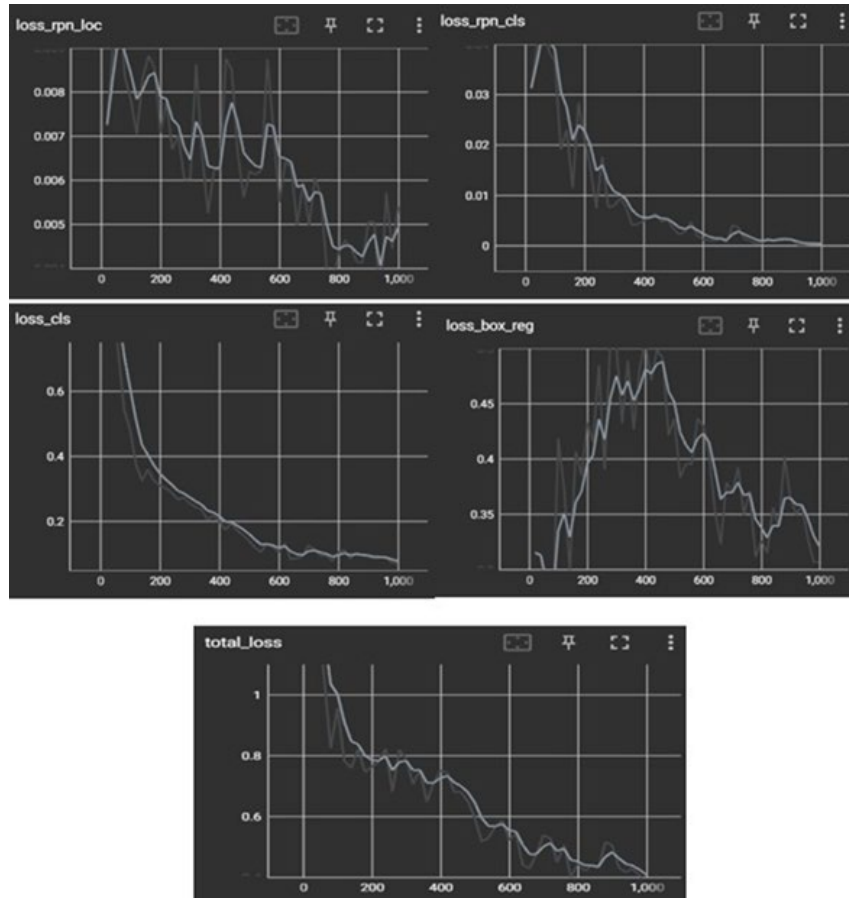


Figure 40: Optimization results on rotated data

Brightness: A similar configuration to the previous experiment was used but a random brightness was applied and augmented the data by 3 to get a mAP= 93.08.

Similar to the previous experiment, the average precision slightly decreased as the `loss_rpn_loc` slightly increased. It was concluded that because the testing environment is controlled and has a fixed lighting, the variation of the light in the video is due to the fire itself and applying a random

light will make it hard for the model to locate the fire. See Figure 41 for brightness optimization results.

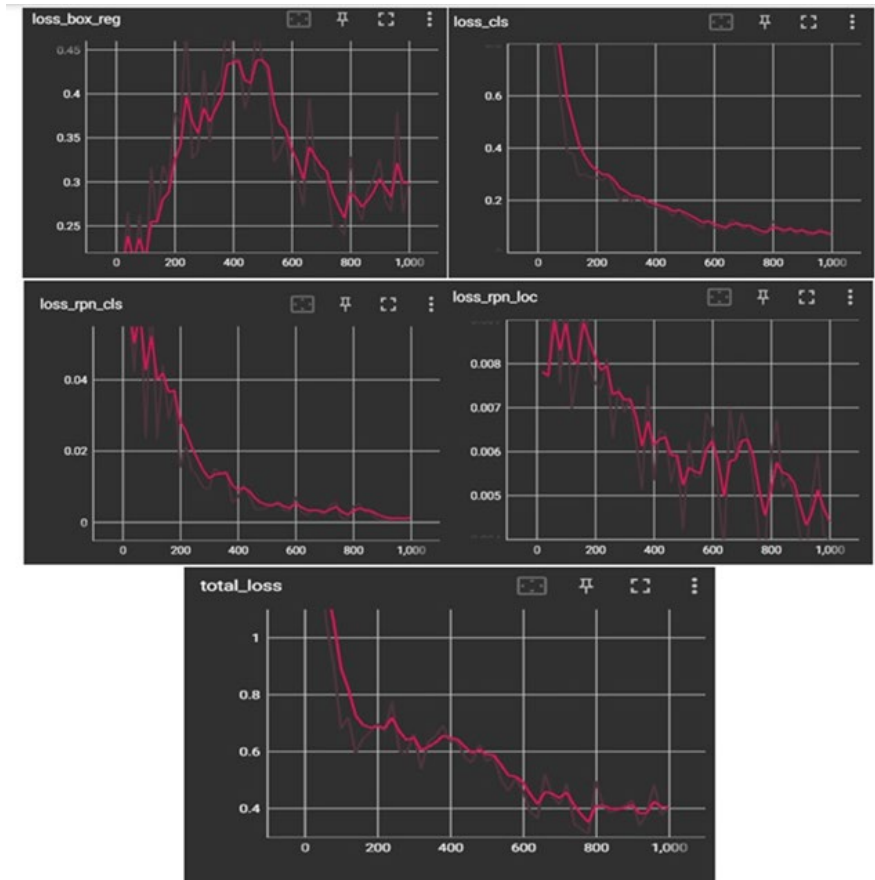


Figure 41: Brightness optimization results

Blur: Taking the raw data and the same configuration while applying a gaussian blur filter and data augmentation *3 yielded a mPA = 94.08

Adding a blur filter worked to enhance the results. The augmentation results for the brightness and rotation were confirmed with another flammability test with different lighting and camera settings. These settings yielded better results than the model with no data processing or augmentation. See Figure 42 for the total loss curve for blur filter and Figure 43 for examples of inference from other data.

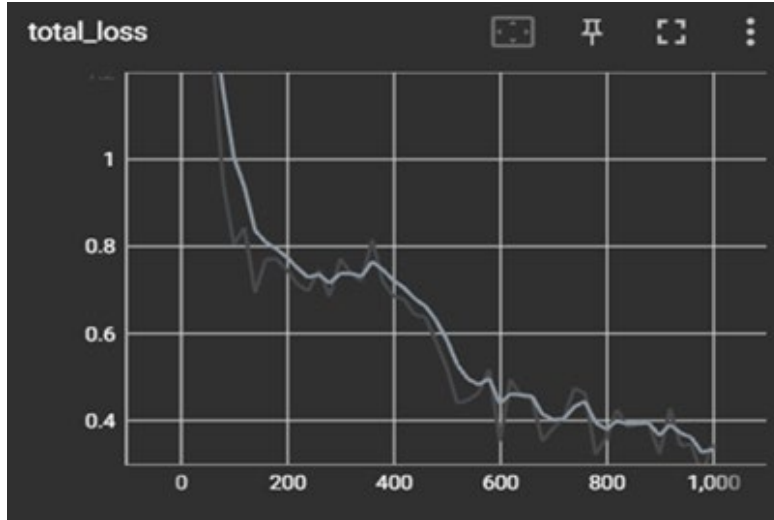


Figure 42: Total loss curve for the blur filter

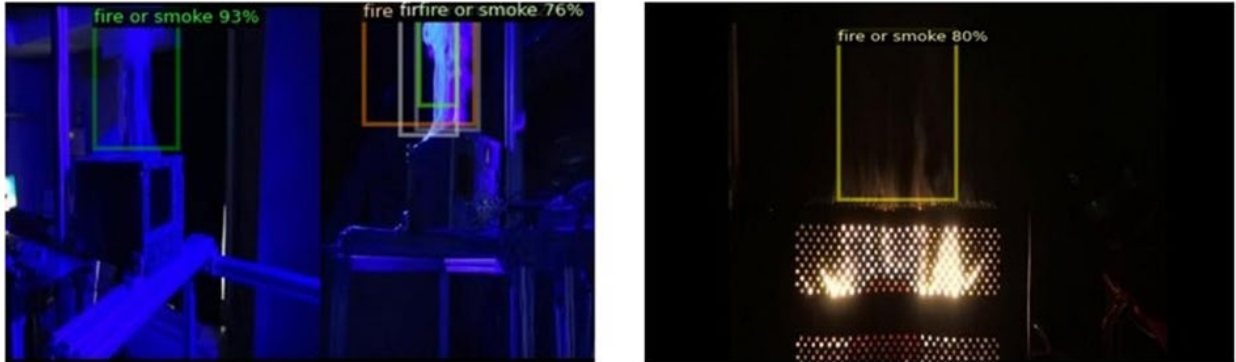


Figure 43: Examples of inference from other data

This confirmed the earlier conclusions that those models generalize better, specially to detect fire in a non-controlled environment, but for this study, the model with the blur filter was used and further refined.

Deeper architecture

The data with the blur filter gave the best results on the R_50_FPN. A similar configuration was used but switched to the R_101_FPN as the backbone with 101 layers to extract more precise features to obtain a mAP = 93.97. The total loss with this architecture was still decreasing after 1000 iteration so it was retrained for 2000 iterations and yielded a mPA = 96.44. This result showed promise and enabled proceeding to the feature extraction step. See Figure 44 for 1000 steps loss vs 2000 steps loss.

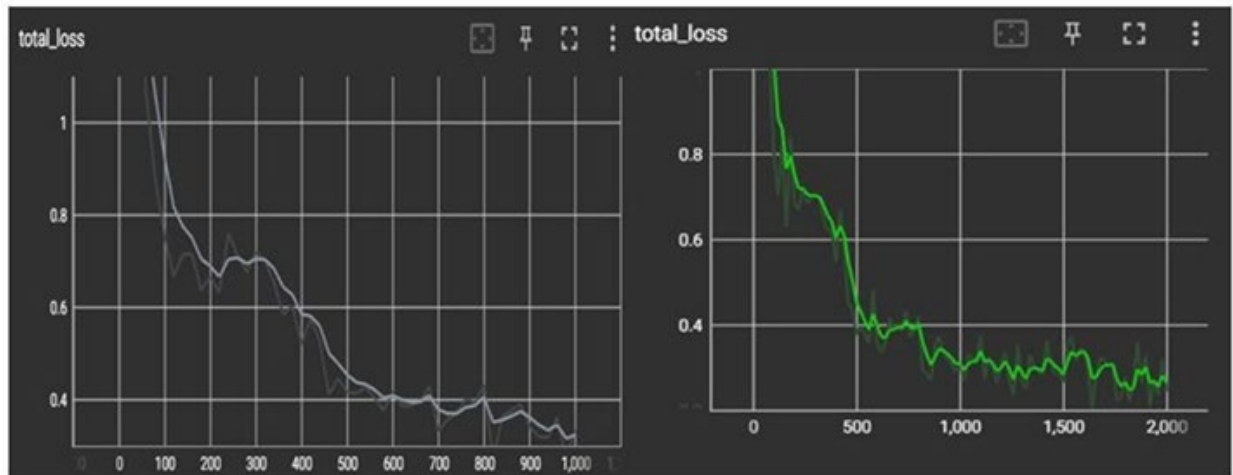


Figure 44: 1000 steps loss vs 2000 steps loss

4.3.2 Feature extraction results

This case involves processing digital video images frame by frame to determine the time and location of flaming combustion when a sample is evaluated in a standardized flammability test. Several FAA regulatory test methods rely on operator observations and interpretation of video records to determine parameters such as time of burning, burn length, dripping, and other flaming characteristics. Automatically processing a video record of the tests can facilitate the data interpretation by removing the operator from the analysis. More precise timing and determinations of burn length, based on the video records can be made and quantified more accurately with analyses performed automatically. See Table 4 for tabulated results for material class.

Table 4: Tabulated results for the material class identified by the algorithm

Features	Material			
	Carbon Fiber	...	Polycarbonate	UPVC
Flame Time (s)	8.29	-	-	-
Burn Scale (in ²)	0.04	-	-	-
Flame Intensity	146	-	-	-
Charring Area (in ²)	0.09	-	-	-
Ignition Time (s)	0.04	-	-	-
Droplets	4	-	-	-

4.3.2.1 Features extracted by the object detection algorithm

Using only a video as the input yielded a set of tabulated results as outputs. After checking the videos manually to do a cross validation of the automatically derived measurements, the results were split into accurate and inaccurate ones. See Figure 45 for object detection results from the algorithm.

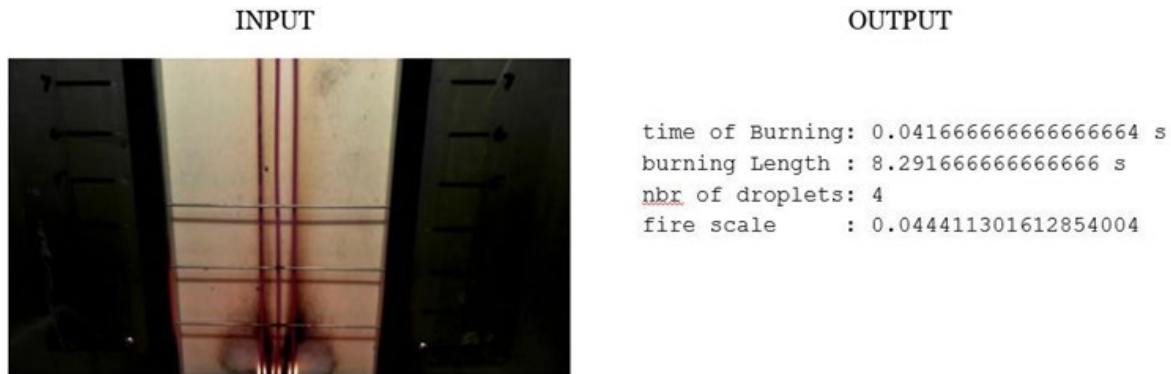


Figure 45: Object detection results from algorithm

Flame time and fire scale as trusted results, and with an FPS= 30 for example, a time measurement with 0.03s precision.

Number of droplets and ignition time is misleading because of the true positive specially for the ignition time.

$L=[0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,2,2,2,2,2,2,3,3,3,3,3,3,4,3,3,3,\dots,0,0,0,0,0,0,0,0]$

It was observed that the time of the true positive significantly affects the ignition time.

4.3.2.2 Features extracted by the image processing techniques

The algorithm can effectively extract the intensity of the fire and identify the time of its max value. After applying a cross validation while examining the videos manually it was concluded that the measurements can be trusted. See Figure 46 for light intensity extraction from flaming sample.



Figure 46: Light intensity extraction from flaming sample

After scaling the area by devising the number of black pixels by the frame size yielded a measurement which enable a comparative study of the burning materials. And while visualizing the results it was concluded that this measurement of the charring area can also be trusted. See Figure 47 for post-burn charring area extraction.

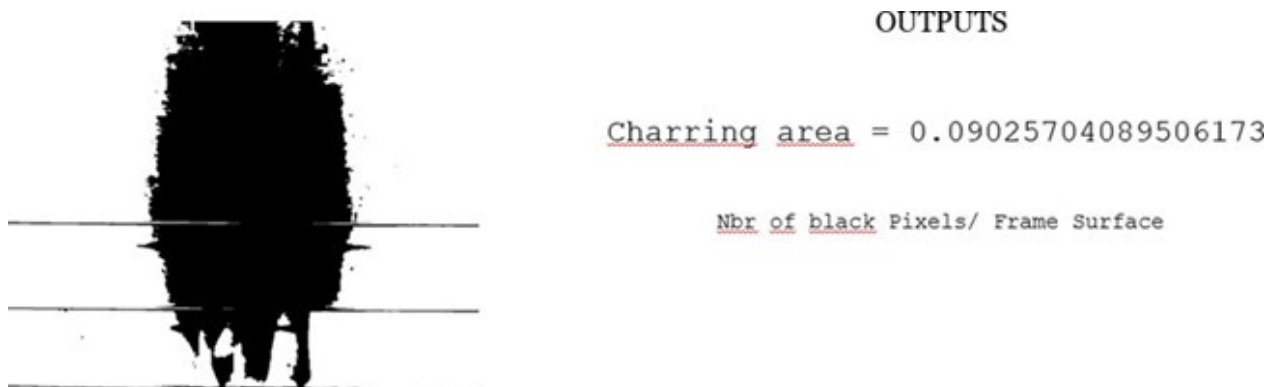


Figure 47: Post-burn charring area extraction

4.4 Discussion

The experiments conducted on the proposed model focused on exploring the characteristics of the input images and investigating the impact of preprocessing techniques on the accuracy and loss of convolutional neural networks specifically tailored to fire detection in this context. The analysis of these experiments yielded valuable insights, allowing the following conclusions:

- *Transfer learning*: Instead of starting the training process from scratch, the weights were leveraged, and features learned by a pre-trained model. This transfer learning approach served as a starting point for the model, enabling the benefit from the knowledge and experience captured by the pre-trained model. By using transfer learning, potential issues

related to overfitting were avoided and improved the overall performance of the model. The use of transfer learning resulted in even better results and a significant increase in average precision, all achieved within a shorter training time.

- *Data preprocessing and augmentation*: The importance of data preprocessing and augmentation techniques in enhancing the performance and results of the models was recognized. But as the results show, we needed to choose carefully the processing technique which better serves our case and purpose even if it's generally better and gives better results on other datasets.
- *Altering backbone architecture*: The backbone architecture of a CNN, typically composed of convolutional layers, plays a significant role in feature extraction. The impact of altering the architecture by increasing the depth of the network was explored. With each additional convolutional layer, the complexity of the features learned by the model increased. This deeper architecture allowed the model to capture more intricate patterns and details in the input data, leading to improved results and enhanced fire detection performance.
- *Examining the results*: The case result analysis directed how to proceed and evaluate the model then enabled the separation between trusted or misleading features. These conclusions highlight the importance of careful experimentation and optimization when developing models for fire detection. By leveraging transfer learning, optimizing data preprocessing and augmentation, and exploring deeper backbone architectures, we were able to push the boundaries of fire detection performance, paving the way for more efficient and accurate fire detection systems.

4.5 Conclusion

The comparative analysis conducted in this study provided valuable insights into the effectiveness of the proposed strategy. The new model's performance was compared with existing approaches and evaluated its accuracy rate and various performance metrics. The results of this comparative analysis were highly encouraging, highlighting the reliability and robustness of this strategy.

Summary

In this project, the main objective was to address the challenge of object detection in the FAA VFP videos for fire detection and quantification. The aim was to develop an automated system that can analyze fire videos and provide accurate assessments. To achieve this, a deep neural

network architecture was employed, and its performance thoroughly examined. To lay the groundwork for the study, an in-depth anatomical analysis of fire was conducted and valuable insights into flammability tests were gained. This comprehensive understanding of the fundamental aspects and characteristics of fire allowed the development of a more effective fire detection and quantification system. Machine learning algorithms, particularly deep neural networks, emerged as promising tools for analyzing fire videos. The proposed the use of a deep neural network architecture, specifically an object detection algorithm based on convolutional neural networks, was utilized for this study. CNNs are renowned for their exceptional capabilities in image classification and visual object detection, making them well-suited for the study objectives.

Additionally, the effects of image preprocessing and augmentation techniques were explored. Designing and implementing a systematic experimental protocol allowed various preprocessing and augmentation methods to be thoroughly investigated. This enabled the performance optimization of the deep neural network and ensured its robustness in detecting and quantifying fires in VFP videos. Furthermore, transfer learning was incorporated into the approach. Leveraging a pretrained model and deeper architectures utilized knowledge learned from large-scale datasets to enhance the performance of the model on the specific task of fire detection. This transfer learning strategy proved to be advantageous, resulting in significant improvements in terms of average precision, false negatives, and false positives reduction. The proposed strategy in this work demonstrated high performance values, showcasing the potential of deep neural networks as valuable tools for the general assessment of fires. The significant improvements achieved through this approach encourage further consideration of deep neural networks as aiding tools in fire analysis.

In conclusion, this study tackled the problem of object detection in VFP videos for fire detection and quantification. Through the utilization of a deep neural network architecture, along with an anatomical study on fire and the exploration of preprocessing and augmentation techniques, an automatic assessment tool for fire video analysis was developed. The positive results obtained from this proposed strategy highlight the potential of deep neural networks to enhance the accuracy and efficiency of fire assessment processes.

5 References

- Alzubaidi, L., Zhang, J., Humaidi, A., & Al-Dujaili, A. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, 8(1).
- Alzubaidi, L., Zhang, J., Humaidi, A., Al-Dujaili, A., Duan, Y., Al-Shamma, O., . . . Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, 8(1).
- Anaya, L., & Rehn, s. (n.d.). *Development of an Alternative Flammability Certification Test for Aircraft Electronic Equipment*. Pending Report.
- Babrauskas, V., & Grayson, S. (1992). *Heat Release in Fires*. New York: Elsevier Science Publishers.
- Barmpoutis, P., Dimitropoulos, K., Kaza, K., & Nikos, G. (2019). Fire Detection from Images using Faster R-CNN and Multidimensional Texture Analysis. *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Brighton, UK. doi:10.1109/ICASSP.2019.8682647
- Celik, T., & Demirel, H. (2009). Fire detection in video sequences using a generic color model. *Fire Safety Journal*, 44(2), 147-158.
- Chang, Y.-H. (2002). *Vision-based Fire Detection Using Video Sequences*. Taiwan: Chung Yuan Christian University. doi:<https://doi.org/10.6840/cycu201400483>
- Chen, T.-H., Wu, P.-H., & Chiou, Y.-C. (2004). An Early Fire-Detection Method Based on Image Processing . *ICIP '04*. Singapore: International Conference on Image Processing (ICIP).
- Chollet, F. (2019). *Deep Learning with Python*. Manning Publications Co.
- DeHaan, J. (1983). *Kirk's, Fire Investigation*. New Jersey: John Wiley & Sons. Inc.
- Demirel, H., & Celik, T. (2009). Fire detection in video sequences using a generic color model. *Fire Safety Journal*, 44(2), 147-158.
- Di Lascio, R., Greco, A., Saggese, A., Vento, M., Campilho, A., & Kamel, M. (2014). Improving Fire Detection Reliability by a Combination of Video analytics. *ICIAR 2014*. 11th International Conference on Image Analysis and Recognition.

- Drysdale, D. (1986). *An Introduction to Fire Dynamics*. Chichester, West Sussex, England: John Wiley & Sons, Ltd.
- Elliot, D. (1992). *A better Activation Function for Artificial Neural Networks*.
- FAA. (2014). *Development of a Flame Propagation Test Apparatus for Inaccessible Area Materials*. Solothurn, Switzerland: FAA.
- Foggia, P., Saggese, A., & Vento, M. (2015). Real-time fire detection for video-surveillance applications using a combination of experts based on color, shape, and motion. *IEEE Transactions on Circuits and Systems for Video Technology*, 25(9).
- Gold, S., & Rangarajan, A. (2022). Softmax to Softassign: Neural Network Algorithms for Combinatorial Optimization. *Journal of Artificial Neural Networks*.
- Habiboglu, Y., Günay, O., & Cetin, A. (2011). Covariance matrix-based fire and flame detection method in video. *Machine Vision and Applications*, 23(6), 1103– 1113.
- Karp, M. E. (2025). *Advanced Early Fire Detection in Aircraft Cargo with MACD And Passive UHF RFID Temperature Sensing While Maintaining False Alarm Resistence*. Federal Aviation Administration.
- Khandual, A. (2016). Testing & quality analysis of flammability of various fabrics, colourage. *Colourage*, 41(7), 41-15.
- Koirala, R. (2020). Fire Image Detection Based on Convolutional Neural Network. *Case Studies in Thermal Engineering*.
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444. doi:10.1038/nature14539
- Li, P., & Zhao, W. (2020, June). Image fire detection algorithms based on CNN. *Case Studies in Thermal Engineering*, 19. doi:10.1016/j.csite.2020.100625
- Li, P., & Zhao, W. (2020). Image fire detection algorithms based on CNN, Case Studies in Thermal Engineering. *Case Studies in Thermal Engineering*, 19. doi:10.1016/j.csite.2020.100625
- Liu, Z., & Kim, A. (2003). Review of recent developments in fire detection technologies. *Journal of Fire Protection Engineering*, 13(2). doi:10.1177/1042391503013002003
- Marker, T. (2019). *Aircraft Materials Fire Test Handbook*.

- Muhammad, K., Ahmad, J., & Baik, S. (2017). Early fire detection using convolutional neural networks during surveillance for effective disaster management. *Neurocomputing*, 288, 30-42. doi:10.1016/j.neucom.2017.04.083
- Muhammad, K., Ahmad, J., Mehmood, I., Rho, S., & Baik, S. (2018). Convolutional neural networks based fire detection in surveillance videos. *IEEE Access PP*, 99.
- NFPA. (1991). *NFPA 325, Guide to Fire Hazard Properties of Flammable Liquids Gases and Volatile Solids*. National Fire Protection Association.
- O'Connor, T., & Hagen, E. (1999). *Activation of Oxygen Generators in Proximity to Combustible Materials*.
- Petra, M., Bakar, M., DeSilva, L., & Umar, M. (2017). State of the art of smoke and fire detection using image processing. *International Journal of Signal and Imaging Systems Engineering*, 10, 22. doi:10.1504/IJSISE.2017.10005428
- Rafiee, A., Dianat, R., Jamshidi, M., Tavakoli, R., & Abbaspour, S. (2011). Fire and smoke detection using wavelet analysis and disorder characteristics. International Conference on Computer Research and Development.
- Rosebrock, A. (2017). *Deep Learning for Computer Vision with Python*. PyImageSearch.
- Shamsoshoara, A., Afghah, F., Razi, A., Zheng, L., Fule, P., & Blasch, E. (2021). Aerial Imagery Pile burn detection using Deep Learning: the FLAME dataset, School of Informatics. *Computer Networks*, 193(4). doi:10.1016/j.comnet.2021.108001
- Singh, H. (2019). *Practical Machine Learning and Image Processing: For Facial Recognition, Object Detection, and Pattern Recognition Using Python*.
- Visa, S., Ramsay, B., Ralescu, A., & van der Knapp, E. (2011). Confusion Matrix-based Feature Selection. *Proceedings of The 22nd Midwest Artificial Intelligence and Cognitive Science Conference (MAICS)*. 710, pp. 120-127. Cincinnati, Ohio: CEUR Workshop Proceedings.
- Yuan, C., Liu, Z., & Zhang, Y. (2015). UAV-based forest fire detection and tracking using image processing techniques. *ICUAS 2015*. International Conference on Unmanned Aircraft Systems. doi:10.1109/ICUAS.2015.7152345
- Zaidi, N., Lokman, N., Daud, M., Achmad, A., & Khor, A. (2015). Fire recognition using RGB and YCbCr color space. *Journal of Engineering and Applied Sciences*, 10, 9786-9790.

Zhao, Z.-Q. (2019, January). Object Detection With Deep Learning. *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1-15.